ADVERSARIAL CAUSAL AUGMENTATION FOR GRAPH COVARIATE SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Out-of-distribution (OOD) generalization on graphs is drawing widespread attention. However, existing efforts mainly focus on the OOD issue of correlation shift. While another type, covariate shift, remains largely unexplored but is the focus of this work. From a data generation view, causal features are stable substructures in data, which play key roles in OOD generalization. While their complementary parts, environments, are unstable features that often lead to various distribution shifts. Correlation shift establishes spurious statistical correlations between environments and labels. In contrast, covariate shift means that there exist unseen environmental features in test data. Existing strategies of graph invariant learning and data augmentation suffer from limited environments or unstable causal features, which greatly limits their generalization ability on covariate shift. In view of that, we propose a novel graph augmentation strategy: Adversarial Causal Augmentation (**AdvCA**), to alleviate the covariate shift. Specifically, it adversarially augments the data to explore diverse distributions of the environments. Meanwhile, it keeps the causal features invariant across diverse environments. It maintains the environmental diversity while ensuring the invariance of the causal features, thereby effectively alleviating the covariate shift. Extensive experimental results with in-depth analyses demonstrate that AdvCA can outperform 14 baselines on synthetic and real-world datasets with various covariate shifts. We also provide an anonymous code link of our work: https://anonymous.4open.science/r/AdvCA-68BF

1 INTRODUCTION

Graph learning mostly follows the assumption that training and test data are independently drawn from an identical distribution. Such an assumption is difficult to be satisfied in the wild, due to out-of-distribution (OOD) issues (Shen et al., 2021), where the training and test data are from different distributions. Hence, OOD generalization on graphs is attracting widespread attention (Li et al., 2022b). However, existing studies mostly focus on correlation shift, which is just one type of OOD issue (Ye et al., 2022; Wiles et al., 2022). While another type, covariate shift, remains largely unexplored but is the focus of our work.

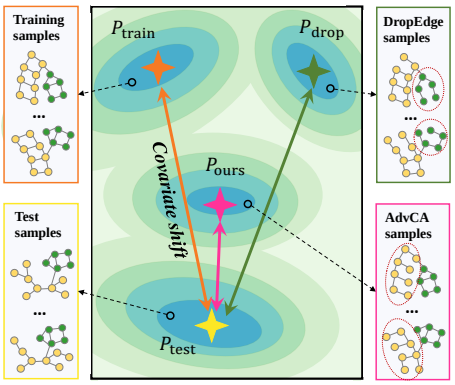

Figure 1: Distribution comparisons in data space. P_{train} and P_{test} denote the training and test distributions. P_{drop} and P_{ours} represent the distributions of augmented data via DropEdge and AdvCA. AdvCA establishes a smaller covariate shift (↔) with test distribution than DropEdge (↔).

Covariate shift is in stark contrast to correlation shift *w.r.t.* causal and environmental features of data. Specifically, from a data generation view, causal features are the substructures of the entire graphs that truly reflect the predictive property of data, while their complementary parts are the environmental features that are noncausal to the predictions. Following prior studies (Arjovsky et al., 2019; Wu et al., 2022b), we assume causal features are stable and invariant across distributions, in contrast to the environmental features. Correlation shift denotes that environments and labels establish inconsistent statistical correlations in training and test data; whereas,

covariate shift means that the environmental features in test data are unseen in training data (Ye et al., 2022; Wiles et al., 2022; Gui et al., 2022). For 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 causal features that determine the predictive property of molecules. While scaffolds (*e.g.,* carbon rings) are irrelevant patterns (Wu et al., 2018), which can be seen as the environments. In practice, we often need to use molecular graphs collected in the past to train models, hoping that the models can predict the properties of molecules with new scaffolds in the future (Hu et al., 2020).

Because of the differences between correlation and covariate shifts, we take a close look at the existing efforts on graph generalization. Existing efforts (Li et al., 2022b) mainly fall into the following research lines, each of which has inherent limitations to solve covariate shift.

- **Invariant graph learning** (Wu et al., 2022b; Liu et al., 2022; Sui et al., 2022) gradually becomes a prevalent paradigm for OOD generalization. The main idea is to capture the causal features by minimizing the empirical risks within different environments. Unfortunately, it implicitly makes a prior assumption that all 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 only alleviate the spurious correlations that are hidden in the training data, but fail to extrapolate test distributions with unseen environments.
- **Graph data augmentation** (Ding et al., 2022; Zhao et al., 2022) perturbs graph features to enrich the distribution seen during training for better generalization. It can be roughly divided into node-level (Kong et al., 2022), edge-level (Rong et al., 2020), and graph-level (Wang et al., 2021; Han et al., 2022) with random (You et al., 2020) or adversarial strategies (Suresh et al., 2021). However, they are prone to destroy the causal features, which easily loses control of the perturbed distributions. For example, in Figure 1, the random strategy of DropEdge (Rong et al., 2020) will inevitably perturb the causal features (highlighted by red circles). As such, it fails to alleviate the covariate shift (↔), even degenerating the generalization ability.

Scrutinizing the limitations of the aforementioned studies, insufficient environments and unstable causal features largely hinder the ability of these generalization efforts against the covariate shift. Hence, we naturally ask a question: "*Can the augmented samples simultaneously preserve the diversity of environmental features and the invariance of causal features?*"

Towards this end, we first propose two principles for graph augmentation: environmental diversity and causal invariance. Specifically, environmental diversity encourages the augmentation to extrapolate unseen environments; meanwhile, causal invariance shortens the distribution gap between the augmented data and test data. To achieve these principles, we design a novel graph augmentation strategy: Adversarial Causal Augmentation (**AdvCA**). Specifically, we augment the graphs by a network, named adversarial augmenter. It adversarially generates the masks on edges and node features, which makes OOD exploration for improving the environmental diversity. To maintain the invariance of the causal features, we adopt another network, named causal generator. It generates the masks that capture causal features. Finally, we delicately combine these masks and apply them to graph data. As shown in Figure 1, AdvCA only perturbs the environmental features, while keeping the causal parts invariant. Our quantitative experiments also verify that AdvCA can narrow the distribution gap between the augmented data and test data, as illustrated in Figure 1 (↔), thereby effectively overcoming the covariate shift issues. Our contributions can be summarized as:

- **Problem**: We argue that existing graph generalization efforts cannot guarantee both environmental diversity and causal invariance. Hence, they expose vulnerability under covariate shift.
- **Method**: We design a graph augmentation method, AdvCA, which focuses on covariate shift issues. It maintains the invariance of causal features while ensuring environmental diversity.
- **Experiment**: We conduct extensive experiments on synthetic and real datasets. The experimental results with in-depth analyses demonstrate the effectiveness of AdvCA.

2 Preliminaries

In this section, we first give the formal definitions of causal features, environmental features, and graph covariate shift. Then we present the problem of graph classification under covariate shift.

2.1 NOTATIONS

In this paper, 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 $g = (\mathbf{A}, \mathbf{X}) \in \mathbb{G}$ denote a graph, where $\mathbf{A}$ and $\mathbf{X}$ 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 $\mathcal{D} = \{(g_i, y_i)\}$ denote a dataset that is divided into a training set $\mathcal{D}_{\text{tr}} = \{(g_i^e, y_i^e)\}_{e \in \mathcal{E}_{\text{tr}}}$ and a test set $\mathcal{D}_{\text{te}} = \{(g_i^e, y_i^e)\}_{e \in \mathcal{E}_{\text{te}}}$. $\mathcal{E}_{\text{tr}}$ and $\mathcal{E}_{\text{te}}$ are the index sets of training and test environments, respectively. In this work, we focus on graph classification scenario, which aims to train models with $\mathcal{D}_{\text{tr}}$ and infer the labels in $\mathcal{D}_{\text{te}}$.

2.2 DEFINITIONS AND PROBLEM FORMATIONS

The inner mechanism of the labeling rule $\mathbb{G} \to \mathbb{Y}$ usually depends on the causal features, which are particular subparts of the entire data (Arjovsky et al., 2019; Kaddour et al., 2022; Wu et al., 2022b; Ye et al., 2022; Lu et al., 2021). While their complementary parts, environmental features, are noncausal for predicting the graphs. Now we give a formal definition of these features.

Definition 1 (Causal & Environmental Feature) *Given G contains two features G_{cau}, G_{env}, and they satisfy: $G_{\text{cau}} \cup G_{\text{env}} = G$. If they obey the following conditions: i) (sufficiency condition) $P(Y|G_{\text{cau}}) = P(Y|G)$; ii) (independence condition) $Y \perp\!\!\!\perp G_{\text{env}} \mid G_{\text{cau}}$, then we define G_{cau} and G_{env} as the causal feature and environmental feature, respectively.*

We can observe that environmental features easily change outside the training distribution, owing to their noncausal nature. Specifically, there exist two main reasons that lead to poor generalization. Firstly, 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 or correlation shift (Ye et al., 2022). Secondly, 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 in literature (Gui et al., 2022). Now we give a formal definition of the covariate shift on graphs.

Definition 2 (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*

$$\text{GCS}(P_{\text{tr}}, P_{\text{te}}) = \frac{1}{2} \int_{\mathcal{S}} |P_{\text{tr}}(g) - P_{\text{te}}(g)| dg, \tag{1}$$

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

$\text{GCS}(P_{\text{tr}}, P_{\text{te}})$ is always bounded in $[0, 1]$. The issue of graph covariate shift is very common in practice. For example, the chemical properties of molecules are mainly determined by specific functional groups, which can be regarded as causal features to predict these properties (Arjovsky et al., 2019; Wu et al., 2022b). While their scaffold structures (Wu et al., 2018), which are often irrelevant to their properties, can be seen as environmental features. In practice, we often need to train models on past molecular graphs, and hope that the model can predict the properties of future molecules with novel scaffolds (Hu et al., 2020). Hence, this work focuses on the covariate shift issues. Now we give a formal definition of this problem as follows.

Problem 1 (Graph Classification under Covariate Shift) *Given the training and test sets with environment sets $\mathcal{E}_{\text{tr}}$ and $\mathcal{E}_{\text{te}}$, they follow distributions P_{tr} and P_{te}, and they satisfy: $\text{GCS}(P_{\text{tr}}, P_{\text{te}}) > 0$. We aim to use the data collected from training environments $\mathcal{E}_{\text{tr}}$, and learn a powerful graph classifier $f^* : \mathbb{G} \to \mathbb{Y}$ that performs well in all possible test environments $\mathcal{E}_{\text{te}}$:*

$$f^* = \arg\min_f \sup_{e \in \mathcal{E}_{\text{te}}} \mathbb{E}^e[\ell(f(g), y)], \tag{2}$$

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

Problem 1 states that it is unrealistic for the training set to cover all possible environments in the test set. It means that we have to extrapolate unknown environments by using the limited training environments at hand, which makes this problem more challenging.

3 METHODOLOGY

In this section, we first propose two principles for graph data augmentation. Guided by these principles, we design a new graph augmentation strategy that can effectively solve Problem 1.

3.1 TWO PRINCIPLES FOR GRAPH AUGMENTATION

Scrutinizing Problem 1, we observe that the covariate shift is mainly caused by the scarcity of training environments. Existing efforts (Liu et al., 2022; Sui et al., 2022; Wu et al., 2022b) make intervention or replacement of the environments to capture causal features. However, these environmental features still stem from the training distribution, which may result in a limited diversity of the environments. Worse still, if the environments are too scarce, the model will inevitably learn the shortcuts between these environmental features, resulting in suboptimal learning of the causal parts. To this end, we propose the first principle for data augmentation:

Principle 1 (Environmental Diversity) *Given a set of graphs $\{g\}$ with distribution function P. Let $T(\cdot)$ denote an augmentation function that augments graphs $\{T(g)\}$ to distribution function $\widetilde{P}$. Then $T(\cdot)$ should meet* $\mathrm{GCS}(P, \widetilde{P}) \to 1$.

Principle 1 states that $\widetilde{P}$ should keep away from the original distribution P. Hence, the distribution of augmented data tends not to overlap with the original distribution, which encourages the diversity of environmental features. However, from a data generation perspective, causal features are invariant and shared across environments (Kaddour et al., 2022), so they are essential features for OOD generalization. Since Principle 1 does not expose any constraint on the invariant property of the augmented distribution, we here propose the second principle for augmentation:

Principle 2 (Causal Invariance) *Given a set of graphs $\{g\}$ with a corresponding causal feature set $\{g_{\mathrm{cau}} = (\mathbf{A}_{\mathrm{cau}}, \mathbf{X}_{\mathrm{cau}})\}$. Let $T(\cdot)$ denote an augmentation function that augments graphs $\{T(g)\}$ with a corresponding causal feature set $\{\widetilde{g}_{\mathrm{cau}} = (\widetilde{\mathbf{A}}_{\mathrm{cau}}, \widetilde{\mathbf{X}}_{\mathrm{cau}})\}$. Then $T(\cdot)$ should meet $\mathbb{E}[\|\mathbf{A}_{\mathrm{cau}} - \widetilde{\mathbf{A}}_{\mathrm{cau}}\|_F^2] \to 0$ and $\mathbb{E}[\|\mathbf{X}_{\mathrm{cau}} - \widetilde{\mathbf{X}}_{\mathrm{cau}}\|_F^2] \to 0$, where $\|\cdot\|_F^2$ is the Frobenius norm.*

Principle 2 emphasizes the invariance of the graph structures and node features in causal parts after data augmentation. As illustrated in Figure 1, Principle 1 keeps the distribution of augmented data away from the training distribution; meanwhile, Principle 2 restricts the distribution of augmented data not too far from the test distribution. These two principles complement each other, and further cooperate together to alleviate the covariate shift.

3.2 OUT-OF-DISTRIBUTION EXPLORATION

Given a GNN model $f(\cdot)$ with parameters θ, we decompose $f = \Phi \circ h$, where $h(\cdot) : \mathbb{G} \to \mathbb{R}^d$ is a graph encoder to yield d-dimensional representations, and $\Phi(\cdot) : \mathbb{R}^d \to \mathbb{Y}$ is a classifier. To comply with Principle 1, we need to do OOD exploration. Inspired by distributionally robust optimization (Sagawa et al., 2020), we consider the following optimization objective:

$$\min_{\theta} \left\{ \sup_{\widetilde{P}} \{\mathbb{E}_{\widetilde{P}}[\ell(f(g), y)] : D(\widetilde{P}, P) \leq \rho\} \right\}, \tag{3}$$

where P and $\widetilde{P}$ denote 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 distance ρ of the distribution P. To better measure the distance between distributions, as suggested by Sinha et al. (2018), we adopt the Wasserstein distance (Arjovsky et al., 2017; Volpi et al., 2018) as the distance metric. The distance metric function can be defined as:

$$D(\widetilde{P}, P) := \inf_{\mu \in \Gamma(\widetilde{P}, P)} \mathbb{E}_{\mu}[c(\widetilde{g}, g)], \tag{4}$$

where $\Gamma(\widetilde{P}, P)$ is the set of all couplings of $\widetilde{P}$ and P, $c(\cdot, \cdot)$ is the cost function. Studies (Dosovitskiy & Brox, 2016; Volpi et al., 2018) 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 following transportation cost:

$$c(\widetilde{g}, g) = \|h(\widetilde{g}) - h(g)\|_2^2. \tag{5}$$

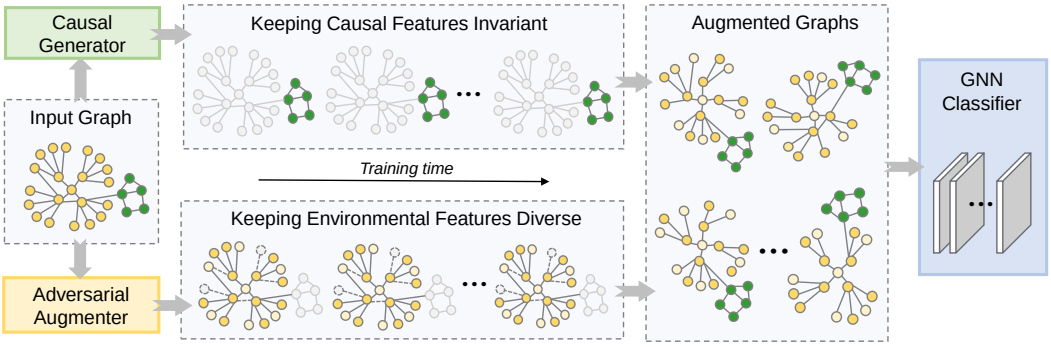

Figure 2: The overview of Adversarial Causal Augmentation (AdvCA) Framework.

It denotes the "cost" of augmenting the graph g to $\widetilde{g}$. We can observe that it is difficult to set a proper ρ in Equation 3. Instead, we consider the Lagrangian relaxation for a fixed penalty coefficient γ. Inspired by Sinha et al. (2018), we can reformulate Equation 3 as follows:

$$\min_{\theta} \left\{ \sup_{\widetilde{P}} \{\mathbb{E}_{\widetilde{P}}[\ell(f(g), y)] - \gamma D(\widetilde{P}, P)\} = \mathbb{E}_{P}[\phi(f(g), y)] \right\}, \tag{6}$$

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

$$\nabla_{\theta} \phi(f(g), y) = \nabla_{\theta} \ell(f(\widetilde{g}^{*}), y), \tag{7}$$

$$\text{where } \widetilde{g}^{*} = \arg\max_{\widetilde{g} \in \mathbb{G}} \{\ell(f(\widetilde{g}), y) - \gamma c(\widetilde{g}, g)\}. \tag{8}$$

$\widetilde{g}^{*}$ is an augmented view of the original data g. Hence, to achieve OOD exploration, we just need to perform data augmentation via Equation 8 on the original data g.

3.3 Adversarial Causal Augmentation

Equation 8 endows the ability of OOD exploration to data augmentation, which makes the augmented data meet Principle 1. In addition, to achieve Principle 2, we also need to implement causal feature learning based on the sufficiency and independence conditions in Definition 1. Hence, we design a novel graph augmentation strategy: Adversarial Causal Augmentation (**AdvCA**). The overview of the proposed framework is depicted in Figure 2, which mainly consists of two components: adversarial augmenter and causal generator. Adversarial augmenter achieves OOD exploration through adversarial data augmentation, which encourages the diversity of environmental features; meanwhile, the causal generator keeps causal feature invariant by identifying causal features from data. Below we elaborate on the implementation details.

Adversarial Augmenter & Causal Generator. We design two networks, adversarial augmenter $T_{\theta_1}(\cdot)$ and causal generator $T_{\theta_2}(\cdot)$, which generate masks for nodes and edges of graphs. They have the same structure and are parameterized by θ_1 and θ_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 $\widetilde{h}(\cdot)$. 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:

$$\mathbf{Z} = \widetilde{h}(g), \quad \mathbf{M}^x_i = \sigma(\mathrm{MLP}_1(\mathbf{h}_i)), \quad \mathbf{M}^a_{ij} = \sigma(\mathrm{MLP}_2([\mathbf{z}_i, \mathbf{z}_j])), \tag{9}$$

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 Causal Augmentation. To estimate $\widetilde{g}^{*}$ in Equation 8, we define the adversarial learning objective as:

$$\max_{\theta_1} \{\mathcal{L}_{\mathrm{adv}} = \mathbb{E}_{P_{\mathrm{tr}}}[\ell(f(T_{\theta_1}(g)), y) - \gamma c(T_{\theta_1}(g), g)]\}. \tag{10}$$

Then we can augment the graph by $T_{\theta_1}(g) = (\mathbf{A} \odot \mathbf{M}^a_{\mathrm{adv}}, \mathbf{X} \odot \mathbf{M}^x_{\mathrm{adv}})$, where $\odot$ is the broadcasted element-wise product. Although adversarially augmented graphs guarantee environmental diversity,

it inevitably destroys the causal parts. Therefore, we utilize the causal generator $T_{\theta_2}(\cdot)$ to capture causal features and combine them with diverse environmental features. Following the sufficiency and independence conditions in Definition 1, we define the causal learning objective as:

$$\min_{\theta,\theta_2} \{\mathcal{L}_{\text{cau}} = \mathbb{E}_{P_{\text{tr}}}[\ell(f(T_{\theta_2}(g)), y) + \ell(f(\widetilde{g}), y)]\}, \tag{11}$$

where $\widetilde{g} = (\mathbf{A} \odot \widetilde{\mathbf{M}}^a, \mathbf{X} \odot \widetilde{\mathbf{M}}^x)$ is the augmented graph. It adopts the mask combination strategy: $\widetilde{\mathbf{M}}^a = (\mathbf{1}^a - \mathbf{M}^a_{\text{cau}}) \odot \mathbf{M}^a_{\text{adv}} + \mathbf{M}^a_{\text{cau}}$ and $\widetilde{\mathbf{M}}^x = (\mathbf{1}^x - \mathbf{M}^x_{\text{cau}}) \odot \mathbf{M}^x_{\text{adv}} + \mathbf{M}^x_{\text{cau}}$, where $\mathbf{M}^a_{\text{cau}}$ and $\mathbf{M}^x_{\text{cau}}$ are generated by $T_{\theta_2}(\cdot)$, $\mathbf{1}^a$ and $\mathbf{1}^x$ are all-one matrices, and if there is no edge between node i and node j, then we set $\mathbf{1}^a_{ij}$ to 0.

Now we explain this combination strategy. Taking $\widetilde{\mathbf{M}}^x$ as an example, since $\mathbf{M}^x_{\text{cau}}$ denotes the captured causal regions via $T_{\theta_2}(\cdot)$, $\mathbf{1}^x - \mathbf{M}^x_{\text{cau}}$ represents the complementary parts, which are environmental regions. $\mathbf{M}^x_{\text{adv}}$ represents the adversarial perturbation, so $(\mathbf{1}^x - \mathbf{M}^x_{\text{cau}}) \odot \mathbf{M}^x_{\text{adv}}$ is equivalent to applying the adversarial perturbation on environmental features, meanwhile, sheltering the causal features. Finally, $+\mathbf{M}_{\text{cau}}$ indicates that the augmented data should retain the original causal features. Hence, this combination strategy achieves both environmental diversity and causal invariance. Inspecting Equation 11, the first term indicates that causal features are enough for predictions, thus satisfying the sufficiency condition. While the second term encourages causal features to make right predictions after perturbing the environments, thereby satisfying the independence condition.

Regularization. For Equation 10, the adversarial optimization tends to remove more nodes and edges, so we should also constrain the perturbations. Although Equation 11 satisfies the sufficiency and independence conditions, it is necessary to impose constraints on the ratio of the causal features to prevent trivial solutions. Hence, we first define the regularization function $r(\mathbf{M}, k, \lambda) = (\sum_{ij} \mathbf{M}_{ij}/k - \lambda) + (\sum_{ij} \mathbb{I}[\mathbf{M}_{ij} > 0]/k - \lambda)$, where k is the total number of elements to be constrained, $\mathbb{I} \in \{0, 1\}$ is an indicator function. The first term penalizes the average ratio close to λ, 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 causal learning as:

$$\mathcal{L}_{\text{reg}_1} = \mathbb{E}_{P_{\text{tr}}}[r(\mathbf{M}^x_{\text{adv}}, n, \lambda_a) + r(\mathbf{M}^a_{\text{adv}}, m, \lambda_a)], \tag{12}$$

$$\mathcal{L}_{\text{reg}_2} = \mathbb{E}_{P_{\text{tr}}}[r(\mathbf{M}^x_{\text{cau}}, n, \lambda_c) + r(\mathbf{M}^a_{\text{cau}}, m, \lambda_c)], \tag{13}$$

where $\lambda_c \in (0, 1)$ is the ratio of causal features, we set $\lambda_a = 1$ for adversarial learning, which can alleviate excessive perturbations. The detailed algorithm of AdvCA is provided in Appendix A.1.

4 Experiments

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

- **RQ1:** Compared to existing efforts, how does AdvCA perform under covariate shift?
- **RQ2:** Can the proposed AdvCA achieve the principles of environmental diversity and causal invariance, thereby effectively alleviating the covariate shift?
- **RQ3:** How do the different components of AdvCA affect performance?

4.1 Experimental Settings

Datasets. We use graph OOD datasets (Gui et al., 2022) and OGB datasets (Hu et al., 2020), which include Motif, CMNIST, Molbbbp and Molhiv. Following Gui et al. (2022), we adopt the base, color, size and scaffold data splitting to create various covariate shifts. The details of the datasets, metrics and implementation details of AdvCA are provided in Appendix A.2 and A.3.

Baselines. We adopt 14 baselines, which can be divided into the following three specific categories:

- **Generalization Algorithms:** Empirical Risk Minimization (ERM), IRM (Arjovsky et al., 2019), GroupDRO (Sagawa et al., 2020), VREx (Krueger et al., 2021).
- **Graph Invariant Learning and Generalization:** DIR (Wu et al., 2022b), CAL (Sui et al., 2022), GSAT (Miao et al., 2022), OOD-GNN (Li et al., 2022a), StableGNN (Fan et al., 2021).
- **Graph Data Augmentation:** DropEdge (Rong et al., 2020), GREA (Liu et al., 2022), FLAG (Kong et al., 2022), M-Mixup (Wang et al., 2021), $\mathcal{G}$-Mixup (Han et al., 2022).

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.

Method	Motif		CMNIST	Molbbbp		Molhiv	
	base	size	color	scaffold	size	scaffold	size
ERM	$68.66_{\pm 4.25}$	$51.74_{\pm 2.88}$	$28.60_{\pm 1.87}$	$68.10_{\pm 1.68}$	$78.29_{\pm 3.76}$	$69.58_{\pm 2.51}$	$59.94_{\pm 2.37}$
IRM	$70.65_{\pm 4.17}$	$51.41_{\pm 3.78}$	$27.83_{\pm 2.13}$	$67.22_{\pm 1.15}$	$77.56_{\pm 2.48}$	$67.97_{\pm 1.84}$	$59.00_{\pm 2.92}$
GroupDRO	$68.24_{\pm 8.92}$	$51.95_{\pm 5.86}$	$29.07_{\pm 3.14}$	$66.47_{\pm 2.39}$	$79.27_{\pm 2.43}$	$70.64_{\pm 2.57}$	$58.98_{\pm 2.16}$
VREx	$\underline{71.47_{\pm 6.69}}$	$52.67_{\pm 5.54}$	$28.48_{\pm 2.87}$	$68.74_{\pm 1.03}$	$78.76_{\pm 2.37}$	$70.77_{\pm 2.84}$	$58.53_{\pm 2.88}$
DIR	$62.07_{\pm 8.75}$	$52.27_{\pm 4.56}$	$\underline{33.20_{\pm 6.17}}$	$66.86_{\pm 2.25}$	$76.40_{\pm 4.43}$	$68.07_{\pm 2.29}$	$58.08_{\pm 2.31}$
CAL	$65.63_{\pm 4.29}$	$51.18_{\pm 5.60}$	$27.99_{\pm 3.24}$	$68.06_{\pm 2.60}$	$\underline{79.50_{\pm 4.81}}$	$67.37_{\pm 3.61}$	$57.95_{\pm 2.24}$
GSAT	$62.80_{\pm 11.41}$	$53.20_{\pm 8.35}$	$28.17_{\pm 1.26}$	$66.78_{\pm 1.45}$	$75.63_{\pm 3.83}$	$68.66_{\pm 1.35}$	$58.06_{\pm 1.98}$
OOD-GNN	$61.10_{\pm 7.87}$	$52.61_{\pm 4.67}$	$26.49_{\pm 2.94}$	$66.72_{\pm 1.23}$	$79.48_{\pm 4.19}$	$70.46_{\pm 1.97}$	$60.60_{\pm 3.77}$
StableGNN	$57.07_{\pm 14.10}$	$46.93_{\pm 8.85}$	$28.38_{\pm 3.49}$	$66.74_{\pm 1.30}$	$77.47_{\pm 4.69}$	$68.44_{\pm 1.33}$	$56.71_{\pm 2.79}$
DropEdge	$45.08_{\pm 4.46}$	$45.63_{\pm 4.61}$	$22.65_{\pm 2.90}$	$66.49_{\pm 1.55}$	$78.32_{\pm 3.44}$	$\underline{70.78_{\pm 1.38}}$	$58.53_{\pm 1.26}$
GREA	$56.74_{\pm 9.23}$	$\underline{54.13_{\pm 10.02}}$	$29.02_{\pm 3.26}$	$\underline{69.72_{\pm 1.66}}$	$77.34_{\pm 3.52}$	$67.79_{\pm 2.56}$	$\underline{60.71_{\pm 2.20}}$
FLAG	$61.12_{\pm 5.39}$	$51.66_{\pm 4.14}$	$32.30_{\pm 2.69}$	$67.69_{\pm 2.36}$	$79.26_{\pm 2.26}$	$68.45_{\pm 2.30}$	$60.59_{\pm 2.95}$
M-Mixup	$70.08_{\pm 3.82}$	$51.48_{\pm 4.91}$	$26.47_{\pm 3.45}$	$68.75_{\pm 0.34}$	$78.92_{\pm 2.43}$	$68.88_{\pm 2.63}$	$59.03_{\pm 3.11}$
$\mathcal{G}$-Mixup	$59.66_{\pm 7.03}$	$52.81_{\pm 6.73}$	$31.85_{\pm 5.82}$	$67.44_{\pm 1.62}$	$78.55_{\pm 4.16}$	$70.01_{\pm 2.52}$	$59.34_{\pm 2.43}$
AdvCA (ours)	$\mathbf{73.64_{\pm 5.15}}$	$\mathbf{55.85_{\pm 7.98}}$	$\mathbf{36.37_{\pm 4.44}}$	$\mathbf{70.79_{\pm 1.53}}$	$\mathbf{81.03_{\pm 5.15}}$	$\mathbf{71.15_{\pm 1.81}}$	$\mathbf{61.64_{\pm 3.37}}$
Improvement	$\uparrow 2.17\%$	$\uparrow 1.72\%$	$\uparrow 3.17\%$	$\uparrow 1.07\%$	$\uparrow 1.53\%$	$\uparrow 0.37\%$	$\uparrow 0.93\%$

4.2 Main results (RQ1)

To demonstrate the superiority of AdvCA, we first make comprehensive comparisons with baseline methods. The implementation settings and details of baselines are provided in Appendix A.4. All experimental results are summarized in Table 1. We have the following **Obs**ervations.

Obs1: Most generalization and augmentation methods fail under covariate shift. Generalization and data augmentation algorithms perform well on certain datasets or shifts. 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 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), $\downarrow$11.92% and $\downarrow$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.

Obs2: AdvCA consistently outperforms all baseline methods. Compared with ERM, AdvCA can obtain significant improvements. For two types of covariate shifts on Motif, AdvCA surpasses ERM by 4.98% and 4.11%, respectively. In contrast to the large performance variances on different datasets achieved by baselines, AdaCA consistently obtains the leading performance across the board. For CMNIST, AdvCA 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 AdvCA can overcome the shortcomings of invariant learning and data augmentation. Armed with the principles of environmental diversity and causal invariance, AdvCA 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 AdvCA under correlation shift, and the results are presented in Appendix C.1.

4.3 Covariate shift and visualizations (RQ2)

In this section, we conduct quantitative experiments to demonstrate that AdvCA can shorten the distribution gap, as shown in Figure 1. Specifically, we utilize $\mathrm{GCS}(\cdot,\cdot)$ as the measurement to quantify the degree of covariate shift. The detailed estimation procedure is provided in Appendix B. To make comprehensive comparisons, we select four different types of covariate shift: base, size, color and scaffold, to conduct experiments. We choose three data augmentation baselines, DropEdge, FLAG and $\mathcal{G}$-Mixup, which augment graphs from different views. 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 $\mathrm{GCS}(P_{\mathrm{aug}}, P_{\mathrm{tr}})$ and $\mathrm{GCS}(P_{\mathrm{aug}}, P_{\mathrm{te}})$, respectively. From the results in Table 2, we have the following observations.

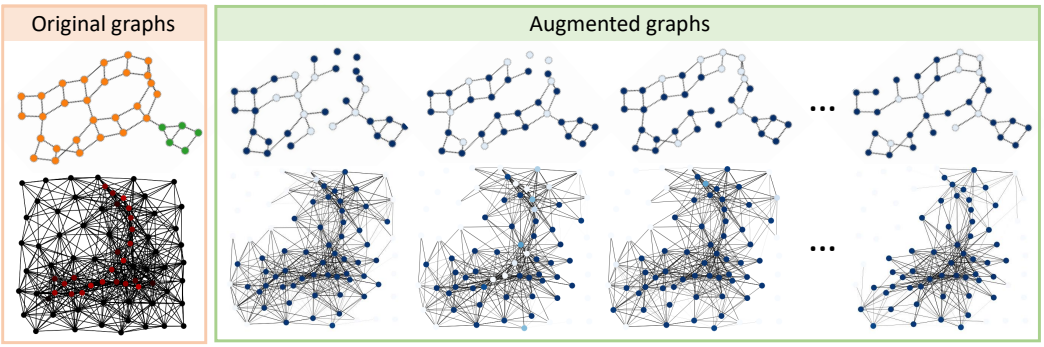

Figure 3: Visualizations of the augmented graphs via AdvCA.

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	$0.557_{\pm 0.141}$	0	$0.522_{\pm 0.421}$	0	$0.490_{\pm 0.226}$	0	$0.419_{\pm 0.079}$
DropEdge	$0.772_{\pm 0.213}$	$0.515_{\pm 0.033}$	$0.851_{\pm 0.138}$	$0.161_{\pm 0.271}$	$0.627_{\pm 0.186}$	$0.539_{\pm 0.260}$	$0.758_{\pm 0.192}$	$0.737_{\pm 0.211}$
FLAG	$0.001_{\pm 0.001}$	$0.533_{\pm 0.016}$	$0.002_{\pm 0.018}$	$0.507_{\pm 0.121}$	$0.003_{\pm 0.002}$	$0.442_{\pm 0.062}$	$0.001_{\pm 0.001}$	$0.413_{\pm 0.088}$
$\mathcal{G}$-Mixup	$0.690_{\pm 0.186}$	$0.472_{\pm 0.043}$	$0.816_{\pm 0.154}$	$0.299_{\pm 0.343}$	$0.408_{\pm 0.228}$	$0.351_{\pm 0.318}$	$0.551_{\pm 0.258}$	$0.545_{\pm 0.231}$
AdvCA	$0.369_{\pm 0.169}$	$\mathbf{0.462_{\pm 0.063}}$	$0.649_{\pm 0.143}$	$\mathbf{0.098_{\pm 0.070}}$	$0.516_{\pm 0.106}$	$\mathbf{0.307_{\pm 0.108}}$	$0.422_{\pm 0.049}$	$\mathbf{0.393_{\pm 0.028}}$

Obs3: AdvCA effectively closes the distribution gap with test distribution. "Original" represents the original training distribution without augmentation. We observe that there exist large covariate shifts between the training and test distributions, ranging from 0.419 to 0.557. DropEdge greatly enlarges *Aug-Train*, *i.e.,* 0.627~0.851. While it fails to reduce *Aug-Test*, *e.g.,* on CMNIST (0.490→0.539) and Molbbbp (0.419→0.737). FLAG only perturbs the node features, leading to small values in *Aug-Train* and an inability to reduce *Aug-Test*. $\mathcal{G}$-Mixup significantly increases *Aug-Train* by generating OOD samples, while it cannot guarantee a decrease in *Aug-Test*. Finally, our proposed AdvCA enlarges the gap with training distribution by augmenting the environmental features. Meanwhile, the invariance of causal features significantly reduces *Aug-Test*, *e.g.,* on Motif-base (0.557→0.462), Motif-size (0.522→0.098) and CMNIST (0.490→0.408).

To verify the environmental diversity and causal invariance of AdvCA, we plot the augmented graphs in Figure 3. These augmented graphs are randomly sampled during training. More visualizations are depicted in Appendix C.2. The Motif and CMNIST graphs are displayed in the first and second rows. Figure 3 (*Left*) shows the original graphs. For Motif, the green part represents the motif-graph, whose type determines the label. While the yellow part denotes the base-graph that contains environmental features. For CMNIST, the red subgraph contains causal features while the complementary parts contain environmental features. Figure 3 (*Right*) displays the augmented samples during training. Nodes with darker colors and edges with wider lines indicate higher soft-mask values. From these visualizations, we have the following observations.

Obs4: AdvCA can achieve both environmental diversity and causal invariance. We can observe that AdvCA only perturbs the environmental features while keeping the causal parts invariant. For Motif dataset, the base-graph is a *ladder* and the motif-graph is a *house*. After augmentation, the nodes and edges of the *ladder* graph are perturbed. In contrast, the *house* part remains invariant and stable during training. The CMNIST graph also exhibits the same phenomenon. The environmental features are frequently perturbed, while the causal subgraph that determines label "2" remains invariant and stable during training. These visualizations further demonstrate that AdvCA can simultaneously guarantee environmental diversity and causal invariance.

4.4 Ablation Study (RQ3)

Adversarial augmentation v.s. Causal learning. They are two vital components that achieve environmental diversity and causal invariance. The results are depicted in Figure 4 (*Left*). "w/o Adv" and "w/o Cau" refer to AdvCA without adversarial augmentation and without causal learning, respectively. RDCA stands for a variant that replaces the adversarial augmentation in AdvCA with

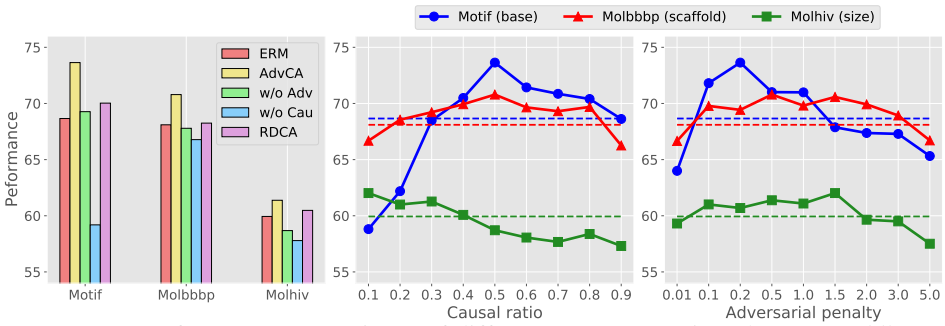

Figure 4: (*Left*): Performance comparisons of different components in AdvCA. (*Middle*): Performance over different causal ratios λ_c. (*Right*): Performance over different penalties γ.

random augmentation (*i.e.,* random masks). Compared to AdvCA, utilizing either causal learning or adversarial augmentation alone will degrade the performance. On the one hand, removing adversarial perturbations loses the invariance condition in causal learning, leading to suboptimal causal features. On the other hand, using adversarial augmentation alone will destroy the causal features, thereby impairing generalization. RDCA exceeds ERM, but is worse than AdvCA, suggesting that randomness will also encourage diversity, even if it is less effective than the adversarial strategy.

Sensitivity Analysis. The causal ratio λ_c and penalty coefficient γ determine the extent of causal features and the strength of adversarial augmentation, respectively. We also study their sensitivities. The experimental results are shown in Figure 4 (*Middle*) and (*Right*). Dashed lines denote the performance of ERM. λ_c with 0.3~0.8 performs well on Motif and Molbbbp, while Molhiv is better in 0.1~0.3. It indicates that the causal ratio is a dataset-sensitive hyper-parameter that needs careful tuning. For the penalty coefficient, the appropriate values on the three datasets range from 0.1~1.5.

5 Related Work

Invariant Causal Learning (Lu et al., 2021) exploits causal features for better generalization. IRM (Arjovsky et al., 2019) minimizes the empirical risks within different environments. Chang et al. (2020) minimize the performance gap between environment-aware and environment-agnostic predictors to discover rationales. Motivated by these efforts, DIR (Wu et al., 2022b) constructs multiple interventional environments for invariant learning. GREA (Liu et al., 2022) and CAL (Sui et al., 2022) learn causal features by challenging different environments. However, they only focus on correlation shift issues. The limited environments hinder their successes on covariate shift.

Graph Data Augmentation (Ding et al., 2022; Zhao et al., 2022; Yoo et al., 2022) enlarges the training distribution by perturbing features in graphs. Recent studies (Ding et al., 2021; Wiles et al., 2022) observe that it often outperforms other generalization efforts (Arjovsky et al., 2019; Sagawa et al., 2020). DropEdge (Rong et al., 2020) randomly removes edges, while FLAG (Kong et al., 2022) augments node features with an adversarial strategy. M-Mixup (Wang et al., 2021) interpolates graphs in semantic space. However, studies (Arjovsky et al., 2019; Lu et al., 2021) point out that causal features are the key to OOD generalization. These augmentation efforts are prone to perturb the causal features, which easily loses control of the perturbed distributions. Due to the space constraints, we put more discussions about OOD generalization in Appendix E.

6 Conclusion & Limitations

In this work, we focus on the graph generalization problem under covariate shift, which is of great need but largely unexplored. We propose a novel graph augmentation strategy, AdvCA, which is based on the principle of environmental diversity and causal invariance. Environmental diversity allows the model to explore more novel environments, thereby better generalizing to possible unseen test distributions. Causal invariance closes the distribution gap between the augmented and test data, resulting in better generalization. We make comprehensive comparisons with 14 baselines and conduct in-depth analyses and visualizations. The experimental results demonstrate that AdvCA can achieve excellent generalization ability under covariate shift. In addition, we also provide more discussions about the limitations of AdvCA and future work in Appendix F.

ETHICS STATEMENT

This paper does not involve any human subjects and does not raise any ethical concerns. We propose a graph data augmentation method to address the OOD issue of covariate shift. We conduct experiments on public datasets and validate the effectiveness on graph classification tasks. Our proposed method can be applied to practical applications, such as the prediction of molecular properties.

REPRODUCIBILITY STATEMENT

To help researchers reproduce our results, we provide detailed instructions here. For the implementation process of AdvCA, we provide the detailed algorithm in Appendix A.1. For datasets, we use publicly available datasets, and a detailed introduction is provided in Appendix A.2. For training details, we provide training settings of AdvCA and baseline settings in Appendix A.3 and A.4, respectively. Furthermore, we also provide an anonymous code link: https://anonymous.4open.science/r/AdvCA-68BF

REFERENCES

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

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

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.

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

Yongqiang Chen, Yonggang Zhang, Han Yang, Kaili Ma, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Invariance principle meets out-of-distribution generalization on graphs. *arXiv preprint arXiv:2202.05441*, 2022.

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

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 *NeurIPS Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

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

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.

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.

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

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

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.

Jean Kaddour, Aengus Lynch, Qi Liu, Matt J Kusner, and Ricardo Silva. Causal machine learning: A survey and open problems. *arXiv preprint arXiv:2206.15475*, 2022.

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*, pp. 60–69, 2022.

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*, pp. 5815–5826, 2021.

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

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

Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph rationalization with environment-based augmentations. In *SIGKDD*, pp. 1069–1078, 2022.

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

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

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*, pp. 5115–5124, 2017.

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

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

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.

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.

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

Yongduo Sui, Xiang Wang, Jiancan Wu, Xiangnan He, and Tat-Seng Chua. Causal attention for interpretable and generalizable graph classification. In *SIGKDD*, pp. 1696–1705, 2022.

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

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

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

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.

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

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

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.

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

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*, pp. 7947–7958, 2022.

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

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

Junchi Yu, Jian Liang, and Ran He. Finding diverse and predictable subgraphs for graph domain generalization. *arXiv preprint arXiv:2206.09345*, 2022.

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

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

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 IMPLEMENTATION DETAILS

A.1 ALGORITHM

We summarize the detailed implementations of AdvCA in Algorithm 1. Inspired by Suresh et al. (2021), we alternately optimize the adversarial augmenter and causal generator with the backbone model, in lines 13 and 14. We adopt the causal features for predictions in the inference stage.

Algorithm 1: Adversarial Causal Augmentation

Require: Training set $\mathcal{D}_{\text{tr}}$; Adversarial augmenter $T_{\theta_1}(\cdot)$; Causal generator $T_{\theta_2}(\cdot)$; GNN classifier $f(\cdot)$ with parameters θ; Learning rates α, β; Batch size N; Causal ratio λ_c; Penalty γ.

1: Randomly initilize θ, θ_1, θ_2
2: **while** not converge **do**
3: Sample a batch $\mathcal{B}_{\text{tr}} \leftarrow \{(g_i, y_i)\}_{i=1}^{N} \subset \mathcal{D}_{\text{tr}}$
4: **for** each $(g_i, y_i) \in \mathcal{B}_{\text{tr}}$ **do**
5: $\mathbf{M}^a_{\text{adv}}, \mathbf{M}^x_{\text{adv}} \leftarrow T_{\theta_1}(g_i)$ `// adversarial perturbations`
6: $\mathbf{M}^a_{\text{cau}}, \mathbf{M}^x_{\text{cau}} \leftarrow T_{\theta_2}(g_i)$ `// regions of causal features`
7: $\widetilde{\mathbf{M}}^a \leftarrow (\mathbf{1} - \mathbf{M}^a_{\text{cau}}) \odot \mathbf{M}^a_{\text{adv}} + \mathbf{M}^a_{\text{cau}}$ `// augment edges`
8: $\widetilde{\mathbf{M}}^x \leftarrow (\mathbf{1} - \mathbf{M}^x_{\text{cau}}) \odot \mathbf{M}^x_{\text{adv}} + \mathbf{M}^x_{\text{cau}}$ `// augment nodes`
9: $\widetilde{g}_i \leftarrow (\mathbf{A}_i \odot \widetilde{\mathbf{M}}^a, \mathbf{X}_i \odot \widetilde{\mathbf{M}}^x)$ `// augmented graph`
10: **end for**
11: Compute $\mathcal{L}_{\text{adv}} - \mathcal{L}_{\text{reg}_1}$ via Equation 10 and Equation 12
12: Compute $\mathcal{L}_{\text{cau}} + \mathcal{L}_{\text{reg}_2}$ via Equation 11 and Equation 13
13: Update parameters of adversarial augmenter via gradient ascent:
$\theta_1 \leftarrow \theta_1 + \alpha \nabla_{\theta_1}(\mathcal{L}_{\text{adv}} - \mathcal{L}_{\text{reg}_1})$
14: Update parameters of GNN and causal generator via gradient descent:
$\theta \leftarrow \theta - \beta \nabla_{\theta}(\mathcal{L}_{\text{cau}} + \mathcal{L}_{\text{reg}_2})$; $\theta_2 \leftarrow \theta_2 - \beta \nabla_{\theta_2}(\mathcal{L}_{\text{cau}} + \mathcal{L}_{\text{reg}_2})$
15: **end while**

A.2 DATASETS AND METRICS

Datasets. In this paper, we conduct experiments on graph OOD datasets (Gui et al., 2022) and OGB datasets (Hu et al., 2020), which include Motif, CMNIST, Molbbbp and Molhiv. We follow Gui et al. (2022) 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 3. Below we give a brief introduction to each dataset.

- **Motif:** It is a synthetic dataset from Spurious-Motif (Wu et al., 2022b; Sui et al., 2022). 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 (Monti et al., 2017). 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 (Wu et al., 2018). 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 (Gui et al., 2022) and use scaffold split to create training, validation and test sets. For size shift, we adopt the large-size of graphs for training and the smaller ones for validation and testing.

Metrics. We adopt classification accuracy as the metric for Motif and CMNIST. As suggested by Hu et al. (2020), we use ROC-AUC for Molhiv and Molbbbp datasets. In addition, we use $\text{GCS}(P, Q)$

Table 3: 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 4: Hyper-parameter details of AdvCA.

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
Epoch	100	100	100	100	100	100	100
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Batch size	512	512	512	32	128	32	512
Learning rate α	1e-3	1e-3	1e-3	1e-3	5e-3	1e-3	1e-2
Learning rate β	5e-3	5e-3	5e-3	1e-3	5e-3	1e-2	1e-2
Causal ratio λ_2	0.5	0.5	0.5	0.5	0.5	0.1	0.1
Adversarial penalty γ	0.2	0.2	0.2	0.5	0.5	0.5	0.5

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.

A.3 Training settings

We use the NVIDIA GeForce RTX 3090 (24GB GPU) to conduct all our experiments. To make a fair comparison, we adopt GIN (Xu et al., 2019) as the default architectures to conduct all experiments. We tune the hyper-parameters in the following ranges: α and $\beta \in \{0.01, 0.005, 0.001\}$; $\lambda_2 \in \{0.1, ..., 0.9\}$; $\gamma \in \{0.01, 0.1, 0.2, 0.5, 1.0, 1.5, 2.0, 3.0, 5.0\}$; batch size $\in \{32, 128, 256, 512\}$; hidden layers $\in \{32, 64, 128, 300\}$. The hyper-parameters of AdvCA are summarized in Table 4.

A.4 Baseline settings

For a more comprehensive comparison, we selected 14 baselines. In this section, we give a detailed introduction to the settings of these methods.

- For ERM, IRM (Arjovsky et al., 2019), GroupDRO (Sagawa et al., 2020), VREx (Krueger et al., 2021), and M-Mixup (Wang et al., 2021), we report the results from the study (Gui et al., 2022) by default and reproduce the missing results on Molbbbp.
- For DIR (Wu et al., 2022b), CAL (Sui et al., 2022), GSAT (Miao et al., 2022), DropEdge (Rong et al., 2020), GREA (Liu et al., 2022), FLAG (Kong et al., 2022) and $\mathcal{G}$-Mixup (Han et al., 2022), 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 (Li et al., 2022a) and StableGNN (Fan et al., 2021), their source codes are not publicly available. We reproduce them based on the codes of StableNet (Zhang et al., 2021).

- For RDCA in Section 4.4, it is a variant that replaces the adversarial augmentation in AdvCA 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 causal learning is consistent with AdvCA.

B 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 training set and test set $\mathcal{D}_{\text{tr}}$ and $\mathcal{D}_{\text{te}}$, they follow probability distribution functions P_{tr} and P_{te}. The process of estimating $\text{GCS}(P_{\text{tr}}, P_{\text{te}})$ is summarized in the following two steps:

- Firstly, it is intractable to directly estimate the distribution in graph space $\mathbb{G}$. Inspired by Ye et al. (2022), we can obtain the graph features and estimate the distribution in feature space $\mathbb{F}$. Specifically, given a sample, we train a binary GNN classifier f to distinguish which distribution it comes from, where $f(\cdot) = \Phi \circ h$, $h(\cdot) : \mathbb{G} \to \mathbb{F}$ is a graph encoder, and $\Phi(\cdot) : \mathbb{F} \to \{0, 1\}$ 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) (Parzen, 1962). Finally, we adopt the Monte Carlo Integration under importance sampling (Binder et al., 1993) to approximate the integrals in Definition 2.

We summarize these implementations in Algorithm 2. In lines 4 and 5, to avoid the label shift (Ye et al., 2022), we adopt sample reweighting to ensure the balance of each class.

Algorithm 2: Estimation of Graph Covariate shift

Require: Training dataset $\mathcal{D}_{\text{tr}}$ and test dataset $\mathcal{D}_{\text{te}}$; Batch size N; Loss function ℓ; GNN $f = \Phi \circ h$; Importance sampling size M; Threshold ϵ.
Ensure: Estimated covariate shift $\text{GCS}(P_{\text{tr}}, P_{\text{te}})$.

1: Initialize parameters of f
2: # Train a graph classifier
3: **while** not converge **do**
4: Sample a batch $\mathcal{B}_{\text{tr}} \leftarrow \{(g_i, y_i)\}_{i=1}^{N} \subset \mathcal{D}_{\text{tr}}$ and relabel all $y_i \leftarrow 0$
5: Sample a batch $\mathcal{B}_{\text{te}} \leftarrow \{(g_i, y_i)\}_{i=1}^{N} \subset \mathcal{D}_{\text{te}}$ and relabel all $y_i \leftarrow 1$
6: $\mathcal{B} \leftarrow \mathcal{B}_{\text{tr}} \cup \mathcal{B}_{\text{te}}$
7: **for** each $(g_i, y_i) \in \mathcal{B}$ **do**
8: Compute loss $\ell(f(g_i), y_i)$ 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 $\mathcal{F}_{\text{tr}}$ and $\mathcal{F}_{\text{te}}$ via encoder h
14: $\mathcal{F} \leftarrow \mathcal{F}_{\text{tr}} \cup \mathcal{F}_{\text{te}}$
15: Scale $\mathcal{F}$ to zero mean and unit variance
16: $\hat{\omega} \leftarrow$ fit by KDE the distribution of $\mathcal{F}$
17: Split $\mathcal{F}$ to recover the original partition $\mathcal{F}'_{\text{tr}}, \mathcal{F}'_{\text{te}}$
18: $\hat{P}_{\text{tr}}, \hat{P}_{\text{te}} \leftarrow$ fit by KDE the distributions of $\mathcal{F}'_{\text{tr}}, \mathcal{F}'_{\text{te}}$
19: # Estimate the covariate shift
20: Initialize $\text{GCS}(P_{\text{tr}}, P_{\text{te}}) \leftarrow 0$
21: **for** $t \leftarrow \{1, ..., M\}$ **do**
22: $z \leftarrow$ sample from $\hat{\omega}$
23: **if** $\hat{P}_{\text{tr}}(z) < \epsilon$ or $\hat{P}_{\text{te}}(z) < \epsilon$ **then**
24: $\text{GCS}(P_{\text{tr}}, P_{\text{te}}) \leftarrow \text{GCS}(P_{\text{tr}}, P_{\text{te}}) + |\hat{P}_{\text{tr}}(z) - \hat{P}_{\text{te}}(z)|/\hat{\omega}(z)$
25: **end if**
26: **end for**
27: $\text{GCS}(P_{\text{tr}}, P_{\text{te}}) \leftarrow \text{GCS}(P_{\text{tr}}, P_{\text{te}})/2M$

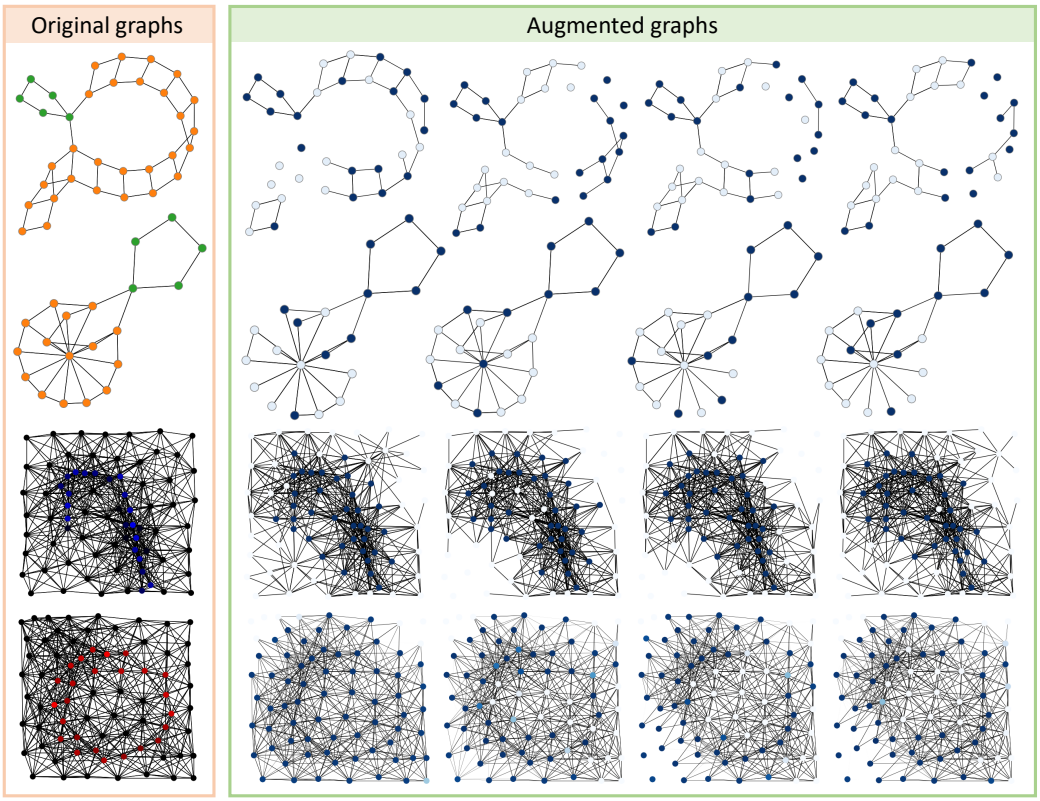

Figure 5: Visualizations of the augmented graphs via AdvCA.

C More experimental results

C.1 Results on correlation shift

Although this work focuses on the OOD issue of covariate shift, for completeness, we also evaluate the performance of AdvCA under correlation shift. Following Gui et al. (2022), 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 (Arjovsky et al., 2019), VREx (Krueger et al., 2021)), three graph generalization methods (*i.e.,* DIR (Wu et al., 2022b), CAL (Sui et al., 2022), OOD-GNN (Li et al., 2022a)) and three data augmentation methods (*i.e.,* DropEdge (Rong et al., 2020), FLAG (Kong et al., 2022), M-Mixup (Wang et al., 2021)). The experimental results are shown in Table 5. We can observe that AdvCA can also effectively alleviate the correlation shift. These results demonstrate that AdvCA learns better causal features by encouraging environmental diversity, which can effectively break spurious correlations that are hidden in the training data.

Table 5: Performance comparisons on synthetic and real-world datasets with correlation shift.

Method	Motif	CMNIST	Molhiv
ERM	$81.44_{\pm 2.54}$	$42.87_{\pm 1.37}$	$63.26_{\pm 1.25}$
IRM	$80.71_{\pm 2.81}$	$42.80_{\pm 1.62}$	$59.90_{\pm 1.17}$
VREx	$81.56_{\pm 2.14}$	$43.31_{\pm 1.03}$	$60.23_{\pm 1.60}$
DIR	$82.25_{\pm 2.15}$	$44.87_{\pm 1.56}$	$64.65_{\pm 1.34}$
CAL	$81.94_{\pm 1.20}$	$41.82_{\pm 0.85}$	$62.36_{\pm 1.42}$
OOD-GNN	$80.22_{\pm 2.28}$	$39.03_{\pm 1.24}$	$57.49_{\pm 1.08}$
DropEdge	$78.97_{\pm 3.41}$	$38.43_{\pm 1.94}$	$54.92_{\pm 1.73}$
FLAG	$80.91_{\pm 1.04}$	$43.41_{\pm 1.38}$	$66.44_{\pm 2.32}$
M-Mixup	$77.63_{\pm 1.12}$	$40.96_{\pm 1.21}$	$64.87_{\pm 1.36}$
AdvCA (ours)	$\mathbf{82.51}_{\pm \mathbf{2.81}}$	$\mathbf{49.73}_{\pm \mathbf{1.70}}$	$\mathbf{68.11}_{\pm \mathbf{1.82}}$

C.2 More visualizations

We display more visualizations of the augmented graph via AdvCA in Figure 5.

D 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 causal generator, respectively. d, d_a and d_c are the dimensions of hidden layers in the GNN backbone, adversarial augmenter and causal generator, respectively.

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

Model size. In addition to the GNN backbone model, we also introduce two small networks for adversarial augmentation and causal learning. In our implementations, the parameters of AdvCA are around twice as large as those of the original GNN model.

E MORE RELATED WORKS

OOD Generalization (Shen et al., 2021) has been widely explored. Recent studies (Ye et al., 2022; Gui et al., 2022; Wiles et al., 2022) point out that OOD falls into two specific categories: correlation shift and covariate 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 (Geirhos et al., 2020) 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. In recent years, OOD generalization on graphs is drawing widespread attention (Li et al., 2022a;b; Fan et al., 2021; Wu et al., 2022a;b; Miao et al., 2022; Yu et al., 2022; Sui et al., 2022; Liu et al., 2022; Chen et al., 2022). However, these efforts mainly focus on correlation shift. While the issue of graph covariate shift is of great need but largely unexplored.

F LIMITATION & FUTURE WORK

Although AdvCA 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.

- AdvCA performs OOD exploration through an adversarial data augmentation strategy to achieve environmental diversity. 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 Principle 1 cannot be thoroughly achieved. In future work, we will attempt to design more advanced data augmentation methods, such as graph generation-based strategies (Zhu et al., 2022), to generate more unseen and novel graph data, for pursuing Principle 1.
- For model training, we adopt adversarial training and causal learning to alternately optimize the adversarial augmenter, causal generator and backbone GNN. This training strategy may make the training process unstable, so the performance of AdvCA 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 D. Hence, in future work, we will explore how to utilize more advanced optimization methods and lightweight models to achieve the principles of environmental diversity and causal invariance.