Rethinking Invariant Graph Representation Learning without Environment Partitions

Yongqiang Chen [1] **Yatao Bian** [2] **Kaiwen Zhou** [1] **Binghui Xie** [1] **Bo Han** [3] **James Cheng** [1]

Abstract

Out-of-distribution generalization on graphs requires graph neural networks to identify the invariance among data from different environments. As the environment partitions on graphs are usually expensive to obtain, augmenting the environment information has become the *de facto* approach. However, the *usefulness* of the augmented environment information has never been verified. In this work, we found that it is fundamentally *impossible* to learn invariant graph representations by augmenting environment information without additional assumptions. Therefore, we develop a set of *minimal assumptions*, including variation sufficiency and variation consistency, for feasible invariant graph learning. Based on the assumptions, we propose **G**raph inv**A**riant **L**earning **A**ssistant (GALA), which adopts an additional assistant model that is prone to distribution shifts, to generate proxy predictions about the environments. We show that maximizing intra-class information guided by the proxy predictions provably identifies the graph invariance given the minimal assumptions. We demonstrate the usefulness of GALA with extensive experiments on 11 datasets containing various graph distribution shifts.

1. Introduction

Learning graph representations using graph neural networks (GNNs) has proven to be highly successful in tasks involving relational information (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018; Xu et al., 2018; 2019). However, it assumes that the training and test graphs are drawn from the same distribution, which is rarely the case in practice (Hu et al., 2020; Koh et al., 2021; Huang et al., 2021). The performance of GNNs could be seriously degenerated by *distribution shifts*, i.e., mismatches between the training and test distributions caused by underlying environmental factors during the data collection process (Ding et al., 2021; Ji et al., 2022; Gui et al., 2022).

To overcome the Out-of-Distribution (OOD) generalization failure, recently there has been a growing interest in incorporating the invariance principle from causality (Peters et al., 2016) into GNNs (Wu et al., 2022a;b; Chen et al., 2022a; Miao et al., 2022; Yu et al., 2022; Liu et al., 2022; Li et al., 2022; Fan et al., 2022; Yang et al., 2022; Gui et al., 2022). The rationale of these invariant graph learning approaches is to identify an underlying *invariant subgraph* of an input graph (or ego-graph of nodes (Wu et al., 2022a)), which shares an invariant correlation with the labels across multiple graph distributions that come from different environments (Wu et al., 2022a; Chen et al., 2022a). Thus, the predictions based on the invariant subgraphs can generalize to OOD graphs (Peters et al., 2016).

As the environment labels or partitions that describe the sources of distribution shifts on graphs are often expensive to obtain (Chen et al., 2022a), augmenting the environment information, such as generating new environments (Wu et al., 2022a;b; Liu et al., 2022) and inferring the environment labels (Li et al., 2022; Yang et al., 2022), has become the *de facto* approach for invariant graph learning. However, little attention has been paid to verifying the *fidelity*, or *faithfulness*, [1] of augmented environment information. For example, if the generated environments or inferred environment labels induce a higher bias or noises, it would make the learning of graph invariance even harder.

Although it looks appealing to learn *both* the environment information and the graph invariance, this approach could easily run into the "no free lunch" dilemma (Wolpert & Macready, 1997). In fact, Lin et al. (2022) found that there exist negative cases in the Euclidean regime where it is infeasible to identify the invariant features without environment partitions. When it comes to the graph regime where the OOD generalization is fundamentally more difficult (Chen et al., 2022a), it raises a challenging question:

[1]The Chinese University of Hong Kong [2]Tencent AI Lab [3]Hong Kong Baptist University. Correspondence to: Yongqiang Chen <yqchen@cse.cuhk.edu.hk>.

[1]The *fidelity* or *faithfulness* refers to whether the augmented environment information can actually improve the OOD generalization on graphs.

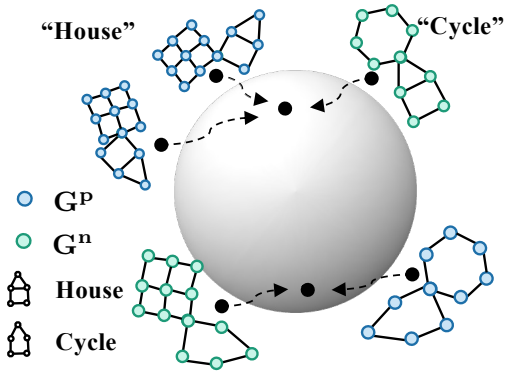


Figure 1. An illustration of GALA. Consider the task of classifying graphs according to whether there exists a "House" or "Cycle" motif. The left half and right half are predicted as "House" and "Cycle" by the environment assistant model that is prone to spurious correlations. The upper left (right) and the bottom right (left) are graphs that are classified correctly (incorrectly), denoted as $\{G^p\}$ ($\{G^n\}$) and colored in blue (green). GALA pulls graphs with the same label but from $\{G^p\}$ and $\{G^n\}$ closer at the latent representation space, hence identifies the invariant subgraph.

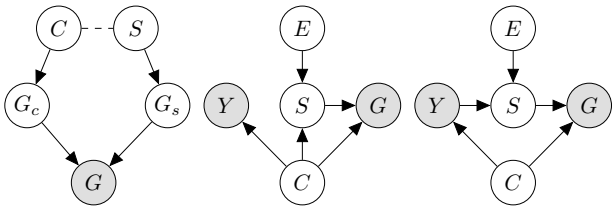
When and how could one learn graph invariance without the environment labels?

In this work, we present a theoretical study of the aforementioned problem and seek a set of *minimal assumptions* on the underlying environments such that identifying the graph invariance is possible. Based on a family of simple graph examples (Def. 3.1), we show that existing environment augmentation approaches can fail to generate faithful environments (Prop. 3.2). In fact, when the underlying environments are not sufficient to cover all the variations of the spurious subgraphs, identifying the invariant subgraph is fundamentally impossible (Theorem 3.3). Sometimes, the augmented environments can even lead to a worse OOD performance. The failure of faithful environment generation implies the necessity of *variation sufficiency* (Assumption 3.4). Moreover, even with sufficient environments, inferring faithful environment labels remains impossible (Prop. 3.5). Since invariant and spurious subgraphs can have an arbitrary degree of correlation strengths with labels, for each training environment, one can always find a corresponding environment with the same joint distribution of $P(G, Y)$ but a different invariant subgraph. In order to prevent the unidentifiable cases, we need to ensure the *variation consistency*, that is, the invariant and spurious correlation strengths should have a consistent relationship.

To resolve the OOD generalization challenge under the established assumptions, we propose **G**raph inv**A**riant **L**earning **A**ssistant (GALA), which incorporates an additional assistant model that is prone to distribution shifts, to generate proxy predictions of the environments. Different from previous approaches (Yang et al., 2022; Li et al., 2022), GALA does not require explicit environment labels but directly maximizes the intra-class mutual information among samples predicted correctly ($\{G^p\}$) and incorrectly ($\{G^p\}$) by the environment assistant model (as illustrated in Fig. 1). As $\{G^p\}$ and $\{G^p\}$ capture the variations of spurious correlations, we show that GALA is able to identify the underlying invariant subgraph under the minimal assumptions (Theorem 4.1). We conducted extensive experiments to validate the effectiveness of GALA using 11 datasets with various graph distribution shifts. Notably, GALA improves the baseline method up to 36% in realistic graph datasets.

2. Background and Preliminaries

We introduce the key concepts and background in this section, while leaving more details to Appendix A.



(a) $\mathcal{G}$-Gen. SCM (b) FIIF SCM (c) PIIF SCM

Figure 2. SCMs on graph distribution shifts (Chen et al., 2022a).

OOD generalization on graphs. This work focuses on graph classification, while the results generalize to node classification as well using the same setting as in Wu et al. (2022a). Specifically, we are given a set of graph datasets $\mathcal{D} = \{\mathcal{D}_e\}_e$ collected from multiple environments $\mathcal{E}_{\text{all}}$. Samples $(G_i^e, Y_i^e) \in \mathcal{D}^e$ from the same environment are considered as drawn independently from an identical distribution $\mathbb{P}^e$. We consider the graph generation process proposed by Chen et al. (2022a) that covers a broad case of graph distribution shifts. As shown in Fig. 2, the generation of the observed graph G and labels Y are controlled by a set of latent causal variable C and spurious variable S. C and S control the generation of the underlying invariant subgraph G_c and spurious subgraph G_s, respectively. Since S can be affected by the environment E, the correlation between Y, S and G_s can change arbitrarily when the environment changes. Besides, the latent interaction among C, S and Y can be further categorized into *Full Informative Invariant Features* (*FIIF*) when $Y \perp\!\!\!\perp S|C$ and *Partially Informative Invariant Features* (*PIIF*) when $Y \not\perp\!\!\!\perp S|C$.

To tackle the OOD generalization challenge on graphs from Fig. 2, the existing invariant graph learning approaches generically aim to identify the underlying invariant subgraph G_c to predict the label Y (Wu et al., 2022a; Chen et al., 2022a). Specifically, the goal of OOD generalization on graphs is to learn an *invariant GNN* $f := f_c \circ g$, which

is composed of two modules: a) a featurizer $g : \mathcal{G} \to \mathcal{G}_c$ that extracts the invariant subgraph G_c; b) a classifier $f_c : \mathcal{G}_c \to \mathcal{Y}$ that predicts the label Y based on the extracted G_c, where $\mathcal{G}_c$ refers to the space of subgraphs of $\mathcal{G}$. The learning objectives of f_c and g are formulated as

$$\max_{f_c,\, g} I(\widehat{G}_c; Y), \text{ s.t. } \widehat{G}_c \perp\!\!\!\perp E,\ \widehat{G}_c = g(G). \quad (1)$$

Since E is not observed, many strategies are proposed to impose the independence of $\widehat{G}_c$ and E. A common approach is to augment the environment information. For example, based on the estimated invariant subgraphs $\widehat{G}_c$ and spurious subgraphs $\widehat{G}_s$, Wu et al. (2022b); Liu et al. (2022); Wu et al. (2022a) proposed to generate new environments, while Yang et al. (2022); Li et al. (2022) proposed to infer the underlying environment labels. However, we show that it is fundamentally impossible to augment faithful environment information in Sec. 3. Yu et al. (2021); Miao et al. (2022); Yu et al. (2022); Miao et al. (2023) adopt graph information bottleneck to tackle FIIF graph shifts, and they cannot generalize to PIIF shifts. Our works focuses on PIIF shifts, as it is more challenging when there is no environment label (Lin et al., 2022). Fan et al. (2022) generalized (Lee et al., 2021) to tackle severe graph biases, i.e., when $H(S|Y) < H(C|Y)$. Chen et al. (2022a) proposed a contrastive framework to tackle both FIIF and PIFF graph shifts, but limited to $H(S|Y) > H(C|Y)$. However, in practice it is usually unknown whether $H(S|Y) < H(C|Y)$ or $H(S|Y) > H(C|Y)$ without environment information.

Invariant learning without environment labels. There are also plentiful studies in invariant learning without environment labels. Creager et al. (2021a) proposed a min-max formulation to infer the environment labels. Liu et al. (2021b) proposed a self-boosting framework based on the estimated invariant and variant features. Liu et al. (2021a); Zhang et al. (2022) proposed to infer labels based the predictions of an ERM trained model. However, Lin et al. (2022) found failure cases in Euclidean data where it is impossible to identify the invariant features without given environment labels. Moreover, as the OOD generalization on graphs is fundamentally more difficult than Euclidean data (Chen et al., 2022a), the question about the feasibility of learning invariant graph representations without environment labels remains unanswered.

3. Pitfalls of Environment Augmentation

Given only the mixed training data without environment partitions, is it possible to learn to generate effective new environment or infer the underlying environment labels?

In the discussion below, we will instantiate the problem with simple two-piece graphs (Kamath et al., 2021), which follows the PIIF distribution shifts as in Fig. 2(c).

Definition 3.1 (Two-piece graphs)**.** *Each environment is defined with two parameters, $\alpha_e, \beta_e \in [0, 1]$, and the dataset $\mathcal{D}_e$ is generated as follows:*

(a) Sample $y^e \in \{-1, 1\}$ uniformly;

(b) Generate G_c and G_s via :

$$G_c \coloneqq f_{gen}^{G_c}(Y \cdot Rad(\alpha_e)),\ G_s \coloneqq f_{gen}^{G_s}(Y \cdot Rad(\beta_e)),$$

where $f_{gen}^{G_c}, f_{gen}^{G_s}$ respectively map input $\{-1, 1\}$ to a specific graph selected from a given set, and $Rad(\alpha)$ is a random variable taking value -1 with probability α and $+1$ with probability $1 - \alpha$;

(c) Sythesize G by randomly concatenating G_c and G_s:

$$G \coloneqq f_{gen}^{G}(G_c, G_s).$$

We denote an environment e with (α_e, β_e) for simplicity. By default, different environments will have a different β_e. As β_e varies, the correlation between G_S and Y will change across different environments, while $P(Y|G_c)$ remains invariant. We use $\mathcal{E}_{\text{tr}}^{\text{mix}}$ to denote the mixed training environments as the environment labels are not available.

3.1. Pitfalls of environment generation

We begin by discussing the cases where there are few environments, which are considered in environment generation approaches (Wu et al., 2022a;b; Liu et al., 2022). When augmenting the data, they provide a set of "virtual" environments $\mathcal{D}_v = \{\mathcal{E}_v\}$ such that we can identify the invariant features by applying a OOD risk to the joint dataset with the augmented data $\mathcal{D}_{\text{tr}}^v = \{\mathcal{E}_{\text{tr}}^{\text{mix}}\} \cup \{\mathcal{E}_v\}$.

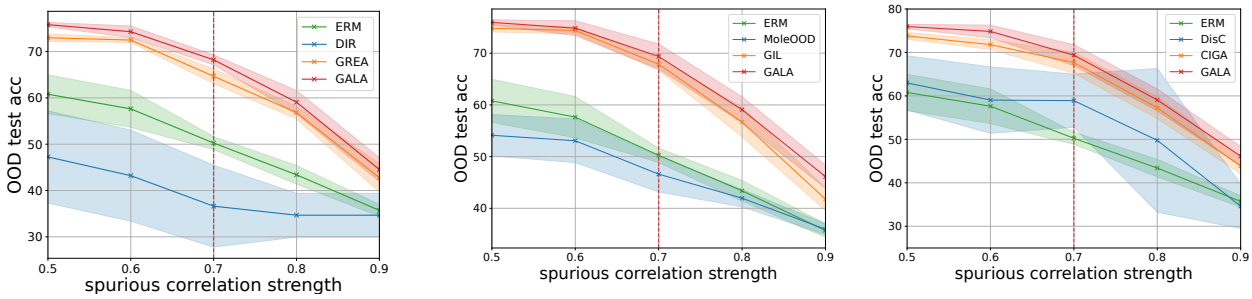
The generation of the "virtual" environments is primarily based on the estimations of the invariant and spurious subgraphs, denoted as $\widehat{G}_c$ and $\widehat{G}_s$, respectively. Wu et al. (2022b); Liu et al. (2022) proposed `DIR` and `GREA` to construct new graphs by assembling $\widehat{G}_c$ and $\widehat{G}_s$ from different graphs. Specifically, given n samples $\{G^i, Y^i\}_{i=1}^n$,[2] the new graph samples in $\mathcal{E}_v$ is generated as follows:

$$G^{i,j} = f_{\text{gen}}^{G}(\widehat{G}_c^i, \widehat{G}_s^j),\ \forall i, j \in \{1...n\},\ Y^{i,j} = Y^i,$$

which generates a new environment $\mathcal{E}_v$ with n^2 samples.

Although both `DIR` and `GREA` gain some empirical success, the faithfulness of $\mathcal{E}_v$ remains questionable, because the generation is merely based on *inaccurate* estimations of the invariant and spurious subgraphs. Specifically, when $\widehat{G}_c$ contains parts of G_s, assigning the same labels to the

[2] We slightly abuse the superscript and subscript when denoting the ith sample to avoid confusion of double superscripts or subscripts.



(a) Failures of env. generation (b) Failures of env. inferring (c) Failures of resolving env. consistency

Figure 3. Failures of finding faithful environment information. Results shown in the figure are based on the 3 class two-piece graphs (Def. 3.1), where the invariant correlation strength is fixed as 0.7 while the spurious correlation strength is varied from 0.5 to 0.7. We can find that both environment augmentation and inferring approaches suffer from severe performance decreases or even underperform ERM when the dominated correlation is not suitable for the method. In contrast, GALA maintains strong OOD performance for both cases.

generated graph is more likely to strengthen the spurious correlation between G_s and Y. Consider an extreme case where the model yields a reversed estimation, i.e., $\widehat{G}_c = G_s$ and $\widehat{G}_s = G_c$. It is possible since there could be $\mathcal{E}_{\text{tr}}^{\text{mix}}$ where the correlation between G_s and Y dominates the overall correlation between G and Y. Unfortunately, we found that the generated environment can even destroy the invariant correlations entirely.

Proposition 3.2. *Consider the two-piece graph dataset* $\mathcal{E}_{tr} = \{(\alpha, \beta_1), (\alpha, \beta_2)\}$ *with* $\alpha \geq \beta_1, \beta_2$ *(e.g.,* $\mathcal{E}_{tr} = \{(0.25, 0.1), (0.25, 0.2)\}$*), and its corresponding mixed environment* $\mathcal{E}_{tr}^{mix} = \{(\alpha, (\beta_1 + \beta_2)/2)\}$ *(e.g.,* $\mathcal{E}_{tr}^{mix} = \{(0.25, 0.15)\}$*). It holds that the augmented environment* $\mathcal{E}_v$ *is also a two-piece graph dataset with*

$$\mathcal{E}_v = \{(0.5, (\beta_1 + \beta_2)/2)\} \text{ (e.g., } \mathcal{E}_v = \{(0.5, 0.15)\}).$$

The proof is given in Appendix C.1. This also extends to the adversarial augmentation (Wu et al., 2022a; Yu et al., 2022), which will destroy the actual $\widehat{G}_c$. We verified the failures of environment generation of DIR and GREA in Fig. 3(a), where both DIR and GREA suffer from severe performance decrease when the spurious correlation dominates.

In fact, when the underlying environments are insufficient to differentiate the variations of the spurious features, it is fundamentally impossible to identify the underlying invariant graph from the spurious subgraph.

Theorem 3.3. *(Variation insufficiency) Given the same graph generation process as Fig. 2, when there exists spurious subgraph* G_s *s.t.* $P^{e_1}(Y|G_s) = P^{e_2}(Y|G_s)$ *for any* $e_1, e_2 \in \mathcal{E}_{tr}$*, where* $P^e(Y|G_s)$ *is the conditional distribution* $P(Y|G_s)$ *under environment* $e \in \mathcal{E}_{all}$*, it is impossible for any learning algorithm to identify* G_c*.*

The proof is given in Appendix C.2. Theorem 3.3 implies a fundamental requirement on $\mathcal{E}_{\text{tr}}$ that the environments therein should cover the variations of each spurious feature.

Assumption 3.4. *(Variation sufficiency) Given the same graph generation process as in Fig. 2, for any* G_s*, there exists two environments* $e_1, e_2 \in \mathcal{E}_{tr}$*, such that* $P^{e_1}(Y|G_s) \neq P^{e_2}(Y|G_s)$*, and* $P^{e_1}(Y|G_c) = P^{e_2}(Y|G_c)$*.*

Assumption 3.4 aligns with the definition of invariance (Kamath et al., 2021; Chen et al., 2022a) that the invariant subgraph G_c is expected to satisfy $P^{e_1}(Y|G_c) = P^{e_2}(Y|G_c)$ for $e_1, e_2 \in \mathcal{E}_{\text{all}}$. If there exist spurious subgraphs G_s also satisfy the invariance condition, then it is impossible to tell G_c from G_s even with environment labels.

3.2. Pitfalls of environment inferring

Although environment sufficiency (Assumption 3.4) relieves the need for generating new environments, is it possible to infer the underlying environment labels via approaches such as MoleOOD (Yang et al., 2022) and GIL (Li et al., 2022), to facilitate invariant graph learning?

Considering the two-piece graph examples $\mathcal{E}_{\text{tr}} = \{(0.2, 0.1), (0.2, 0.3)\}$, when given the underlying environment labels, it is easy to identify the invariant subgraphs from spurious subgraphs. However, when there is no environment label, we have $\mathcal{E}_{\text{tr}}^{\text{mix}} = \{(0.2, 0.2)\}$, where $P(Y|G_c) = P(Y|G_s)$. The identifiability of G_s is *ill-posed*, as it does not affect the $\mathcal{E}_{\text{tr}}^{\text{mix}}$ even if we switch G_c and G_s. More formally, consider the environment mixed from two two-piece graph environments $\{(\alpha, \beta_1)\}$ and $\{(\alpha, \beta_2)\}$, then we have $\mathcal{E}_{\text{tr}}^{\text{mix}} = \{(\alpha, (\beta_1 + \beta_2)/2\}$. For each $\mathcal{E}_{\text{tr}}^{\text{mix}}$, we can also find a corresponding $\mathcal{E}_{\text{tr}}^{\text{mix}'} = \{((\beta_1' + \beta_1')/2, \alpha')\}$ with $\{(\beta_1', \alpha')\}$ and $\{(\beta_2', \alpha')\}$. Then, let

$$\alpha = (\beta_1' + \beta_1')/2 = \alpha' = (\beta_1 + \beta_2)/2. \tag{2}$$

We now obtain $\mathcal{E}_{\text{tr}}^{\text{mix}}$ and $\mathcal{E}_{\text{tr}}^{\text{mix}'}$ which share the identical distribution $P(Y, G)$ while the underlying G_c can be different. More generally, we have the following proposition.

Proposition 3.5. *There exist* 2 *two-piece graph training*

environments $\mathcal{E}_{tr}$ and $\mathcal{E}_{tr}{}'$, whose mixed training environments are the same, such that any learning algorithm will fail to capture the invariance of at least one of the training environments.

The proof is given in Appendix C.3. Figure 3(b) shows that both MoleOOD and GIL fail to infer faithful environment labels or even underperform ERM. The failure implies that, whenever it allows the existence of identical $\mathcal{E}_{\text{tr}}^{\text{mix}}$s by mixing different environments, the OOD generalization on graphs is impossible. Therefore, we seek an additional assumption that excludes the unidentifiability case. More specifically, we propose to constrain the relationship between α (i.e., $H(Y|G_c)$) and β_e (i.e., $H(Y|G_s)$).

Assumption 3.6. *(Variation consistency) For all environments in $\mathcal{E}_{tr}$, either $H(C|Y) > H(S|Y)$ holds or $H(C|Y) < H(S|Y)$ holds.*

Intuitively, Assumption 3.6 imposes the consistency requirement on the correlation strengths between invariant and spurious subgraphs with labels. For two-piece graphs with consistent variations, mixing up the environments will yield a new environment with the same variation strength relationships. Thus, Assumption 3.6 gets rid of the previous unidentifiability cases, as which require the existence of environments with different variation relationships. The requirement on variation consistency also resembles the "no free lunch" theorem (Wolpert & Macready, 1997). Otherwise, we can always find some environment that assembles the unidentifiability case and prevents OOD generalization.

Corollary 3.7. *(No Free Lunch for Graph OOD) Without Assumption 3.6, there does not exist a learning algorithm that captures the invariance of all of the two-piece graph environments.*

The proof is given in Appendix C.4. Different from our work, Lin et al. (2022) proposed to incorporate additional auxiliary information that satisfies certain requirements to mitigate the unidentifiability case. However, such auxiliary information is often unavailable and expensive to obtain on graphs. More importantly, the requirements are also unverifiable without more assumptions, which motivates us to consider the relaxed case implied by Assumption 3.6.

3.3. Challenges of environment augmentation

Assumption 3.4 and Assumption 3.6 establish the minimal conditions for identifying the underlying invariant subgraphs. However, it also raises new challenges, as shown in Table. 1. Chen et al. (2022a) proposed CIGA to maximize the intra-class mutual information of the estimated invariant subgraphs to tackle the case when $H(C|Y) < H(S|Y)$. While for the case when $H(S|Y) < H(C|Y)$, Fan et al. (2022) proposed DisC that adopts GCE loss (Lee et al., 2021) to extract the spurious subgraph with a larger learning step size such that the left subgraph is invariant. However, both CIGA and DisC can fail when there is no prior knowledge about the relations between $H(C|Y)$ and $H(S|Y)$. We verify the failures of DisC and CIGA in Fig. 3(c). The failure thus raises a challenging question:

Table 1. Assumption 3.6 raises new challenges for environment augmentation, where no existing works could handle both cases.

	$H(S\|Y) < H(C\|Y)$	$H(S\|Y) > H(C\|Y)$
DisC	✓	✗
CIGA	✗	✓
GALA (Ours)	✓	✓

Given the established minimal assumptions, is there a unified framework that tackles both cases when $H(C|Y) < H(S|Y)$ and $H(C|Y) > H(S|Y)$?

4. Learning Invariant Graph Representations with Environment Assistant

We provide affirmative answers to the previous question by proposing a new framework, GALA: **G**raph inv**A**riant **L**earning **A**ssistant, which adopts an assistant model to provide faithful information about the underlying environments.

4.1. Learning with An Environment Assistant

Intuitively, a straightforward approach to tackle the aforementioned challenge is to extend the framework of either DisC (Fan et al., 2022) or CIGA (Chen et al., 2022a) to resolve the other case. As DisC always destroys the first learned features which tends to be more difficult to extend, we are thus motivated to extend the framework of CIGA to resolve the case when $H(S|Y) < H(C|Y)$.

Understanding the success and failure of CIGA. The principle of CIGA lies in maximizing the mutual information of the estimated invariant subgraphs from the same class, i.e.,

$$\max_{f_c,g} I(\widehat{G}_c;Y),\ \text{s.t.}\ \widehat{G}_c \in \underset{\widehat{G}_c=g(G),|\widehat{G}_c|\le s_c}{\arg\max} I(\widehat{G}_c;\widetilde{G}_c|Y), \tag{3}$$

where $\widetilde{G}_c = g(\widetilde{G})$ and $\widetilde{G} \sim \mathbb{P}(G|Y)$, i.e., $\widetilde{G}$ is sampled from training graphs that share the same label Y as $\widehat{G}$. The key reason for the success of Eq. 3 is that, given the data generation process as in Fig. 2 and the same C, the underlying invariant subgraph G_c maximizes the mutual information of subgraphs from the two environments, i.e., $\forall e_1, e_2 \in \mathcal{E}_{\text{all}}$,

$$G_c^{e_1} \in \arg\max_{\widehat{G}_c^{e_1}} I(\widehat{G}_c^{e_1};\widehat{G}_c^{e_2}|C), \tag{4}$$

where $\widehat{G}_c^{e_1}$ and $\widehat{G}_c^{e_2}$ are the estimated invariant subgraphs corresponding to the same latent causal variable $C = c$

under the two environments e_1, e_2, respectively. Since C is not directly observable, CIGA adopts Y as a proxy for C, as when $H(S|Y) > H(C|Y)$, G_c maximizes $I(\widehat{G}_c^{e_1}; \widehat{G}_c^{e_2}|Y)$ and thus $I(\widehat{G}_c; \widetilde{G}_c|Y)$. However, when $H(S|Y) < H(C|Y)$, the proxy no longer holds. Given the absence of E, simply maximizing intra-class mutual information favors the spurious subgraph G_s instead, i.e.,

$$G_s \in \arg\max_{\widehat{G}_c} I(\widehat{G}_c; \widetilde{G}_c|Y). \tag{5}$$

Invalidating spuriousness dominance. To mitigate the issue, we are motivated to find a new proxy that samples $\widetilde{G}_c$ for Eq. 5, while preserving only the G_c as the solution.

To begin with, we first consider resolving the failure case of CIGA. When $H(S|Y) < H(C|Y)$, although the correlation between G_s and Y dominates the intra-class mutual information, Assumption 3.4 implies that there exists a subset of training data where $P(Y|G_s)$ varies, while $P(Y|G_c)$ remains invariant. Therefore, the dominance of spurious correlations no longer holds in samples from the subset. Incorporating the subset into Eq. 3 as $\widetilde{G}_c$ can further invalidate the dominance of G_s in Eq. 3. Denote subset as $\{\widehat{G}_c^n\}$, then

$$G_c \in \arg\max_{\widehat{G}_c^p} I(\widehat{G}_c^p; \widetilde{G}_c^n|Y), \tag{6}$$

where $\widehat{G}_c^p \in \{\widehat{G}_c^p\}$ is sampled from the subset $\{\widehat{G}_c^p\}$ dominated by spurious correlations, while $\widetilde{G}_c^n \in \{\widehat{G}_c^n\}$ is sampled from the subset $\{\widehat{G}_c^n\}$ where spurious correlation no long dominates, or dominated by invariant correlations. We prove the effectiveness of Eq. 6 in Theorem 4.1.

Environment assistant model A. The remaining challenge is to find the desired subsets $\{\widehat{G}_c^p\}$ and $\{\widetilde{G}_c^n\}$. Motivated by the success in tackling spuriousness dominated OOD generalization via learning from a biased predictors (Nam et al., 2020; Lee et al., 2021; Liu et al., 2021a; Zhang et al., 2022), we propose to incorporate an assistant model A that is prone to spurious correlations. Training A with ERM using the spuriousness dominated data enables A learns spurious correlations, and hence we can identify the subsets where the spurious correlations hold or shifts, according to whether the predictions of A are accurate or not. More formally, we have

$$\begin{aligned} \{\widehat{G}_c^p\} &= \{g(G_i^p)|A(G_i^p) = Y_i\}, \\ \{\widetilde{G}_c^n\} &= \{g(G_i^n)|A(G_i^n) \neq Y_i\}, \end{aligned} \tag{7}$$

where $A = \arg\max_{\widehat{A}} I(\widehat{A}(G); Y)$.

Reducing to invariance dominance case. Although Eq. 6 resolves the spuriousness dominance case, can it still preserves G_c as the only solution when $H(S|Y) > H(C|Y)$?

Algorithm 1 GALA: **G**raph inv**A**riant **L**earning **A**ssistant

Input: Training data $\mathcal{D}_{\text{tr}}$; environment assistant A; featurizer g; classifier f_c; length of maximum training epochs e; batch size b;
Initialize environment assistant A;
for $p \in [1, \ldots, e]$ **do**
  Sample a batch of data $\{G_i, Y_i\}_{i=1}^b$ from $\mathcal{D}_{\text{tr}}$;
  Obtain Environment Assistant predictions $\{\hat{y}_i^e\}_{i=1}^b$;
  for each sample $G_i, y_i \in \{G_i, Y_i\}_{i=1}^b$ **do**
    Find *postive* graphs with same y_i and different $\hat{y}_i^e$;
    Find *negative* graphs with different y_i but same environment assistant prediction $\hat{y}_i^e$;
    Calculate GALA risk via Eq. 10;
    Update f_c, g via gradients from GALA risk;
  end for
end for
return final model $f_c \circ g$;

Fortunately, we find positive answers. Considering training A with ERM using the invariance dominated data, A will learn both invariant correlations and spurious correlations. Therefore, $\{\widetilde{G}_c^n\}$ switches to the subset that is dominated by spurious correlations, while $\{\widehat{G}_c^p\}$ switches to the subset dominated by invariant correlations. Then, Eq. 6 establishes a lower bound for the intra-class mutual information, i.e.,

$$I(\widehat{G}_c^p; \widetilde{G}_c^n|Y) \leq I(\widehat{G}_c; \widetilde{G}_c|Y), \tag{8}$$

where $\widehat{G}_c^p \in \{\widehat{G}_c^p\}, \widetilde{G}_c^n \in \{\widetilde{G}_c^n\}$, and $\widehat{G}_c, \widetilde{G}_c$ are the estimated subgraphs for two random samples with the same label. The equality is achieved by taking G_c as the solution for the featurizer g.

4.2. Theoretical analysis

In the following theorem, we show that the derived objective from Sec. 4.1 can identify the underlying invariant subgraph and yields an invariant GNN defined in Sec. 2.

Theorem 4.1. *Given i) the same data generation process as in Fig. 2; ii) $\mathcal{D}_{\text{tr}}$ that satisfies variation sufficiency (Assumption 3.4) and variation consistency (Assumption 3.6); iii) $\{G^p\}$ and $\{G^n\}$ are distinct subsets of $\mathcal{D}_{\text{tr}}$ such that $I(G_s^n; G_s^p|Y) = 0$, $\forall G_s^p = \arg\max_{\widehat{G}_s^p} I(\widehat{G}_s^p; Y)$ under $\{G^p\}$, and $\forall G_s^n = \arg\max_{\widetilde{G}_s^n} I(\widetilde{G}_s^n; Y)$ under $\{G^n\}$; suppose $|G_c| = s_c$, $\forall G_c$, resolving the following GALA objective elicits an invariant GNN defined via Eq. 13,*

$$\max_{f_c, g} I(\widehat{G}_c; Y), \text{ s.t. } g \in \arg\max_{\hat{g}, |\widehat{G}_c^p| \leq s_c} I(\widehat{G}_c^p; \widetilde{G}_c^n|Y), \tag{9}$$

where $\widetilde{G}_c^p \in \{\widehat{G}_c^p = g(G^p)\}$ and $\widetilde{G}_c^n \in \{\widetilde{G}_c^n = g(G^n)\}$ are the estimated invariant subgraphs via g from $\{G^p\}$ and $\{G^n\}$, respectively.

The proof is given in Appendix C.5. Essentially, assumption iii) in Theorem 4.1 is an implication of the variation sufficiency (Assumption 3.4). When given the distinct subsets $\{G^p\}$ and $\{G^n\}$ with different relations of $H(C|Y)$ and $H(S|Y)$, since $H(C|Y)$ remains invariant across different subsets, the variation happens to be the spurious correlations between S and Y. By differentiating spurious correlations into distinct subsets, maximizing the intra-class mutual information helps identify the true invariance. The fundamental rationale of GALA relies on the commutative law of mutual information.

4.3. Practical implementation

Environment assistant implementation. Theorem 4.1 shows the effectiveness of GALA when given proper subsets of $\{G^p\}$ and $\{G^n\}$. In practice, we can implement the environment assistant in multiple forms. As discussed in Sec. 4.1, ERM trained model can serve as a reliable proxy. Since ERM tends to learn the first dominant features, when $H(S|Y) < H(C|Y)$, ERM will first learn to extract spurious subgraphs G_s to make predictions. Therefore, we can obtain $\{G^p\}$ by finding samples where ERM correctly predicts the labels, and we obtain $\{G^n\}$ for samples where ERM predicts incorrect labels. In addition to direct label predictions, the clustering predictions of the hidden representations yielded by environment assistant models can also be used for sampling $\{G^p\}$ and $\{G^n\}$ (Zhang et al., 2022). Besides, we can also incorporate models that are easier to overfit to the first dominant features to better differentiate $\{G^p\}$ from $\{G^n\}$. We provide more discussions about the implementations of environment assistant in Appendix D.

Objective implemention. As the estimation of mutual information could be highly expensive (van den Oord et al., 2018; Belghazi et al., 2018), inspired by Chen et al. (2022a), we adopt the contrastive learning to approximates the mutual information between subgraphs in Eq. 9 (Khosla et al., 2020; Chopra et al., 2005; Salakhutdinov & Hinton, 2007; van den Oord et al., 2018; Belghazi et al., 2018):

$$I(\widehat{G}_c^p;\widetilde{G}_c^n|Y) \approx \mathbb{E}_{\substack{\{\widehat{G}_c^p,\widetilde{G}_c^n\}\sim\mathbb{P}_g(G|\mathcal{Y}=Y)\\ \{G_c^i\}_{i=1}^M\sim\mathbb{P}_g(G|\mathcal{Y}\neq Y)}} \log \frac{e^{\phi(h_{\widehat{G}_c^p},h_{\widetilde{G}_c^n})}}{e^{\phi(h_{\widehat{G}_c^p},h_{\widetilde{G}_c^n})}+\sum_{i=1}^M e^{\phi(h_{\widehat{G}_c},h_{G_c^i})}}, \tag{10}$$

where $(\widehat{G}_c^p,\widetilde{G}_c^n)$ are subgraphs extracted by g from $\{G^p\},\{G^n\}$ that share the same label, $\{G_c^i\}_{i=1}^M$ are subgraphs extracted by g from G that has a different label. $h_{\widehat{G}_c^p}, h_{\widetilde{G}_c^n}, h_{G_c^i}$ are the graph presentations of the extracted subgraphs, and ϕ measures the similarity between graph representations. As $M\to\infty$, Eq. 10 approximates $I(\widehat{G}_c^p;\widetilde{G}_c^n|Y)$ (Ahmad & Lin, 1976; Kandasamy et al., 2015; Wang & Isola, 2020).

Table 2. OOD generalization performance under various invariant and spurious correlation degrees in the two-piece motif datasets. Each dataset is generated from a variation of two-piece graph model, denoted as $\{a, b\}$, where a refers to the invariant correlation strength and b refers to the spurious correlation strength.

DATASETS	$\{0.8, 0.6\}$	$\{0.8, 0.7\}$	$\{0.8, 0.9\}$	$\{0.7, 0.9\}$
ERM	66.91 (2.55)	62.55 (2.38)	41.90 (1.74)	36.02 (1.55)
IRM	70.06 (1.22)	61.78 (1.05)	42.63 (2.22)	35.85 (0.84)
V-REX	69.32 (1.88)	65.10 (2.46)	43.42 (1.90)	35.17 (1.85)
EIIL	66.10 (3.47)	61.40 (1.71)	40.47 (2.15)	36.48 (0.90)
IB-IRM	64.89 (2.23)	61.76 (1.81)	42.25 (2.13)	37.31 (1.73)
GREA	79.39 (1.25)	75.88 (0.83)	54.19 (4.06)	42.59 (2.27)
GSAT	79.87 (1.05)	76.68 (1.65)	53.14 (1.80)	40.99 (0.92)
MOLEOOD	66.61 (1.72)	52.75 (4.73)	41.73 (0.81)	34.25 (2.45)
GIL	80.72 (0.75)	77.87 (0.75)	54.48 (2.09)	42.18 (2.09)
DISC	63.49 (9.56)	59.64 (4.96)	49.25 (15.7)	38.90 (9.67)
CIGAV1	80.39 (0.80)	78.11 (0.89)	54.52 (1.54)	44.34 (6.03)
GALA	**82.21 (1.00)**	**80.99 (0.76)**	**57.00 (1.57)**	**45.05 (2.78)**
ORACLE (IID)	82.37 (0.77)	83.25 (0.86)	82.61 (0.48)	76.47 (0.66)

For each G^p, in addition to sampling only positive samples from $\{G^n\}$, we can also find hard negative samples based on environment assistant predictions, which have a different ground truth labels Y but being predicted as the same as G^p (Zhang et al., 2022). The detailed algorithm description of GALA is shown as in Algorithm 1.

5. Experimental Evaluation

We evaluated GALA with both synthetic and realistic graph distribution shifts. Specifically, we are interested in the following two questions: (a) Can GALA improve over the state-of-the-art invariant graph learning methods when the spurious subgraph has a stronger correlation with the labels? (b) Will GALA affect the performance when the invariant correlations are stronger?

5.1. Datasets

We prepared both synthetic and realistic graph datasets containing various distribution shifts. We will briefly introduce each dataset. More details are given in Appendix E.1.

Two-Piece Motif. We adopted BA-2motifs (Luo et al., 2020) to implement 4 variants of 3-class two-piece graph (Def. 3.1) datasets. The datasets contain different relationships of $H(C|Y)$ and $H(S|Y)$ by controlling the α and β in the mixed environment. We consider 4 cases of $\alpha-\beta$, ranging from $\{+0.2, +0.1, -0.1, -0.2\}$.

Realistic datasets. We also adopted datasets containing various realistic graph distribution shifts to comprehensively evaluate the OOD performance of GALA. We adopted 3 dataset from DrugOOD benchmark (Ji et al., 2022), including splits using Assay, Scaffold, and Size splits from the EC50 category (denoted as **EC50-***). We also adopted graphs converted from the ColoredMNIST

Table 3. OOD generalization performance under realistic graph distribution shifts.

DATASETS	EC50-ASSAY	EC50-SCA	EC50-SIZE	CMNIST-SP	GRAPH-SST2	GRAPH-SST5	TWITTER	AVG (RANK)†
ERM	76.42 (1.59)	64.56 (1.25)	62.79 (1.15)	21.56 (5.38)	81.54 (1.13)	42.62 (2.54)	59.34 (1.13)	58.40 (7.14)
IRM	76.51 (1.89)	64.82 (0.55)	63.23 (0.56)	23.29 (7.82)	82.22 (0.88)	42.77 (1.26)	60.42 (1.06)	59.04 (5.29)
V-REX	76.73 (2.26)	62.83 (1.20)	59.27 (1.65)	24.62 (8.75)	79.27 (1.95)	42.48 (1.67)	60.50 (2.05)	57.96 (7.57)
EIIL	76.96 (0.25)	64.95 (1.12)	62.65 (1.88)	24.55 (13.3)	80.70 (1.21)	43.79 (1.19)	60.15 (1.44)	59.11 (5.00)
IB-IRM	76.72 (0.98)	64.43 (0.85)	64.10 (0.61)	13.06 (1.97)	82.12 (1.43)	43.02 (1.94)	60.80 (2.50)	57.75 (5.57)
GREA	73.47 (2.48)	62.00 (1.36)	61.88 (1.00)	18.64 (6.44)	80.20 (0.56)	43.29 (0.85)	59.92 (1.48)	57.06 (8.57)
GSAT	66.87 (8.70)	63.31 (1.17)	61.70 (1.48)	12.77 (2.00)	80.55 (2.18)	43.24 (0.61)	60.13 (1.51)	55.51 (8.57)
DISC	65.40 (5.34)	54.97 (3.86)	56.79 (2.56)	54.07 (15.3)	79.10 (2.09)	40.67 (1.19)	57.89 (2.02)	58.41 (10.1)
MOLEOOD	61.94 (1.90)	59.53 (3.47)	56.08 (1.45)	39.55 (4.35)	80.78 (1.13)	40.36 (1.85)	59.26 (1.67)	56.79 (9.71)
GIL	72.13 (4.70)	63.05 (1.04)	62.08 (1.60)	18.04 (4.39)	82.04 (0.97)	43.30 (1.24)	61.78 (1.66)	57.49 (6.29)
CIGAV1	78.46 (0.45)	**66.05 (1.29)**	65.35 (0.88)	23.66 (8.65)	81.97 (0.87)	44.05 (1.48)	61.15 (0.72)	60.10 (3.00)
GALA	**79.24 (1.36)**	66.00 (1.86)	**66.01 (0.84)**	**59.16 (3.64)**	**82.50 (0.86)**	**44.88 (1.02)**	**62.45 (0.62)**	**65.75 (1.14)**
ORACLE (IID)	85.18 (1.13)	83.02 (0.77)	86.07 (0.33)	67.25 (0.76)	91.40 (0.26)	47.96 (1.34)	63.66 (0.79)	

†Averaged rank is also reported in the parentheses because of dataset heterogeneity. Lower rank is better.

dataset of IRM (Arjovsky et al., 2019) using the algorithm from Knyazev et al. (2019) that contains distribution shifts in node attributes (denoted as **CMNIST-sp**). In addition, we adopted **Graph-SST2**, **Graph-SST5** and **Twitter** (Yuan et al., 2020) and inject degree biases. The training set in Graph-SST2 and Graph-SST5 contain graphs with smaller average degrees than the test set, while the training set in Twitter contains a larger average degree.

5.2. Baselines and experiment setup

We adopted the state-of-the-art OOD methods from the Euclidean regime, including IRMv1 (Arjovsky et al., 2019), VREx (Krueger et al., 2021), EIIL (Creager et al., 2021b) and IB-IRM (Ahuja et al., 2021), as well as from the graph regime, including GREA (Liu et al., 2022), GSAT (Miao et al., 2022), MoleOOD (Yang et al., 2022), GIL (Li et al., 2022), DisC (Fan et al., 2022) and CIGAv1 (Chen et al., 2022a). We excluded DIR (Wu et al., 2022b) and GIB (Yu et al., 2021) as GREA and GSAT are their sophisticated variants. We also excluded CIGAv2 (Chen et al., 2022a) as GALA focuses on improving the contrastive sampling via environment assistant for the objective in CIGAv1.

All methods adopted the same GIN backbone (Xu et al., 2019) as the graph encoder, as well as an identical optimization protocol for fair comparisons. We tuned the hyperparmeters following the recommended settings in previous works. More details are given in Appendix E.2.

5.3. Experimental results and analysis

Controlled study with two-piece motif. The results in Two-Piece Motif datasets are reported in Table 2. All the previous environment augmentation approaches fail either in datasets where the invariant correlations dominate or where the spurious correlations dominate, which is consistent with our discussions in Sec. 3. In particular, GREA, CIGA and GIL achieve high performance when invariant correlation strength is stronger, but suffer great performance decrease when the spurious correlations are stronger. Although DisC is expected to succeed when spurious correlations dominate, DisC fails to outperform others because its excessive destruction of the learned information. MoleOOD could also yield degraded performance, which is likely caused by the failures of inferring reliable environment labels. In contrast, GALA achieves consistently high performance under *both* cases and improves CIGAv1 up to 3%, which validates our theoretical results in Sec. 4.2.

OOD generalization performance in realistic graphs. The results in realistic datasets are reported in Table 3. Aligning with our previous discussion, DisC and MoleOOD succeed in CMNIST-sp as the spurious correlations in node features are much stronger than the graph digit shape. However, GALA achieves an even higher performance in CMNIST-sp and improves CIGA by 36%. As for the other datasets, since there is no prior knowledge about the dominance of invariant and spurious features, they are particularly challenging for OOD generalization. The results show that all the previous methods can suffer performance degradation in some datasets. In contrast, owning to the theoretical power of generalizing to both spurious and invariant correlation dominated cases, GALA continues to succeed and achieves the state-of-the-art performance under the challenging realistic graph distribution shifts.

6. Conclusions

We conducted a retrospective study on the faithfulness of the augmented environment information for OOD generalization on graphs. By showing the impossiblility results of the problems considered in existing approaches, we developed a set of minimal assumptions for feasible invariant graph learning. Built upon the assumptions, we further proposed GALA to learn the invariant graph representations guided by an environment assistant model. Extensive experiments with 11 datasets verified the superiority of GALA.

References

Ahmad, I. and Lin, P.-E. A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.). *IEEE Transactions on Information Theory*, 22(3):372–375, 1976. (Cited on page 7)

Ahuja, K., Caballero, E., Zhang, D., Gagnon-Audet, J.-C., Bengio, Y., Mitliagkas, I., and Rish, I. Invariance principle meets information bottleneck for out-of-distribution generalization. In *Advances in Neural Information Processing Systems*, 2021. (Cited on pages 8 and 23)

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. (Cited on pages 8, 22 and 23)

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International Conference on Machine Learning*, volume 80, pp. 531–540, 10–15 Jul 2018. (Cited on page 7)

Bevilacqua, B., Zhou, Y., and Ribeiro, B. Size-invariant graph representations for graph classification extrapolations. In *International Conference on Machine Learning*, pp. 837–851, 2021. (Cited on page 14)

Chen, Y., Zhang, Y., Bian, Y., Yang, H., Ma, K., Xie, B., Liu, T., Han, B., and Cheng, J. Learning causally invariant representations for out-of-distribution generalization on graphs. In *Advances in Neural Information Processing Systems*, 2022a. (Cited on pages 1, 2, 3, 4, 5, 7, 8, 13, 14, 15, 17, 23 and 24)

Chen, Y., Zhou, K., Bian, Y., Xie, B., Ma, K., Zhang, Y., Yang, H., Han, B., and Cheng, J. Pareto invariant risk minimization. *arXiv preprint*, arXiv:2206.07766, 2022b. (Cited on page 22)

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pp. 539–546, 2005. (Cited on page 7)

Creager, E., Jacobsen, J., and Zemel, R. S. Environment inference for invariant learning. In *International Conference on Machine Learning*, pp. 2189–2200, 2021a. (Cited on pages 3, 15 and 23)

Creager, E., Jacobsen, J., and Zemel, R. S. Environment inference for invariant learning. In *International Conference on Machine Learning*, volume 139, pp. 2189–2200, 2021b. (Cited on pages 8 and 23)

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019. (Cited on page 22)

Ding, M., Kong, K., Chen, J., Kirchenbauer, J., Goldblum, M., Wipf, D., Huang, F., and Goldstein, T. A closer look at distribution shifts and out-of-distribution generalization on graphs. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. (Cited on page 1)

Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., and Xu, K. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Annual Meeting of the Association for Computational Linguistics*, pp. 49–54, 2014. (Cited on page 22)

Fan, S., Wang, X., Mo, Y., Shi, C., and Tang, J. Debiasing graph neural networks via learning disentangled causal substructure. In *Advances in Neural Information Processing Systems*, 2022. (Cited on pages 1, 3, 5, 8, 14, 15 and 23)

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. (Cited on page 24)

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M. E., Schmitz, M., and Zettlemoyer, L. Allennlp: A deep semantic natural language processing platform. *arXiv preprint*, arXiv:1803.07640, 2018. (Cited on page 22)

Gui, S., Li, X., Wang, L., and Ji, S. GOOD: A graph out-of-distribution benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. (Cited on page 1)

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. (Cited on pages 22 and 23)

Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017. (Cited on page 1)

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020. (Cited on page 1)

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y. H., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. (Cited on page 1)

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, volume 37, pp. 448–456, 2015. (Cited on page 22)

Ji, Y., Zhang, L., Wu, J., Wu, B., Huang, L.-K., Xu, T., Rong, Y., Li, L., Ren, J., Xue, D., Lai, H., Xu, S., Feng, J., Liu, W., Luo, P., Zhou, S., Huang, J., Zhao, P., and Bian, Y. DrugOOD: Out-of-Distribution (OOD) Dataset Curator and Benchmark for AI-aided Drug Discovery – A Focus on Affinity Prediction Problems with Noise Annotations. *arXiv preprint*, arXiv:2201.09637, 2022. (Cited on pages 1, 7, 22 and 23)

Jin, W., Zhao, T., Ding, J., Liu, Y., Tang, J., and Shah, N. Empowering graph representation learning with test-time graph transformation. *arXiv preprint*, arXiv:2210.03561, 2022. (Cited on page 14)

Kamath, P., Tangella, A., Sutherland, D., and Srebro, N. Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pp. 4069–4077, 2021. (Cited on pages 3 and 4)

Kamhoua, B. F., Zhang, L., Chen, Y., Yang, H., KAILI, M., Han, B., Li, B., and Cheng, J. Exact shape correspondence via 2d graph convolution. In *Advances in Neural Information Processing Systems*, 2022. (Cited on page 14)

Kandasamy, K., Krishnamurthy, A., Poczos, B., Wasserman, L., and robins, j. m. Nonparametric von mises estimators for entropies, divergences and mutual informations. In *Advances in Neural Information Processing Systems*, volume 28, 2015. (Cited on page 7)

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673, 2020. (Cited on pages 7 and 20)

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. (Cited on pages 15 and 23)

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. (Cited on page 1)

Knyazev, B., Taylor, G. W., and Amer, M. R. Understanding attention and generalization in graph neural networks. In *Advances in Neural Information Processing Systems*, pp. 4204–4214, 2019. (Cited on pages 8 and 22)

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning,*, pp. 5637–5664, 2021. (Cited on page 1)

Krueger, D., Caballero, E., Jacobsen, J., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. C. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826, 2021. (Cited on pages 8 and 23)

Lee, J., Kim, E., Lee, J., Lee, J., and Choo, J. Learning debiased representation via disentangled feature augmentation. In *Advances in Neural Information Processing Systems*, 2021. (Cited on pages 3, 5, 6 and 14)

Li, H., Zhang, Z., Wang, X., and Zhu, W. Learning invariant graph representations for out-of-distribution generalization. In *Advances in Neural Information Processing Systems*, 2022. (Cited on pages 1, 2, 3, 4, 8, 14, 15 and 23)

Lin, Y., Zhu, S., Tan, L., and Cui, P. ZIN: When and how to learn invariance without environment partition? In *Advances in Neural Information Processing Systems*, 2022. (Cited on pages 1, 3, 5, 14 and 15)

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792, 2021a. (Cited on pages 3, 6 and 15)

Liu, G., Zhao, T., Xu, J., Luo, T., and Jiang, M. Graph rationalization with environment-based augmentations. *arXiv preprint arXiv:2206.02886*, 2022. (Cited on pages 1, 3, 8, 14, 15 and 23)

Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. Heterogeneous risk minimization. In *International Conference on Machine Learning*, volume 139, pp. 6804–6814, 2021b. (Cited on pages 3 and 15)

Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., and Zhang, X. Parameterized explainer for graph neural network. In *Advances in Neural Information Processing Systems*, pp. 19620–19631, 2020. (Cited on pages 7, 15 and 22)

Mahdavi, S., Swersky, K., Kipf, T., Hashemi, M., Thrampoulidis, C., and Liao, R. Towards better out-of-distribution generalization of neural algorithmic reasoning tasks. *arXiv preprint arXiv:2211.00692*, 2022. (Cited on page 14)

McInnes, L., Healy, J., Saul, N., and Grossberger, L. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. (Cited on page 19)

Mendez, D., Gaulton, A., Bento, A. P., Chambers, J., Veij, M. D., Felix, E., Magariños, M. P., Mosquera, J. F., Mutowo-Meullenet, P., Nowotka, M., Gordillo-Marañón, M., Hunter, F. M. I., Junco, L., Mugumbate, G., Rodríguez-López, M., Atkinson, F., Bosc, N., Radoux, C. J., Segura-Cabrera, A., Hersey, A., and Leach, A. R. Chembl: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(Database-Issue):D930–D940, 2019. (Cited on page 22)

Miao, S., Liu, M., and Li, P. Interpretable and generalizable graph learning via stochastic attention mechanism. *arXiv preprint arXiv:2201.12987*, 2022. (Cited on pages 1, 3, 8, 14 and 23)

Miao, S., Luo, Y., Liu, M., and Li, P. Interpretable geometric deep learning via learnable randomness injection. In *International Conference on Learning Representations*, 2023. (Cited on pages 3 and 14)

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: Training debiased classifier from biased classifier. In *Advances in Neural Information Processing Systems*, 2020. (Cited on page 6)

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019. (Cited on page 24)

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016. (Cited on page 1)

Salakhutdinov, R. and Hinton, G. E. Learning a nonlinear embedding by preserving class neighbourhood structure. In *International Conference on Artificial Intelligence and Statistics*, pp. 412–419, 2007. (Cited on page 7)

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013. (Cited on page 22)

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint*, arXiv:1807.03748, 2018. (Cited on page 7)

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018. (Cited on page 1)

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939, 2020. (Cited on page 7)

Wolpert, D. and Macready, W. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. (Cited on pages 1 and 5)

Wu, Q., Zhang, H., Yan, J., and Wipf, D. Handling distribution shifts on graphs: An invariance perspective. In *International Conference on Learning Representations*, 2022a. (Cited on pages 1, 2, 3, 4, 13 and 14)

Wu, Y., Wang, X., Zhang, A., He, X., and Chua, T.-S. Discovering invariant rationales for graph neural networks. In *International Conference on Learning Representations*, 2022b. (Cited on pages 1, 3, 8, 14 and 15)

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pp. 5449–5458, 2018. (Cited on pages 1 and 22)

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. (Cited on pages 1, 8, 13, 15 and 22)

Xu, K., Zhang, M., Li, J., Du, S. S., Kawarabayashi, K., and Jegelka, S. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations*, 2021. (Cited on page 14)

Yang, N., Zeng, K., Wu, Q., Jia, X., and Yan, J. Learning substructure invariance for out-of-distribution molecular representations. In *Advances in Neural Information Processing Systems*, 2022. (Cited on pages 1, 2, 3, 4, 8, 14, 15 and 23)

Yehudai, G., Fetaya, E., Meirom, E., Chechik, G., and Maron, H. From local structures to size generalization in graph neural networks. In *International Conference on Machine Learning*, pp. 11975–11986, 2021. (Cited on page 14)

Yeung, R. *Information Theory and Network Coding*. 01 2008. ISBN 978-0-387-79233-0. (Cited on page 18)

Yu, J., Xu, T., Rong, Y., Bian, Y., Huang, J., and He, R. Graph information bottleneck for subgraph recognition. In *International Conference on Learning Representations*, 2021. (Cited on pages 3, 8 and 14)

Yu, J. C., Liang, J., and He, R. Finding diverse and predictable subgraphs for graph domain generalization. *arXiv preprint arXiv:2206.09345*, 2022. (Cited on pages 1, 3, 4 and 14)

Yuan, H., Yu, H., Gui, S., and Ji, S. Explainability in graph neural networks: A taxonomic survey. *arXiv preprint*, arXiv:2012.15445, 2020. (Cited on pages 8 and 22)

Zhang, M., Sohoni, N. S., Zhang, H. R., Finn, C., and Ré, C. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. *arXiv preprint*, arXiv:2203.01517, 2022. (Cited on pages 3, 6, 7, 15 and 19)

Zhou, Y., Kutyniok, G., and Ribeiro, B. OOD link prediction generalization capabilities of message-passing GNNs in larger test graphs. In *Advances in Neural Information Processing Systems*, 2022. (Cited on page 14)

Appendix of GALA

Contents

A. Full Details of the Background

We give a more detailed background introduction about GNNs and Invariant Learning in this section.

Graph Neural Networks. Let $G = (A, X)$ denote a graph with n nodes and m edges, where $A \in \{0,1\}^{n\times n}$ is the adjacency matrix, and $X \in \mathbb{R}^{n\times d}$ is the node feature matrix with a node feature dimension of d. In graph classification, we are given a set of N graphs $\{G_i\}_{i=1}^N \subseteq \mathcal{G}$ and their labels $\{Y_i\}_{i=1}^N \subseteq \mathcal{Y} = \mathbb{R}^c$ from c classes. Then, we train a GNN $\rho \circ h$ with an encoder $h : \mathcal{G} \to \mathbb{R}^h$ that learns a meaningful representation h_G for each graph G to help predict their labels $y_G = \rho(h_G)$ with a downstream classifier $\rho : \mathbb{R}^h \to \mathcal{Y}$. The representation h_G is typically obtained by performing pooling with a READOUT function on the learned node representations:

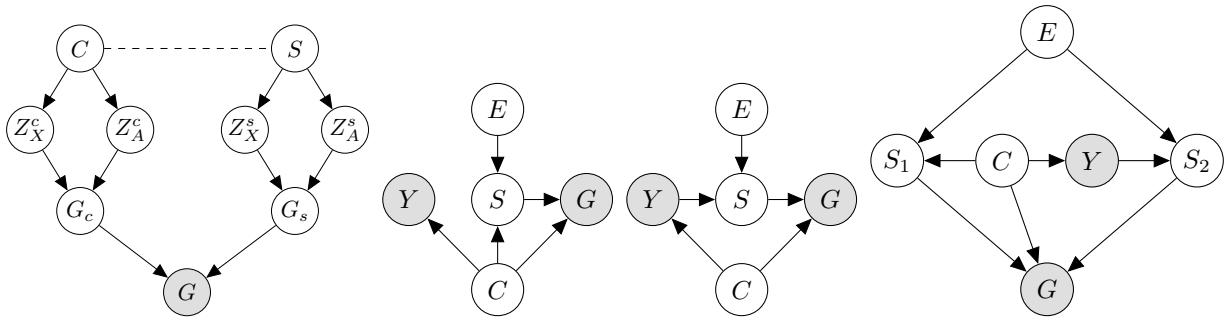
$$h_G = \text{READOUT}(\{h_u^{(K)} | u \in V\}), \tag{11}$$

where the READOUT is a permutation invariant function (e.g., SUM, MEAN) (Xu et al., 2019), and $h_u^{(K)}$ stands for the node representation of $u \in V$ at K-th layer that is obtained by neighbor aggregation:

$$h_u^{(K)} = \sigma(W_K \cdot a(\{h_v^{(K-1)}\} | v \in \mathcal{N}(u) \cup \{u\})), \tag{12}$$

where $\mathcal{N}(u)$ is the set of neighbors of node u, $\sigma(\cdot)$ is an activation function, e.g., ReLU, and $a(\cdot)$ is an aggregation function over neighbors, e.g., MEAN.

Graph generation process. This work focuses on graph classification, while the results generalize to node classification as well using the same setting as in Wu et al. (2022a). Specifically, we are given a set of graph datasets $\mathcal{D} = \{\mathcal{D}_e\}_e$ collected from multiple environments $\mathcal{E}_{\text{all}}$. Samples $(G_i^e, Y_i^e) \in \mathcal{D}^e$ from the same environment are considered as drawn independently from an identical distribution $\mathbb{P}^e$. We consider the graph generation process proposed by Chen et al. (2022a) that covers a broad case of graph distribution shifts. Fig. 4 shows the full graph generation process considered in Chen



(a) Graph generation SCM (b) FIIF SCM (c) PIIF SCM (d) MIIF SCM

Figure 4. Full SCMs on Graph Distribution Shifts (Chen et al., 2022a).

et al. (2022a). The generation of the observed graph G and labels Y are controlled by a set of latent causal variable C and spurious variable S, i.e.,

$$G \coloneqq f_{\text{gen}}(C, S).$$

C and S control the generation of G by controlling the underlying invariant subgraph G_c and spurious subgraph G_s, respectively. Since S can be affected by the environment E, the correlation between Y, S and G_s can change arbitrarily when the environment changes. C and S control the generation of the underlying invariant subgraph G_c and spurious subgraph G_s, respectively. Since S can be affected by the environment E, the correlation between Y, S and G_s can change arbitrarily when the environment changes. Besides, the latent interaction among C, S and Y can be further categorized into *Full Informative Invariant Features* (*FIIF*) when $Y \perp\!\!\!\perp S|C$ and *Partially Informative Invariant Features* (*PIIF*) when $Y \not\perp\!\!\!\perp S|C$. Furthermore, PIIF and FIIF shifts can be mixed together and yield *Mixed Informative Invariant Features* (*MIIF*), as shown in Fig. 4. We refer interested readers to Chen et al. (2022a) for a detailed introduction of the graph generation process.

Invariant graph representation learning. To tackle the OOD generalization challenge on graphs from Fig. 4, the existing invariant graph learning approaches generically aim to identify the underlying invariant subgraph G_c to predict the label Y (Wu et al., 2022a; Chen et al., 2022a). Specifically, the goal of OOD generalization on graphs is to learn an *invariant GNN* $f \coloneqq f_c \circ g$, which is composed of two modules: a) a featurizer $g : \mathcal{G} \to \mathcal{G}_c$ that extracts the invariant subgraph G_c; b) a classifier $f_c : \mathcal{G}_c \to \mathcal{Y}$ that predicts the label Y based on the extracted G_c, where $\mathcal{G}_c$ refers to the space of subgraphs of $\mathcal{G}$. The learning objectives of f_c and g are formulated as

$$\max_{f_c,\, g} I(\widehat{G}_c; Y), \text{ s.t. } \widehat{G}_c \perp\!\!\!\perp E, \ \widehat{G}_c = g(G). \tag{13}$$

Since E is not observed, many strategies are proposed to impose the independence of $\widehat{G}_c$ and E. A common approach is to augment the environment information. For example, based on the estimated invariant subgraphs $\widehat{G}_c$ and spurious subgraphs $\widehat{G}_s$, Wu et al. (2022b); Liu et al. (2022); Wu et al. (2022a) proposed to generate new environments, while Yang et al. (2022); Li et al. (2022) proposed to infer the underlying environment labels. However, we show that it is fundamentally impossible to augment faithful environment information in Sec. 3. Yu et al. (2021); Miao et al. (2022); Yu et al. (2022); Miao et al. (2023) adopt graph information bottleneck to tackle FIIF graph shifts, and they cannot generalize to PIIF shifts. Our works focuses on PIIF shifts, as it is more challenging when without environment labels (Lin et al., 2022). Fan et al. (2022) generalized (Lee et al., 2021) to tackle severe graph biases, i.e., when $H(S|Y) < H(C|Y)$. Chen et al. (2022a) proposed a contrastive framework to tackle both FIIF and PIFF graph shifts, but limited to $H(S|Y) > H(C|Y)$. However, in practice it is usually unknown whether $H(S|Y) < H(C|Y)$ or $H(S|Y) > H(C|Y)$ without environment information.

More OOD generalization on graphs. In addition to the aforementioned invariant learning approaches, Yehudai et al. (2021); Bevilacqua et al. (2021); Zhou et al. (2022) study the OOD generalization as extrapolation from small graphs to larger graphs in the task of graph classification and link prediction. In contrast, we study OOD generalization against various graph distribution shifts formulated in Fig. 4. In addition to the standard OOD generalization tasks studied in this paper, Xu et al. (2021); Mahdavi et al. (2022) study the OOD generalization in tasks of algorithmic reasoning on graphs. Jin et al. (2022) study the test-time adaption in the graph regime. Kamhoua et al. (2022) study the 3D shape matching under the presence of noises.

Invariant learning without environment labels. There are also plentiful studies in invariant learning without environment labels. Creager et al. (2021a) proposed a minmax formulation to infer the environment labels. Liu et al. (2021b) proposed a self-boosting framework based on the estimated invariant and variant features. Liu et al. (2021a); Zhang et al. (2022) proposed to infer labels based the predictions of an ERM trained model. However, Lin et al. (2022) found failure cases in Euclidean data where it is impossible to identify the invariant features without given environment labels. Moreover, as the OOD generalization on graphs is fundamentally more difficult than Euclidean data (Chen et al., 2022a), the question about the feasibility of learning invariant subgraphs without environment labels remains unanswered.

B. More Details about the Failure Cases

We provide details about the failure case verification experiments in complementary to Sec. 3. The failure cases are constructed according to the two-piece graph generation models. The specific description is given as the following.

Definition B.1 (3-class two-piece graphs)**.** *Each environment is defined with two parameters, $\alpha_e, \beta_e \in [0, 1]$, and the dataset $\mathcal{D}_e$ is generated as follows:*

(a) Sample $y^e \in \{0, 1, 2\}$ uniformly;

(b) Generate G_c and G_s via :

$$G_c \coloneqq f_{gen}^{G_c}(Y \cdot Rad(\alpha_e)),\ G_s \coloneqq f_{gen}^{G_s}(Y \cdot Rad(\beta_e)),$$

where $f_{gen}^{G_c}, f_{gen}^{G_s}$ respectively map input $\{0, 1, 2\}$ to a specific graph selected from a given set, and $Rad(\alpha)$ is a random variable with probability α taking a uniformly random value from $\{0, 1, 2\}$, and a probability of $1 - \alpha$ taking the value of $+1$;

(c) Sythesize G by randomly concatenating G_c and G_s:

$$G \coloneqq f_{gen}^{G}(G_c, G_s).$$

In experiments, we implement the 3-class two-piece graphs with the BA-motifs (Luo et al., 2020) model.

In experiments, we adopt a 3-layer GIN (Xu et al., 2019) with a hidden dimension of 32 and a dropout rate of 0.0 as the GNN encoder. The optimization is proceeded with Adam (Kingma & Ba, 2015) using a learning rate of $1e-3$. All experiments are repeated with 5 different random seeds of $\{1, 2, 3, 4, 5\}$. The mean and standard deviation are reported from the 5 runs.

We implement DIR (Wu et al., 2022b), GREA (Liu et al., 2022), MoleOOD (Yang et al., 2022), GIL (Li et al., 2022), DisC (Fan et al., 2022), and CIGA (Chen et al., 2022a), according to the author provided codes (if available). As for the hyperparameters in each method, we use a penalty weight of $1e-2$ for DIR following its original experiment in spurious motif datasets generated similarly using BA-motifs (Wu et al., 2022b). We use a penalty weight of 1 for GREA as we empirically it does not affect the performance by changing to different weights. For MoleOOD and GIL, we set the number of environments as 3. We tune the penalty weights of MoleOOD with values from $\{1e-2, 1e-1, 1, 10\}$ but did not observe much performance differences. We tune the penalty weights of GIL with values from $\{1e-5, 1e-3, 1e-1\}$ recommended by the authors. For DisC, we tune only the q weight from $\{0.9, 0.7, 0.5\}$ in the GCE loss as we did not observe performance differences by changing the weight of the other term. We tune the penalty weight of CIGA with values from $\{0.5, 1, 2, 4, 8, 16, 32\}$ as recommended by the authors.

C. Proofs for Theorems and Propositions

C.1. Proof of Proposition 3.2

Proposition C.1. *(Restatement of Proposition 3.2) Consider the two-piece graph dataset $\mathcal{E}_{tr} = \{(\alpha, \beta_1), (\alpha, \beta_2)\}$ with $\alpha \geq \beta_1, \beta_2$ (e.g., $\mathcal{E}_{tr} = \{(0.25, 0.1), (0.25, 0.2)\}$), and its corresponding mixed environment $\mathcal{E}_{tr}^{mix} = \{(\alpha, (\beta_1 + \beta_2)/2\}$ (e.g., $\mathcal{E}_{tr}^{mix} = \{(0.25, 0.15)\}$). It holds that the augmented environment $\mathcal{E}_v$ is also a two-piece graph dataset with*

$$\mathcal{E}_v = \{(0.5, (\beta_1 + \beta_2)/2)\} \text{ (e.g., } \mathcal{E}_v = \{(0.5, 0.15)\}).$$

Proof. From Definition 3.1, we known that for each graph $G_i \sim \mathcal{E}_{\text{tr}}^{\text{mix}} = \{(\alpha, (\beta_1 + \beta_2)/2)\}$, G_i is the concatenation of the G_c^i and G_s^i defined as

$$G_c^i \coloneqq f_{\text{gen}}^{G_c}(Y_i \cdot \text{Rad}(\alpha)_i), \quad G_s^i \coloneqq f_{\text{gen}}^{G_s}(Y_i \cdot \text{Rad}((\beta_1 + \beta_2)/2)_i),$$

where $\text{Rad}(\cdot)_i$ denotes the ith sample of the random variable $\text{Rad}(\cdot)$.

Denote

$$G_A = f_{\text{gen}}^{G_c}(+1),\ G_B = f_{\text{gen}}^{G_c}(-1),$$

and

$$G_C = f_{\text{gen}}^{G_s}(+1),\ G_D = f_{\text{gen}}^{G_s}(-1),$$

Considering applying the augmentation to $2n$ samples randomly sampled from $\mathcal{E}_{\text{tr}}^{\text{mix}}$, since the featurizer g separates each $G \in \mathcal{E}_{\text{tr}}^{\text{mix}}$ into $\widehat{G}_c = G_s$ and $\widehat{G}_s = G_c$, and the augmented graph G^i is obtained by

$$G^{i,j} = f_{\text{gen}}^{G}(\widehat{G}_c^i,\ \widehat{G}_s^j), \forall i, j \in \{1...n\}.$$

Then, the new α_v, β_v in $\mathcal{E}_v$ can be obtained by summing up the overall numbers of G_A, G_B, G_C, G_D concatenated into $2n^2$ samples in $\mathcal{E}_v$.

Specifically, we can inspect the changes of the distributions of motifs and labels. Let $\bar{\beta} = (\beta_1 + \beta_2)/2$, without loss of generality, we focus on inspecting the changes given $Y = +1$, since the changes given $Y = -1$ is symmetric as $Y = +1$. The original distribution is shown as follows:

$Y = +1$	G_A	G_B
G_C	$(1-\alpha)(1-\bar{\beta})n$	$\alpha(1-\bar{\beta})n$
G_D	$(1-\alpha)\bar{\beta}n$	$\alpha\bar{\beta}n$

Then, new distributions of the motifs and labels are determined by the number of original motifs identified as $\widehat{G}_c$ and $\widehat{G}_s$, respectively. When $\widehat{G}_c = G_s$ and $\widehat{G}_s = G_c$, in the new environment $\mathcal{E}_v$, given $Y = +1$, G_C contributes $(1-\bar{\beta})n * 2n$ samples as the "invariant" subgraph. More specifically, G_C will be concatenated with G_A and G_B by n times, respectively. Then we have the new distribution tables shown as follows:

$Y = +1$	G_A	G_B
G_C	$(1-\beta)n^2$	$(1-\beta)n^2$
G_D	βn^2	βn^2

Since given the same Y, the spurious subgraph G_C and G_D will still have the same chance being flipped, we have $\beta_v = \bar{\beta}$. While as G_A and G_B appear the same times given the same Y, it suffices to know that $\alpha_v = 0.5$. □

C.2. Proof of Theorem 3.3

Theorem C.2. *(Restatement of Theorem 3.3) Given the same graph generation process as in Fig. 2, when there exists spurious subgraph G_s such that $P^{e_1}(Y|G_s) = P^{e_2}(Y|G_s)$ for any two environments $e_1, e_2 \in \mathcal{E}_{tr}$, where $P^e(Y|G_s)$ is the conditional distribution $P(Y|G_s)$ under environment $e \in \mathcal{E}_{all}$, it is impossible for any learning algorithm applied to $f_c \circ g$ to differentiate G_c from G_s.*

Proof. Let G_s^* be the spurious subgraph such that $P^{e_1}(Y|G_s) = P^{e_2}(Y|G_s)$ for any two environments $e_1, e_2 \in \mathcal{E}_{\text{tr}}$, and G_c be the invariant subgraph which $P^{e_1}(Y|G_c) = P^{e_2}(Y|G_c),\ \forall e_1, e_2 \in \mathcal{E}_{\text{tr}}$ by definition. Consider a learning algorithm applied to $f_c \circ g$ that accepts the input of $\mathcal{E}_{\text{tr}}^{\text{mix}}$, and extracts a subgraph $\widehat{G}_c = g(Y)$ as an estimation of the invariant subgraph for any G to predict Y via $f_c(\widehat{G}_c)$ in a deterministic manner. If the algorithm succeed to extract G_c from $\mathcal{E}_{\text{tr}}^{\text{mix}}$, then there always exists a $\mathcal{E}_{\text{tr}}^{\text{mix}'}$ with the desired spurious subgraph G_s' and a underlying invariant subgraph G_c', such that $G_s' = G_c$ and $G_c' = G_s^*$. Due to the deterministic nature, the algorithm fails to identify G_c' in $\mathcal{E}_{\text{tr}}^{\text{mix}'}$. □

C.3. Proof of Proposition 3.5

Proposition C.3. *(Restatement of Proposition 3.5) There exist 2 two-piece graph training environments $\mathcal{E}_{tr}$ and $\mathcal{E}_{tr}{}'$, whose mixed training environments are the same, such that any learning algorithm will fail to capture the invariance of at least one of the training environments.*

Proof. Let the mixed training environment of $\mathcal{E}_{\text{tr}}$ and $\mathcal{E}_{\text{tr}}{}'$ be $\mathcal{E}_{\text{tr}}^{\text{mix}} = \{(\alpha, \beta)\}$. Based on the definition of two-piece graphs (Definition 3.1), the joint distribution of the mixed training dataset $(G = \text{Concat}[G_c, G_s], Y)$ can be computed as

$$\begin{cases} Y = +1, & \text{with probability } 0.5, \\ Y = -1, & \text{with probability } 0.5, \\ \text{Bit}^{G_c}(G_c) = \text{Bit}^{G_s}(G_s) = Y, & \text{with probability } (1-\alpha)(1-\beta), \\ \text{Bit}^{G_c}(G_c) \neq \text{Bit}^{G_s}(G_s) = Y, & \text{with probability } \alpha(1-\beta), \\ \text{Bit}^{G_s}(G_s) \neq \text{Bit}^{G_c}(G_c) = Y, & \text{with probability } (1-\alpha)\beta, \\ \text{Bit}^{G_c}(G_c) = \text{Bit}^{G_s}(G_s) \neq Y, & \text{with probability } \alpha\beta. \end{cases}$$

Here we use $\text{Bit}^{G_c}(G_c)$ to obtain the input bit of a subgraph G_c (or $(f_{\text{gen}}^{G_c})^{-1}$), and $\text{Bit}^{G_s}(G_s)$ for G_s, respectively.

Any learning algorithm that tries to identify the invariant subgraph from this training dataset will compute a model that uses subgraph G_c, or subgraph G_s, or both G_c and G_s to predict Y deterministically. Thus, as long as the joint distribution does not change, the resulting model will always identify the same invariant subgraph. Without loss of generality, let us assume that the model correctly identifies G_c as the invariant subgraph for $\mathcal{E}_{\text{tr}} = \{(\alpha, \beta_1), (\alpha, \beta_2)\}$ with $\beta = (\beta_1 + \beta_2)/2$.

Now let the other training environment be $\mathcal{E}_{\text{tr}}{}' = \{(\alpha_1, \beta), (\alpha_2, \beta)\}$ with $\alpha = (\alpha_1 + \alpha_2)/2$. It is clear that since the mixed training environment of $\mathcal{E}_{\text{tr}}{}'$ is still $\{(\alpha, \beta)\}$, the model keeps regarding G_c as the invariant subgraph. However, for $\mathcal{E}_{\text{tr}}{}'$, the model fails to identify the invariance since now the invariant subgraph is G_s.

□

C.4. Proof of Corollary 3.7

Corollary C.4. *(Restatement of Corollary 3.7) Without Assumption 3.6, then there does not exist a learning algorithm that captures the invariance of all of the two-piece graph environments.*

Proof. Consider a learning algorithm applied to $f_c \circ g$ that accepts the input of $\mathcal{E}_{\text{tr}}^{\text{mix}}$, and extracts a subgraph $\widehat{G}_c = g(Y)$ as an estimation of the invariant subgraph for any G to predict Y via $f_c(\widehat{G}_c)$ in a deterministic manner. Without the holding of Assumption 3.6, due to Proposition 3.5, there exists $\mathcal{E}_{\text{tr}}^{\text{mix}}{}'$ for each $\mathcal{E}_{\text{tr}}^{\text{mix}}$ that have the identical joint distribution but different underlying invariant subgraph. Thus, any learning algorithm that succeeds in either $\mathcal{E}_{\text{tr}}^{\text{mix}}$ or $\mathcal{E}_{\text{tr}}^{\text{mix}}{}'$ will fail in the other. □

C.5. Proof of Theorem 4.1

Theorem C.5. *(Restatement of Theorem 4.1) Given, i) the same data generation process as in Fig. 2; ii) $\mathcal{D}_{\text{tr}}$ that satisfies variation sufficiency (Assumption 3.4) and variation consistency (Assumption 3.6); iii) $\{G^p\}$ and $\{G^n\}$ are distinct subsets of $\mathcal{D}_{\text{tr}}$ such that $I(G_s^n; G_s^p | Y) = 0$, $\forall G_s^p \arg\max_{\widehat{G}_s^p} I(\widehat{G}_s^p; Y)$ under $\{G^p\}$, and $\forall G_s^n \arg\max_{\widetilde{G}_s^n} I(\widetilde{G}_s^n; Y)$ under $\{G^n\}$; suppose $|G_c| = s_c$, $\forall G_c$, resolving the following* GALA *objective elicits an invariant GNN defined via Eq. 13,*

$$\max_{f_c, g} \; I(\widehat{G}_c; Y), \; \textit{s.t. } g \in \arg\max_{\hat{g}, |\widehat{G}_c^p| \leq s_c} I(\widehat{G}_c^p; \widetilde{G}_c^n | Y), \tag{14}$$

where $\widetilde{G}_c^p \in \{\widehat{G}_c^p = g(G^p)\}$ and $\widetilde{G}_c^n \in \{\widetilde{G}_c^n = g(G^n)\}$ are the estimated invariant subgraphs via g from $\{G^p\}$ and $\{G^n\}$, respectively.

Proof. Without loss of generality, we assume that $\{G^p\}$ has the same spurious dominance situation as $\mathcal{E}_{\text{tr}}$. In other words, when $H(S|Y) < H(C|Y)$, the data distribution in $\{G^p\}$ also follows $H(S|Y) < H(C|Y)$, while $H(S|Y) > H(C|Y)$ in $\{G^n\}$. To proceed, we will use the language of Chen et al. (2022a).

We begin by discussing the case of $H(S|Y) < H(C|Y)$. Given $H(S|Y) < H(C|Y)$, we have $H(S|Y) < H(C|Y)$ in $\{G^p\}$ and $H(S|Y) > H(C|Y)$ in $\{G^n\}$. Then, we claim that

$$G_c \in \arg\max_{\widehat{G}_c^p, |\widehat{G}_c^p| \leq s_c} I(\widehat{G}_c^p; \widetilde{G}_c^n | Y). \tag{15}$$

Otherwise, consider there exists a subgraph of the spurious subgraph $G_s^p \subseteq G_s$ in $\widehat{G}_c^p$, which takes up the space of $\widehat{G}_c^l \subseteq G_c$ from $\widehat{G}_c^p$. Then, we can inspect the changes to $I(\widehat{G}_c^p; \widetilde{G}_c^n|Y)$ led by G_s^p:

$$\begin{aligned}\triangle I(\widehat{G}_c^p; \widetilde{G}_c^n|Y) &= \triangle H(\widehat{G}_c^p|Y) - \triangle H(\widehat{G}_c^p|\widehat{G}_c^p, \widetilde{G}_c^n, Y) \\ &= \Big[H(\widehat{G}_c^l, \widehat{G}_s^p|Y) - H(\widehat{G}_c^l, \widehat{G}_c^p|Y)\Big] - \Big[H(\widehat{G}_c^l, \widehat{G}_s^p|\widetilde{G}_c^n, Y) - H(\widehat{G}_c^l, \widehat{G}_c^p|\widetilde{G}_c^n, Y)\Big] \\ &= \Big[H(\widehat{G}_s^p|\widehat{G}_c^l, Y) - H(\widehat{G}_c^p|\widehat{G}_c^l, Y)\Big] - \Big[H(\widehat{G}_s^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y) - H(\widehat{G}_c^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y)\Big],\end{aligned} \tag{16}$$

where the last equality is obtained via expanding the conditional entropy. Then, considering the contents in $\widetilde{G}_c^n$, without loss of generality, we can divide all of the possible cases into two:

(i) $\widetilde{G}_c^n$ contains only the corresponding invariant subgraph G_c^n;

(ii) $\widetilde{G}_c^n$ contains from the corresponding spurious subgraph G_s^n, denoted as $\widetilde{G}_s^n \subseteq G_s^n$;

For case (i), it easy to write Eq. 16 as:

$$\begin{aligned}\triangle I(\widehat{G}_c^p; \widetilde{G}_c^n|Y) &= \Big[H(\widehat{G}_s^p|\widehat{G}_c^l, Y) - H(\widehat{G}_c^p|\widehat{G}_c^l, Y)\Big] - \Big[H(\widehat{G}_s^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y) - H(\widehat{G}_c^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y)\Big], \\ &= -H(\widehat{G}_c^p|\widehat{G}_c^l, Y) + H(\widehat{G}_c^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y),\end{aligned} \tag{17}$$

since $H(\widehat{G}_s^p|\widehat{G}_c^l, Y) = H(\widehat{G}_s^p|\widetilde{G}_c^n, \widehat{G}_c^l, Y) = H(\widehat{G}_s^p|Y)$ given $C \perp\!\!\!\perp S|Y$ for PIIF shifts. Then, it suffices to know that $\triangle I(\widehat{G}_c^p; \widetilde{G}_c^n|Y) \leq 0$ as conditioning on new variables will not increase the entropy (Yeung, 2008).

For case (ii), we have :

$$\begin{aligned}\triangle I(\widehat{G}_c^p; \widetilde{G}_c^n|Y) &= \Big[H(\widehat{G}_s^p|\widehat{G}_c^l, Y) - H(\widehat{G}_c^p|\widehat{G}_c^l, Y)\Big] - \Big[H(\widehat{G}_s^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y) - H(\widehat{G}_c^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y)\Big], \\ &= \Big[-H(\widehat{G}_c^p|\widehat{G}_c^l, Y) + H(\widehat{G}_c^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y)\Big] + \Big[H(\widehat{G}_s^p|\widehat{G}_c^l, Y) - H(\widehat{G}_s^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y)\Big],\end{aligned} \tag{18}$$

where we claim that $H(\widehat{G}_s^p|\widehat{G}_c^l, Y) - H(\widehat{G}_s^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y) = 0$, and similarly conclude that $\triangle I(\widehat{G}_c^p; \widetilde{G}_c^n|Y) \leq 0$. More specifically, we can rewrite the first term in Eq. 18 as

$$\begin{aligned}H(\widehat{G}_s^p|\widehat{G}_c^l, Y) - H(\widehat{G}_s^p|\widehat{G}_c^l, \widetilde{G}_c^n, Y) &= H(\widehat{G}_s^p|Y) - H(\widehat{G}_s^p|\widehat{G}_s^n, Y) \\ &= I(\widehat{G}_s^p; \widehat{G}_s^n|Y) = 0,\end{aligned}$$

using the variation condition for $\widehat{G}_s^p$ under $\{G^p\}$, and $\widetilde{G}_s^n$ under $\{G^n\}$.

After showing the success of GALA in tackling $H(S|Y) < H(C|Y)$, it is also suffices to know that the aforementioned discussion also generalizes to the other case, i.e., when $H(S|Y) > H(C|Y)$ in $\{G^p\}$ and $H(S|Y) < H(C|Y)$ in $\{G^n\}$. □

D. More Discussions on Practical Implementations of GALA

In this section, we provide more implementation discussions about GALA in complementary to Sec. 4.3.

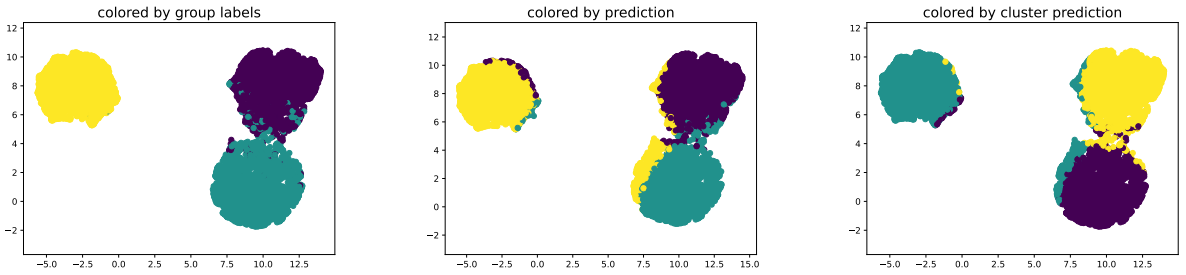
Environment assistant implementation. Theorem 4.1 shows the effectiveness of GALA when given proper subsets of $\{G^p\}$ and $\{G^n\}$. In practice, we can implement the environment assistant into multiple forms. As discussed in Sec. 4.1, ERM trained model can serve as a reliable proxy. Since ERM tends to learn the first dominant features, when $H(S|Y) < H(C|Y)$, ERM will firstly learn to extract spurious subgraphs G_s to make predictions. Therefore, we can obtain $\{G^p\}$ by finding samples where ERM correctly predicts the labels, while $\{G^n\}$ for samples that ERM predicts an incorrect label. In addition to direct label predictions, we can also adopt clustering (Zhang et al., 2022) to yield environment assistant predictions for better contrastive sampling. We provide the detailed description of the clustering based variant of GALA in Algorithm 2.

Algorithm 2 GALA: Clustering based **G**raph inv**A**riant **L**earning **A**ssistant

Input: Training data $\mathcal{D}_{\text{tr}}$; environment assistant A; featurizer g; classifier f_c; length of maximum training epochs e; batch size b;
Initialize environment assistant A;
for $p \in [1, \dots, e]$ **do**
  Sample a batch of data $\{G_i, Y_i\}_{i=1}^b$ from $\mathcal{D}_{\text{tr}}$;
  Obtain Environment Assistant predictions $\{\hat{c}_i^e\}_{i=1}^b$ using k-means clustering on the graph representations yielded by A;
  for each sample $G_i, y_i \in \{G_i, Y_i\}_{i=1}^b$ **do**
    Find *postive* graphs with same y_i and different $\hat{c}_i^e$;
    Find *negative* graphs with different y_i but same environment assistant prediction $\hat{c}_i^e$;
    Calculate GALA risk via Eq. 10;
    Update f_c, g via gradients from GALA risk;
  end for
end for
return final model $f_c \circ g$;

Empirically, we find clustering based variant can provide better performance when the spurious correlations are well learned by the environment assistant model. More concretely, we plot the umap visualizations (McInnes et al., 2018) of ERM trained environment assistant model as in Fig. 5, where we can find that clustering predictions provide a better approximations to the underlying group labels.

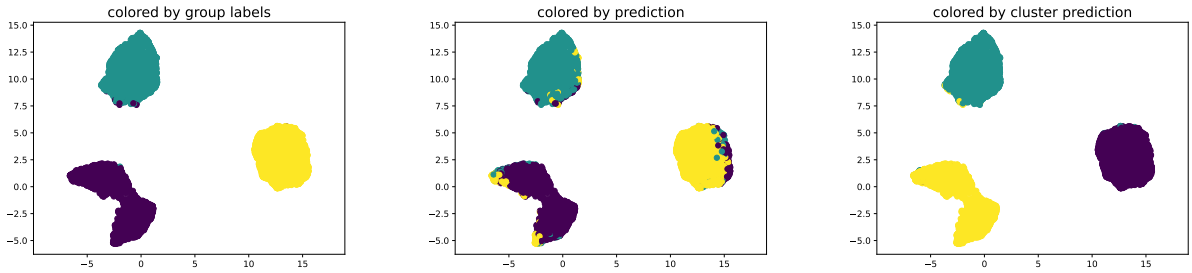
Besides, we can also incorporate models that are easier to overfit to the first dominant features to better differentiate $\{G^p\}$ from $\{G^n\}$. To demonstrate the difference of environment assistant implementations, we conduct more studies with interpretable GNNs with a interpretable ratio of 30% trained with ERM and also with a CIGAv1 penalty of 4.



(a) Colored by environment labels. (b) Colored by label predictions. (c) Colored by clustering predictions.

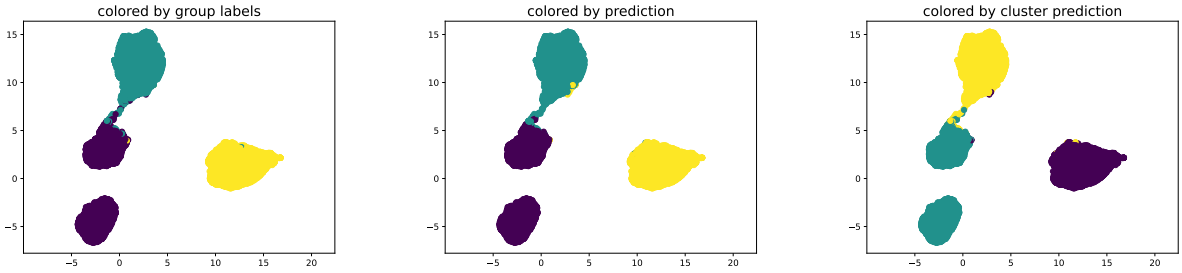
Figure 5. Umap visualizations of learned graph representations in ERM trained environment assistant model based on the 3-class two-piece graph $\{0.7, 0.9\}$.

In Fig. 6 and Fig. 7, it can be found that the interpretable GNN learns hidden representations that are better clustered with group labels. The clustering based predictions yields a better approximation of the underlying environment labels.



(a) Colored by environment labels. (b) Colored by label predictions. (c) Colored by clustering predictions.

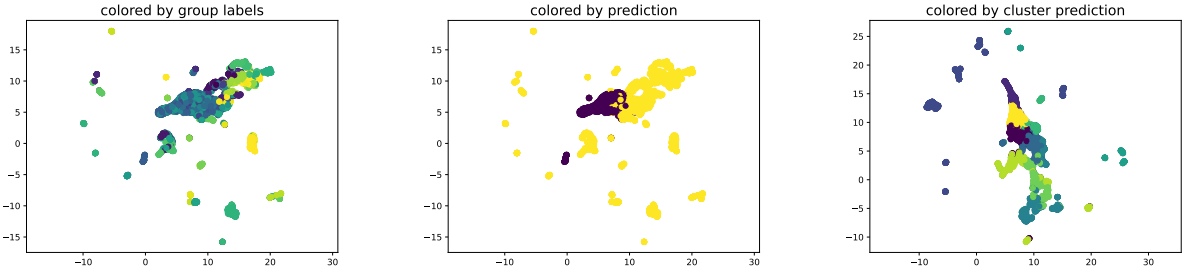
Figure 6. Umap visualizations of learned graph representations in an interpretable GNN model (ratio=30%) trained with ERM based on the 3-class two-piece graph $\{0.7, 0.9\}$.



(a) Colored by environment labels. (b) Colored by label predictions. (c) Colored by clustering predictions.

Figure 7. Umap visualizations of learned graph representations in an interpretable GNN model (ratio=30%) trained with ERM based on the 3-class two-piece graph $\{0.7, 0.9\}$.

Furthermore, when implementing the environment assistant model using a interpretable GNN as well as a CIGAv1 penalty, which facilitates the overfitting to the spurious correlations, then the vanilla label predictions can also yield a good approximation of the underlying environment labels.

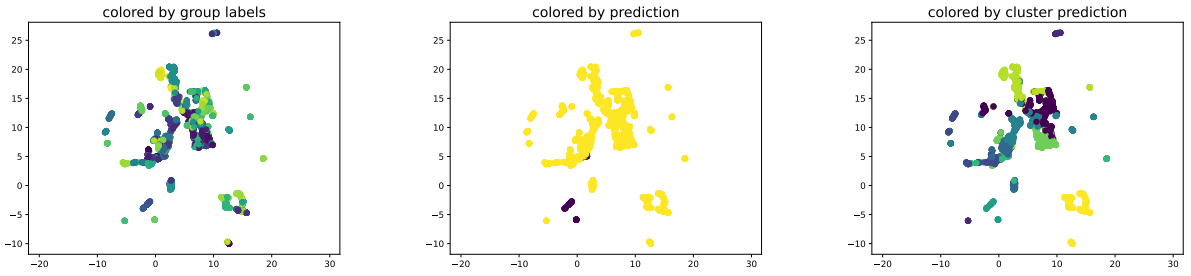


(a) Colored by environment labels. (b) Colored by label predictions. (c) Colored by clustering predictions.

Figure 8. Umap visualizations of learned graph representations of a interpretable GNN trained by ERM on EC50-Assay.

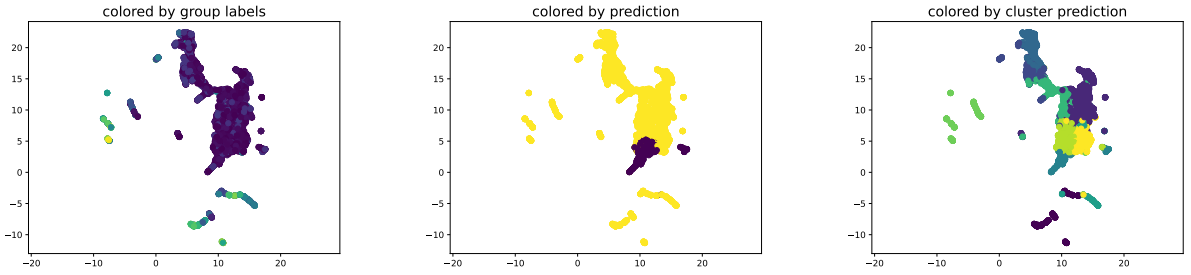
Although using the clustering predictions seem to be promising, we also find negative cases. For example, in DrugOOD datasets, the number of curated environment labels are much larger that learning a well clusterd hidden representations for the environment labels appears to be difficult. Shown as in Fig. 8 to Fig. 10, the learned representations have poor quality for approximating the underlying environment labels. Empirically, we also find that direct using label predictions in DrugOOD datasets generically yield better performance.

One-side contrastive sampling. The original supervised contrastive implementation (Khosla et al., 2020) takes positive and negative samples within the batch using two-side contrastive sampling. That is, all the samples will be considered as



(a) Colored by environment labels. (b) Colored by label predictions. (c) Colored by clustering predictions.

Figure 9. Umap visualizations of learned graph representations of a interpretable GNN trained by ERM on EC50-Scaffold.



(a) Colored by environment labels. (b) Colored by label predictions. (c) Colored by clustering predictions.

Figure 10. Umap visualizations of learned graph representations of a interpretable GNN trained by ERM on EC50-Size.

anchor points. However, when it is used to contrast samples from $\widehat{G}_c^p$ and $\widetilde{G}_c^n$, there could be undesired behaviors. First, it can often happen that there are few to no negative cases when the spurious correlations are too strong. The samples from $\{G^p\}$ in a batch may pull the representations of samples from $\{G^n\}$ to even closer, which makes the model further overfitted to the spurious correlations. Second, the sampling over $\widehat{G}_c^p$ and $\widetilde{G}_c^n$, can be seen as hard positive and negative samples, that may impose a too strong regularizations that preventing the learning of any correlations. Therefore, we propose to use one-side sampling. That is, only using the incorrectly predicted samples as anchor points. We empirically observe one-side sampling could yield better performance in two-piece graphs.

E. More Details about the Experiments

In this section, we provide more details about the experiments, including the dataset preparation, baseline implementations, models and hyperparameters selection as well as the evaluation protocols.

Table 4. Information about the datasets used in experiments. The number of nodes and edges are respectively taking average among all graphs.

DATASETS	# TRAINING	# VALIDATION	# TESTING	# CLASSES	# NODES	# EDGES	METRICS
TWO-PIECE GRAPHS $\{0.8, 0.6\}$	$9,000$	$3,000$	$3,000$	3	26.14	36.21	ACC
TWO-PIECE GRAPHS $\{0.8, 0.7\}$	$9,000$	$3,000$	$3,000$	3	26.18	36.27	ACC
TWO-PIECE GRAPHS $\{0.8, 0.9\}$	$9,000$	$3,000$	$3,000$	3	26.13	36.22	ACC
TWO-PIECE GRAPHS $\{0.7, 0.9\}$	$9,000$	$3,000$	$3,000$	3	26.13	36.22	ACC
CMNIST-SP	$40,000$	$5,000$	$15,000$	2	56.90	373.85	ACC
GRAPH-SST2	$24,881$	$7,004$	$12,893$	2	10.20	18.40	ACC
GRAPH-SST5	$6,090$	$1,186$	$2,240$	5	19.85	37.70	ACC
TWITTER	$3,238$	694	$1,509$	3	21.10	40.20	ACC
EC50-ASSAY	$4,978$	$2,761$	$2,725$	2	40.89	87.18	ROC-AUC
EC50-SCAFFOLD	$2,743$	$2,723$	$2,762$	2	35.54	75.56	ROC-AUC
EC50-SIZE	$5,189$	$2,495$	$2,505$	2	35.12	75.30	ROC-AUC

E.1. Datasets

We provide more details about the motivation and construction method of the datasets that are used in our experiments. Statistics of the datasets are presented in Table 4.

Two-piece graph datasets. We construct 3-class synthetic datasets based on BAMotif (Luo et al., 2020) following Def. B.1, where the model needs to tell which one of three motifs (House, Cycle, Crane) the graph contains. For each dataset, we generate 3000 graphs for each class at the training set, 1000 graphs for each class at the validation set and testing set, respectively. Each dataset is defined with two variables $\{a, b\}$ referring to the strength of invariant and spurious correlations. Given $\{a, b\}$, we generate the training data following the percise generation process as Def. B.1. While for the generation of validation sets, we use a $b_v = \max(1/3, b - 0.2)$ that facilitates the model selection for OOD generalization (Gulrajani & Lopez-Paz, 2021; Chen et al., 2022b). While for the generation of test datasets, we merely use a $b = 0.33$ that contains no distribution shifts, to fully examine to what extent the model learns the invariant correlations. During the construction, we merely inject the distribution shifts in the training data while keeping the testing data and validation data without the biases.

CMNIST-sp. To study the effects of PIIF shifts, we select the ColoredMNIST dataset created in IRM (Arjovsky et al., 2019). We convert the ColoredMnist into graphs using super pixel algorithm introduced by Knyazev et al. (2019). Specifically, the original Mnist dataset are assigned to binary labels where images with digits $0 - 4$ are assigned to $y = 0$ and those with digits $5 - 9$ are assigned to $y = 1$. Then, y will be flipped with a probability of 0.25. Thirdly, green and red colors will be respectively assigned to images with labels 0 and 1 an averaged probability of 0.15 (since we do not have environment splits) for the training data. While for the validation and testing data the probability is flipped to 0.9.

Graph-SST datasets. Inspired by the data splits generation for studying distribution shifts on graph sizes, we split the data curated from sentiment graph data (Yuan et al., 2020), that converts sentiment sentence classification datasets **Graph-SST2**, **Graph-SST5** and **SST-Twitter** (Socher et al., 2013; Dong et al., 2014) into graphs, where node features are generated using BERT (Devlin et al., 2019) and the edges are parsed by a Biaffine parser (Gardner et al., 2018). Our splits are created according to the averaged degrees of each graph. Specifically, we assign the graphs as follows: Those that have smaller or equal than 50-th percentile averaged degree are assigned into training, those that have averaged degree large than 50-th percentile while smaller than 80-th percentile are assigned to validation set, and the left are assigned to test set. For **Graph-SST2** and **Graph-SST5** we follow the above process while for **Twitter** we conduct the above split in an inversed order to study the OOD generalization ability of GNNs trained on large degree graphs to small degree graphs.

DrugOOD datasets. To evaluate the OOD performance in realistic scenarios with realistic distribution shifts, we also include three datasets from DrugOOD benchmark (Ji et al., 2022). DrugOOD is a systematic OOD benchmark for AI-aided drug discovery, focusing on the task of drug target binding affinity prediction for both macromolecule (protein target) and small-molecule (drug compound). The molecule data and the notations are curated from realistic ChEMBL database (Mendez et al., 2019). Complicated distribution shifts can happen on different assays, scaffolds and molecule sizes. In particular, we select `DrugOOD-lbap-core-ec50-assay`, `DrugOOD-lbap-core-ec50-scaffold`, and `DrugOOD-lbap-core-ec50-size`, from the task of Ligand Based Affinity Prediction which uses `ic50` measurement type and contains `core` level annotation noises. We directly use the data files provided by the authors. [3] For more details, we refer interested readers to Ji et al. (2022).

E.2. Baselines and Evaluation Setup

During the experiments, we do not tune the hyperparameters exhaustively while following the common recipes for optimizing GNNs. Details are as follows.

GNN encoder. For fair comparison, we use the same GNN architecture as graph encoders for all methods. By default, we use 3-layer GIN (Xu et al., 2019) with Batch Normalization (Ioffe & Szegedy, 2015) between layers and JK residual connections at the last layer (Xu et al., 2018). The hidden dimension is set to 32 for Two-piece graphs, CMNIST-sp, and 128 for SST5, Twitter, and DrugOOD datasets. The pooling is by default a mean function over all nodes. The only exception is DrugOOD, where we follow the backbone used in the paper (Ji et al., 2022), i.e., 4-layer GIN with sum readout.

Interpretable GNN backbone. As mentioned in Sec. 2 that most of the existing invariant graph learning approaches adopt the interpretable GNN as the basic backbone model for the whole predictor $f = f_c \circ g$, where $g : \mathcal{G} \to \mathcal{G}_c$ is a featurizer GNN and $f_c : \mathcal{G}_c \to \mathcal{Y}$ is a classifier GNN. g first calculates the sampling weights as in $\widehat{G}_c$ for each edge. More formally,

[3] https://drugood.github.io/

given a graph G containing n nodes, a soft mask is predicted through the following equation:

$$Z = \text{GNN}(G) \in \mathbb{R}^{n\times h},\ M = \text{a}(Z, A) \in \mathbb{R}^{n\times n},$$

where a calculates the sampling weights for each edge using a MLP: $M_{ij} = \text{MLP}([Z_i, Z_j])$. Based on the continuous sampling score M, g could sample discrete edges according to the predicted scores. For two-piece graph datasets and EC50-Assay, EC50-Scaffold, EC50-Size,, we will directly use the score to reweight the messaging passing process along the edge, as we empirically find it yields more stable performance. While for CMNIST-sp, Graph-SST2, Graph-SST5, Twitter, we will sample a ratio $r\%$ of all edges for each graph. The ratios adopted are $80\%, 60\%, 50\%, 60\%$, respectively, following previous works (Chen et al., 2022a; Ji et al., 2022).

Besides, we also have various implementation options for obtaining the features in $\widehat{G}_c$, for further obtaining $h_{\widehat{G}_c}$, as well as for obtaining predictions based on $\widehat{G}_s$. By default, we feed the graph representations of featurizer GNN to the classifier GNN, as well as to the contrastive loss. For classifying G based on $\widehat{G}_s$, we use a separate MLP downstream classifier in the classifier GNN f_c.

Optimization and model selection. By default, we use Adam optimizer (Kingma & Ba, 2015) with a learning rate of $1e-3$ and a batch size of 128 for all models at all datasets. Except for CMNIST-sp, we use a batch size of 256 to facilitate the evaluation following previous works (Miao et al., 2022). To avoid underfitting, we pre-train models for 20 epochs for all datasets by default. While in two-piece graphs, we find pre-training by 100 epochs yields more stable performance. To avoid overfitting, we also employ an early stopping of 5 epochs according to the validation performance. Meanwhile, dropout is also adopted for some datasets. Specifically, we use a dropout rate of 0.5 for CMNIST, Graph-SST2, Graph-SST5, Twitter, EC50-Assay and EC50-Scaffold, 0.1 for EC50-Size according to the validation performance, following previous works (Chen et al., 2022a).

The final model is selected according to the performance at the validation set. All experiments are repeated with 5 different random seeds of $\{1, 2, 3, 4, 5\}$. The mean and standard deviation are reported from the 5 runs.

Implementations of Euclidean OOD methods. When implementing IRM (Arjovsky et al., 2019), V-Rex (Krueger et al., 2021) and IB-IRM (Ahuja et al., 2021), we refer the implementations from DomainBed (Gulrajani & Lopez-Paz, 2021). Since the environment information is not available, we perform random partitions on the training data to obtain two equally large environments for these objectives following previous works (Creager et al., 2021a; Chen et al., 2022a). Moreover, we select the weights for the corresponding regularization from $\{0.01, 0.1, 1, 10, 100\}$ for these objectives according to the validation performances of IRM and stick to it for others, since we empirically observe that they perform similarly with respect to the regularization weight choice. For EIIL (Creager et al., 2021b), we use the author released implementations about assigning different samples the weights for being put in each environment and calculating the IRM loss.

Implementations of invariant graph learning methods. We implement GSAT (Miao et al., 2022), GREA (Liu et al., 2022), MoleOOD (Yang et al., 2022), GIL (Li et al., 2022), DisC (Fan et al., 2022), and CIGA (Chen et al., 2022a), according to the author provided codes (if available). As for the hyperparameters in each method, we use a penalty weight of 1 and a ratio of 0.7 for GSAT following the author-recommended implementations. We use a penalty weight of 1 for GREA as we empirically it does not affect the performance by changing to different weights. We tune the penalty weights of MoleOOD with values from $\{1e-2, 1e-1, 1, 10\}$ but did not observe much performance differences. Hence we stick the penalty weight as 1 for all datasets. We tune the penalty weights of GIL with values from $\{1e-5, 1e-3, 1e-1\}$ recommended by the authors. For DisC, we tune only the q weight from $\{0.9, 0.7, 0.5\}$ in the GCE loss as we did not observe performance differences by changing the weight of the other term. We tune the penalty weight of CIGA with values from $\{0.5, 1, 2, 4, 8, 16, 32\}$ as recommended by the authors.

All of the graph learning methods adopt a interpretable GNN as the backbone by default. The only exception is MoleOOD, we follow the original implementation while using a shared GNN encoder for the variational losses to ensure the fairness of comparison. Besides, for DisC, we find the soft masking implementation in two-piece graphs will incur a sever performance degeneration hence we use a ratio of 25% for the interpretable GNN backbone.

For environment inferring methods, we fix the number of environments in two-piece graphs as 3 (since there are 3 spurious graphs), while search the number of inferred environments according to the validation performance. Specifically, in CMNIST, Graph-SST2, Graph-SST5, we search the number of environments from $\{2, 3, 4\}$ following previous practice (Li et al., 2022). In DrugOOD datasets, we search the number of environments from $\{2, 5, 10, 20, 100\}$ following previous practice (Yang et al., 2022).

Implementations of GALA. For a fair comparison, GALA uses the same GNN architecture for GNN encoders as the baseline methods. By default, we fix the temperature to be 1 in the contrastive loss, and merely search the penalty weight of the contrastive loss from $\{0.5, 1, 2, 4, 8, 16, 32\}$ according to the validation performances, following the CIGA implementations (Chen et al., 2022a). By default, we implement the environment assistant as a ERM model, and adopt directly the environment assistant predictions to sample possible and negative graph pairs. Nevertheless, as discussed in Sec. 4 that there could be multiple implementation choices for the environment assistant and the use of its predictions. In experiments, we find that using specific implementations could improve the OOD performance. For two-piece graphs, we implement the environment assistant model as an interpretable GNN with a ratio of 30% and adopt the cluster predictions of the graph representations of the environment assistant model to sample positive and negative pairs. Since GALA imposes a strong regularization to the data that may hinder the learning of graph representations, we pre-train the model by 10 epochs using ERM and then impose the GALA penalty implemented as one-side contrastive loss as discussed in Sec. D. For CMNIST-sp, we find implementing the environment assistant model as an interpretable GNN trained with ERM yields better performance. For Graph-SST2, Graph-SST5, Twitter, and DrugOOD datasets, we implement the environment assistant as a ERM model while clustering the learned graph representations of the model to sample postive and negative pairs.

E.3. Software and Hardware

We implement our methods with PyTorch (Paszke et al., 2019) and PyTorch Geometric (Fey & Lenssen, 2019). We ran our experiments on Linux Servers installed with V100 graphics cards and CUDA 10.2.