Empowering Graph Invariance Learning with Deep Spurious Infomax

Tianjun Yao [*1] **Yongqiang Chen** [*1 2] **Zhenhao Chen** [1] **Kai Hu** [3] **Zhiqiang Shen** [1] **Kun Zhang** [1 3]

Abstract

Recently, there has been a surge of interest in developing graph neural networks that utilize the invariance principle on graphs to generalize the out-of-distribution (OOD) data. Due to the limited knowledge about OOD data, existing approaches often pose assumptions about the correlation strengths of the underlying spurious features and the target labels. However, this prior is often unavailable and will change arbitrarily in the real-world scenarios, which may lead to severe failures of the existing graph invariance learning methods. To bridge this gap, we introduce a novel graph invariance learning paradigm, which induces a robust and general inductive bias. The paradigm is built upon the observation that the infomax principle encourages learning spurious features regardless of spurious correlation strengths. We further propose the `EQuAD` framework that realizes this learning paradigm and employs tailored learning objectives that provably elicit invariant features by disentangling them from the spurious features learned through infomax. Notably, `EQuAD` shows stable and enhanced performance across different degrees of bias in synthetic datasets and challenging real-world datasets up to 31.76%.

1. Introduction

Despite the enormous success of Graph Neural Networks (GNNs) (Kipf & Welling, 2016; Xu et al., 2018; Veličković et al., 2018a), they generally assume that the testing and training graphs are independently sampled from an identical distribution, i.e., the I.I.D. assumption, which can not be guaranteed in many real-world applications (Hu et al., 2021; Koh et al., 2021; Huang et al., 2021). Therefore, it has drawn great attention from the community to overcome the Out-of-Distribution (OOD) generalization challenge that enables GNNs to generalize to new environments outside the training distributions. Recent studies incorporate the invariance principle from causality (Peters et al., 2015) into GNNs (Wu et al., 2022a; Chen et al., 2022), in which the rationale is to learn the invariant graph features or identify invariant subgraphs that only focus on the direct causes of the target label and discards the other features whose correlations with the target labels may change across different environments (e.g., graph sizes). Due to the non-Euclidean and abstraction nature of graph data, the environment labels for distinguishing distribution shifts are usually expensive or unavailable, as collecting these labels typically requires expert knowledge (Wu et al., 2022a; Chen et al., 2022). Therefore, existing approaches generally rely on intermediate partitions of invariant and spurious graph features either from input space (Fan et al., 2022; Chen et al., 2022; Wu et al., 2022b) or latent space (Li et al., 2022; Liu et al., 2022; Zhuang et al., 2023) and adopts additional assumptions in order to learn the desired graph invariant features.

A significant challenge arises in the application of these graph invariance learning algorithms, as the assumptions typically pose a strong prior about the joint distribution of spurious features S and class label Y, i.e., $\mathbb{P}(S, Y)$, in an implicit or explicit way. In real-world scenarios, however, $\mathbb{P}(S, Y)$ can vary arbitrarily, leading to varying correlation degrees between S and Y. Consequently, this variability may conflict with the assumptions underlying these algorithms, resulting in potential failures. For example, DisC (Fan et al., 2022) presumes a strong correlation between S and Y and uses the presumption to identify a biased graph. Then DisC contrasts against the separated biased subgraph to learn the unbiased graph features. On the other hand, environment inference (Yang et al., 2022; Li et al., 2022) and augmentation (Wu et al., 2022b; Liu et al., 2022) algorithms typically presume a weaker correlation between S and Y to accurately infer or augment the environments. However, the premise can be easily broken when spurious correlation strengths shift, and lead to the failure of graph invariance learning (Chen et al., 2023). The brittleness of relying on presumed correlations as inductive bias, raises a

[*]Equal contribution [1]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE [2]The Chinese University of Hong Kong, Hong Kong, China [3]Carnegie Mellon University, Pittsburgh, USA. Correspondence to: Tianjun Yao <tianjun.yao@mbzuai.ac.ae>, Zhiqiang Shen <zhiqiang.shen@mbzuai.ac.ae>, Kun Zhang <kun.zhang@mbzuai.ac.ae>.

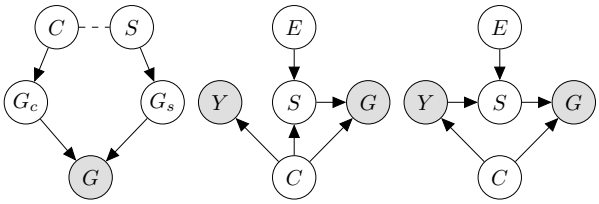
Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

challenging research question:

Is there a reliable inductive bias that remains robust against varying degrees of correlation between S and Y, and enhances graph invariance learning?

Present work. To address these challenges, we turn to self-supervised learning (SSL), which alleviates the dependence on Y, and eliminates the need for any assumption regarding $\mathbb{P}(S, Y)$. Notably, we show that employing global-local mutual information (MI) maximization, or the *infomax principle* (Hjelm et al., 2019; Veličković et al., 2018b; Linsker, 1988; Bell & Sejnowski, 1995) as the self-supervision objective, enables the model $f(\cdot)$ to capture spurious features with provable guarantees. Building on this insight, we have developed a new paradigm that decouples the learning of invariant features C and spurious features S. Specifically, we first learn representations containing S predominantly, then use these representations to uncover C. Furthermore, we propose a flexible framework **E**ncoding-**QuA**ntifying-**D**ecorrelation (`EQuAD`) in order to realize the novel learning paradigm, where off-the-shelf algorithms can serve as plug-ins for specific implementations for each step. `EQuAD` consists of the following three key steps: 1) *Encoding.* We first utilize infomax-based SSL to obtain graph representations fully encoding the spurious features S. 2) *Quantifying.* We then quantify the data samples into a low-dimensional latent space, which accurately captures the correlation degree between S and Y for each sample. 3) *Decorrealtion.* Finally, we retrain a GNN model from scratch to learn graph invariant representation by leveraging the spurious features obtained from previous steps. Our contributions can be summarized as follows:

- We reveal that self-supervised learning, when grounded in the infomax principle, can reliably isolate spurious features under certain mild conditions. In light of the finding, we introduce a new learning paradigm for graph invariance learning, which induces a robust inductive bias that relieves the reliance on presuming spurious correlation strengths between S and Y. (Sec. 4)
- We propose a flexible learning framework called `EQuAD` to realize the learning paradigm, as well as a new learning objective that is tailored for `EQuAD`, which provably learns the invariant graph representations. (Sec. 5)
- We conduct extensive experiments on 7 synthetic datasets and 8 real-world datasets with various types of distribution shifts. The results demonstrate the superiority of our method compared to state-of-the-art approaches. Notably, our method exhibits stable and enhanced performance across different degrees of bias in the synthetic datasets, and outperforms the baseline methods by an average of 31.76%. (Sec. 6). Our code is available at `https://github.com/tianyao-aka/EQuAD`.



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

Figure 1. Structural causal models for graph generation.

2. Preliminaries

2.1. Notations and Problem Definition

Notations. Throughout this work, we use C and S to denote contents (invariant factors) and styles (spurious factors) respectively, which are interchangeably with $\boldsymbol{h}_c$ (invariant representations) and $\boldsymbol{h}_s$ (spurious representations) in this work. $\widehat{C}$ and $\widehat{S}$ denote the estimated invariant and spurious factors, similarly for $\widehat{\boldsymbol{h}}_c$ and $\widehat{\boldsymbol{h}}_s$. We use $[K] := \{1, 2, \cdots, K\}$ to denote a index set, $f(\cdot)$ to denote a function, w to denote a scalar value, $\mathbf{w}$ and $\mathbf{W}$ to denote a vector and matrix, respectively. A more complete set of notations are presented in Appendix A.

Problem Definition. We focus on OOD generalization in graph classification. Given a set of graph datasets $\mathcal{G} = \{\mathcal{G}^e\}_{e \in \mathcal{E}_{\text{tr}} \subseteq \mathcal{E}_{\text{all}}}$, a GNN model f, denoted as $\rho \circ h$, comprises an encoder $h\colon \mathbb{G} \to \mathbb{R}^F$ that learns a representation $\boldsymbol{h}_G$ for each graph G, followed by a downstream classifier $\rho\colon \mathbb{R}^F \to \mathbb{Y}$ to predict the label $\widehat{Y}_G = \rho(\boldsymbol{h}_G)$. The objective of OOD generalization on graphs is to learn an optimal GNN model $f^*(\cdot) : \mathbb{G} \to \mathbb{Y}$ with data from training environments $\mathcal{G}_{\text{tr}} = \{\mathcal{G}^e\}_{e \in \mathcal{E}_{\text{tr}}}$ that effectively generalizes across all (unseen) environments:

$$f^*(\cdot) = \arg\min_{f} \sup_{e \in \mathcal{E}_{all}} \mathcal{R}(f \mid e), \tag{1}$$

where $\mathcal{R}(f \mid e) = \mathbb{E}^e_{\mathrm{G},\mathrm{Y}}[l(f(\mathrm{G}), \mathrm{Y})]$ is the risk of the predictor f on the environment e, and $l(\cdot, \cdot) : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}_+$ denotes a loss function.

2.2. Data Generating Process

We consider the graph generation process most widely discussed in the literature (Wu et al., 2022a; Chen et al., 2022; Miao et al., 2022; Fan et al., 2022; Li et al., 2022; Chen et al., 2023). As shown in Fig. 1, the observed graph G consists of an underlying invariant subgraph G_c and spurious subgraph G_s, which are generated under the control of the invariant

latent factor C and spurious latent factor S, respectively. C causally determines Y while S could be affected by the changes in the environment E. C, S and Y can exhibit two kinds of relations, i.e., Fully Informative Invariant Features (FIIF) when $Y \perp\!\!\!\perp S|C$ and Partial Informative Invariant Features (PIIF) when $Y \not\perp\!\!\!\perp S|C$. More details are included in Appendix B.

3. Related Work

Graph Invariance Learning. In recent years, there has been an increasing focus on learning graph representations that are robust to distribution shifts, especially from the perspective of invariant learning. Some works involve environment inference (Yang et al., 2022; Li et al., 2022) or environment augmentation (Wu et al., 2022b; Liu et al., 2022) algorithms, which infer environmental labels, or perform environment augmentation, and then use this information to learn graph invariant features. Another line of work adopts alternative strategies to achieve invariant learning, without directly dealing with the unobserved environments (Fan et al., 2022; Chen et al., 2022; 2023). For instance, CIGA (Chen et al., 2022) utilizes contrastive learning within the same class labels, assume samples with the same label share invariant substructures; DisC (Fan et al., 2022), conversely, leverages biased information to initially learn a biased graph for subsequent invariance learning. However, most of these methods often rely on strong assumptions about the joint distribution $\mathbb{P}(S, Y)$, which can lead to potential failures in real-world scenarios for OOD generalization. In this work, we propose a new learning paradigm which induces a robust inductive bias by eliminating the reliance on the correlation between S and Y.

Identifiability in Self-Supervised Learning. Self-supervised learning with augmentations has gained huge success in learning useful graph representations (Veličković et al., 2019; You et al., 2020). Existing analysis of self-supervised learning focuses on showing the desired property such as identifying the content from style (Kügelgen et al., 2021), or invariant subgraph from spurious one (Chen et al., 2022; 2023; Li et al., 2023). In contrast, we show that infomax principle tends to learn the spurious features, which can be leveraged to learn graph invariant features.

4. Learning Spurious Features with Self-supervision

In this section, we delve into how self-supervision can effectively identify spurious features with provable guarantee, which serves as *a key step in our proposed algorithm*. Concretely, We show that by employing a self-supervised approach based on the infomax principle, we can decouple the supervised learning and the identification of S, while in the meantime reducing the reliance on the presuming spurious correlation strengths between S and Y. First, we outline the infomax principle in Eqn. 2.

$$\max_{\theta} \frac{1}{|G||\mathcal{G}|} \sum_{G \in \mathcal{G}} \sum_{i \in |G|} I\left(\widehat{\mathbf{h}}_i; \widehat{\mathbf{h}}_G\right), \tag{2}$$

where $\widehat{\mathbf{h}}_i$ and $\widehat{\mathbf{h}}_g$ denote the node and graph representations respectively, and θ denotes the parameters of the encoder. The goal of Eqn. 2 is to maximize the MI between a global representation (e.g., a graph) and local parts of the inputs (e.g., nodes), which encourages the encoder to carry information presented in all locations. Intuitively, this maximization encourages the encoder to capture information presented across all locations. However, it is important to note that the global representations learned through Eqn. 2 might favor spurious correlations rather than causally-related high-level semantics, especially if the object of interest occupies a relatively small size within the global context. More formally, we present the following theorem:

Theorem 4.1. *Given the same data generation process as in Fig. 1 with Shannon entropy $H(S) = H(C) = \delta_f$, assuming the node representations encode proper information of the underlying latent factors, i.e., $\delta_r \geq I(\widehat{\boldsymbol{h}}_i; C) - I(\widehat{\boldsymbol{h}}_i; S) \geq \delta_l, \forall i \in G_c$ and $\delta_r \geq I(\widehat{\boldsymbol{h}}_i; S) - I(\widehat{\boldsymbol{h}}_i; C) \geq \delta_l, \forall i \in G_s$, the graph representation $\widehat{\boldsymbol{h}}_G \in \mathbb{R}^k$ have sufficient capacity to encode k independent features $\{\widehat{\boldsymbol{h}}_G[j]\}_{j=1}^k$ with $H(\widehat{\boldsymbol{h}}_G[j]) \leq \delta_f$, then, if $|G_s|/|G_c| > \delta_r/\delta_l$, the graph representation elicited by the infomax principle (Eqn. 2) exclusively contain spurious features S, i.e.,*

$$\mathbf{h}_s = \arg\max_{\theta} \frac{1}{|G||\mathcal{G}|} \sum_{G \in \mathcal{G}} \sum_{i \in |G|} I\left(\widehat{\mathbf{h}}_i; \widehat{\mathbf{h}}_G\right).$$

The proof of Theorem 4.1 is given in Appendix D.1. The characterization of features in the graph representation is motivated by the feature learning literature that neural networks tend to repeatedly encode features (Addepalli et al., 2023; Zhang & Bottou, 2023). The key observation from Eqn. 2 is based on the inductive bias that the spurious subgraph is usually *larger* than the invariant across a variety of applications. For example, the biochemical property of molecular graphs is usually determined by a small functional group in a molecule (Murray & Rees, 2009). On the other hand, the spurious subgraph such as the scaffold of the molecule usually takes a large part of the graph and easily biases the GNNs (Ji et al., 2022).

In addition to the proof, we also conducted a empirical study using SPMotif datasets (Wu et al., 2022b), following the method proposed in (Kirichenko et al., 2023). Specifically, we utilize the representations obtained through Eqn. 2 to examine the extent to which these representations contain invariant features, as illustrated in Figure 2: While the training accuracy is relatively high, there is a significant decrease in performance on the test set after feature

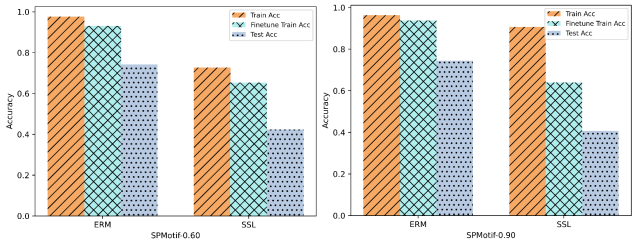


Figure 2. Investigation of the representation quality of ERM and Infomax-Based SSL in capturing spurious features. The experimental results implies that Infomax-Based SSL primarily learns spurious correlations. Further details on the experimental setup are provided in Appendix F.

reweighting (fine-tuning), compared to ERM. Specifically, the accuracy falls below 40% on both datasets. This decline implies that the representations derived from Eqn. 2 are predominantly composed of spurious features. More details about the experiments are included in Appendix F.

5. The EQuAD Framework

So far, we have discussed how to obtain representations that solely comprise $\mathbf{h}_s$. Next, we introduce our proposed learning paradigm: *Encoding, Quantifying, and Decorrelation*. This paradigm relies on the representations derived from infomax based self-supervised learning to achieve the decorrelation of $\mathbf{h}_s$ and $\widehat{\mathbf{h}}_c$, thereby obtaining invariant features for OOD generalization. The overall framework of EQuAD is illustrated in Figure 3.

First, we introduce the following learning objective for subsequent discussion, whose optimal solution under both FIIF and PIIF, can elicit invariant representations.

$$\max\ I\left(\widehat{\mathbf{h}}_c; Y\right), \text{s.t. } \widehat{\mathbf{h}}_c \perp\!\!\!\perp E, \widehat{\mathbf{h}}_c = h(G), E \subseteq \mathcal{E}_{tr}. \quad (3)$$

Eqn. 3 is widely adopted in previous works (Yang et al., 2022; Chen et al., 2022; Li et al., 2022; Wu et al., 2022a), hence we omit the proof. To effectively solve Eqn. 3, we first generate latent representations that maximally include S with self-supervised learning. Then, we establish connections between $\widehat{\mathbf{h}}_c$ and $\mathbf{h}_s$, and transform $\mathbf{h}_s$ to a low-dimensional space. Finally, we leverage $\mathbf{h}_s$ (in the low-dimensional space) to recover $\mathbf{h}_c$ with our proposed learning objective. Our approach is detailed as follows.

Step 1: Encoding. In the first step, we utilize Eqn. 2 to train an encoder $h(\cdot)$ for generating representations that predominantly contain $\mathbf{h}_s$. However, due to potential optimization errors or model architectures, the representations learned may encompass only a subset of $\mathbf{h}_s$. This limitation may impact the subsequent effectiveness of the decorrelation process. To mitigate this issue, we generate a collection of latent representations based on different training epochs and model architectures, i.e., $\mathcal{H} := \left\{\mathbf{H}^{(i,j)} \in \mathbb{R}^{N\times F} \mid i \in [K], j \in \mathcal{P}\right\}$, aiming to comprehensively cover spurious features. Here, $\mathcal{P}$ denote a set of pre-defined epochs, and K is the total number of model architectures, N is the data sample size and F is the embedding dimension.

Step 2:Quantifying. Having acquired a set of latent representations $\mathcal{H}$, we focus on the term $\widehat{\mathbf{h}}_c \perp\!\!\!\perp E$ in Eqn. 3. Assuming there exists an inverse and subjective function f_{spu}^{-1} such that $E = \mathrm{f}_{spu}^{-1}(S, C)$ under FIIF, and $E = \mathrm{f}_{spu}^{-1}(Y, C)$ under PIIF. To fulfill the condition $\widehat{\mathbf{h}}_c \perp\!\!\!\perp E$, we have the following optimization problem and its upper bound for FIIF:

$$\begin{aligned}
&\min_{\widehat{C}}\ I(E; \widehat{C}) \\
&= I\left(f_{spu}^{-1}(S, \widehat{C}); \widehat{C}\right) \\
&\leq I(S, \widehat{C}; \widehat{C}) \\
&= I(S; \widehat{C}) + I(\widehat{C}; \widehat{C} \mid S) \\
&\leq H(S) - H(S \mid \widehat{C}).
\end{aligned} \quad (4)$$

The upper bound for PIIF can be derived in a similar manner. The first upper bound can be obtained as f_{spu}^{-1} is a subjective function, and the second upper bound is due to that $I(\widehat{C}; \widehat{C} \mid S) > 0$. This upper bound is equivalent to the following optimization problem:

$$\max\ H\big(\mathbf{h}_s \mid \widehat{\mathbf{h}}_c\big). \quad (5)$$

However, given that $\mathbf{h}_s$ are high-dimensional vectors, maximizing $H(\mathbf{h}_s \mid \widehat{\mathbf{h}}_c)$ remains a challenging task in practice. To make Eqn. 5 more tractable, we first transform $\mathbf{h}_s$ into a more compact representation while preserving essential information, through the following approach: We employ ground-truth labels Y to train multiple classifiers $g : \mathbb{R}^F \to \mathbb{R}^{|\mathcal{C}|}$ (e.g., linear SVMs or MLPs) using $\mathcal{H}$ as inputs via ERM. Since $\mathcal{H}$ only contains a subset of S, $\mathbf{s}_i \in \mathbb{R}^{|\mathcal{C}|}$ can only depend on spurious features to make the prediction. Consequently, $\mathbf{s}_i$ can only reflect the correlation degree between the spurious pattern of sample i and its corresponding label. Therefore, $\mathbf{s}_i$ can serve as a more compact representation of $\mathbf{h}_s$, also revealing side information about the training environments. With this quantification stage, we obtain the logits matrix $\mathcal{S} := \left\{\mathbf{S}^{(i,j)} \in \mathbb{R}^{N\times|\mathcal{C}|} | \mathbf{S}^{(i,j)} = g(\mathbf{H}^{(i,j)}, Y), i \in [K], j \in \mathcal{P}\right\}$. Figure 4 illustrates logits distribution for *Cycle* class of SPMotif, which demonstrates the effectiveness of logits in identifying the spurious correlations for the data samples.

Step 3: Decorrelation. Having obtained $\mathcal{S}$, we first formulate a learning objective to facilitate decorrelation of $\widehat{\mathbf{h}}_c$ and $\mathbf{h}_s$. We further refine this learning objective to mitigate the data imbalance problem, thus achieving better OOD generalization capability. Now our goal is to maximize the entropy term $H\big(\mathbf{s}_i \mid \widehat{\mathbf{h}}_c^{(i)}\big), \forall i \in [N]$. To solve this problem, we

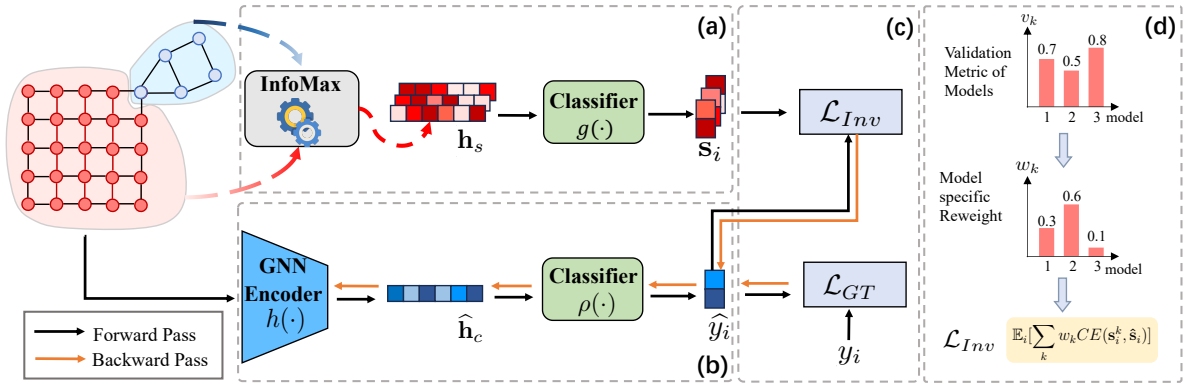


Figure 3. The overall framework of EQuAD. With an input graph consisting of G_c (shown in blue) and G_s (shown in red), the following procedures in EQuAD are illustrated: **(a)** Encoding and quantifying: First, the infomax-based SSL is performed to learn a collection of spurious representations (3 in this case), followed by $g(\cdot)$ to obtain the corresponding prediction logits as targets. **(b+c)** Decorrelation: In (b), a GNN encoder $h(\cdot)$ is re-trained from scratch to generate $\widehat{\boldsymbol{h}}_c$ followed by a classifier $\rho(\cdot)$ to get the prediction $\widehat{y}_i$; In (c), $\widehat{y}_i$ is fed into loss function $\mathcal{L} = \mathcal{L}_{GT} + \lambda \mathcal{L}_{Inv}$ for learning invariant features, where $\widehat{\mathbf{s}}_i$ and y_i serve as targets. **(d)** Detailed illustration of the process for model-specific reweighting. Finally, $\widehat{y}_i$ and $\widehat{\mathbf{s}}_i$ are obtained using the same classifier,i.e., $\rho(\cdot) = \rho'(\cdot)$.

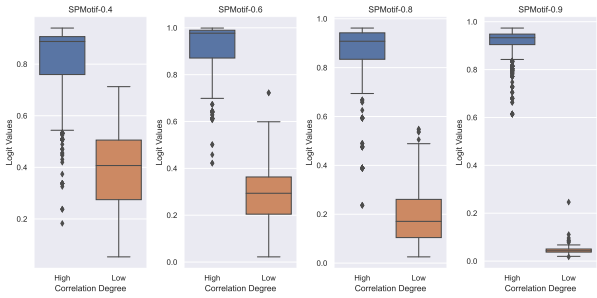


Figure 4. Logits distribution of SPMotif datasets for the *Cycle* class, where the samples are devided into two subgroups: the first subgroup consists of samples with a high correlation between S and Y, and the second subgroup contains samples with a low correlation between S and Y.

propose the following learning objective:

$$\min_{\rho,h} -\frac{1}{N}\sum_i\sum_j s_{ij}\log\left(\widehat{s}_{ij}\right), \quad \text{s.t. } \widehat{\mathbf{s}}_i = \rho\left(\widehat{\mathbf{h}}_c^{(i)}\right) = \rho\left(h\left(G_i\right)\right), i \in [N], j \in [|\mathcal{C}|]. \tag{6}$$

Here, $\widehat{\mathbf{s}}_i$ is the estimated logits vector, and s_{ij} is the scalar logit value for class j. $\mathbf{s}_i$ is the target normalized logits vector drawn from one of $\mathcal{S}$. In step 3, the encoder $h(\cdot)$ and classifier $\rho(\cdot)$ are trained from scratch to generate invariant representations. Here we offer an intuitive explanation of why Eqn. 6 achieve decorrelation of $\widehat{\mathbf{h}}_c$ and $\mathbf{h}_s$ by presenting a toy example as following.

Example. Considering a set of positive data points $\mathcal{D}_+$ with label $y = 1$, if $g(\cdot)$ divides $\mathcal{D}_+$ into two equal-sized subsets: $\mathcal{D}_+^1$ exploiting strongly correlated spurious patterns, and $\mathcal{D}_+^2$ without such patterns. Assume $\mathcal{D}_+^1$ has prediction logits of the form $(p, 1-p)$, e.g., $(0.9, 0.1)$, while $\mathcal{D}_+^2$ has the inverse logits $(1-p, p)$, e.g., $(0.1, 0.9)$. It can be shown that the estimated logits $\widehat{\mathbf{s}}_i$ minimizing Eqn. 6 is $(0.5, 0.5)$. In other words, $H\left(\mathbf{s}_i \,|\, \widehat{\mathbf{h}}_c^{(i)}\right)$ is maximized, regardless of the value of p. Formally, we present the following theorem:

Theorem 5.1. *Let $s_{iy} \in \mathbb{R}_+$ denote the logits value for samples whose class label $Y = y$ and belonging to bin B_i, and $d(\cdot,\cdot)$ denote the distance function. Assuming that for each class label $Y = y$, there exists $2K$ bins $B_i \in \mathcal{B}, \forall i \in [2K]$ with equal sample size, furthermore, the $2K$ bins can be arranged into K pairs of bins that are symmetrically located around the value 0.5, i.e., $|s_{iy}| = |s_{(i+K)y}|, \forall i \in [K]$, and $d(\mathbb{E}[s_{iy}], 0.5) = d(\mathbb{E}[s_{(i+K)y}], 0.5)$, where $sign(\mathbb{E}[s_{iy}] - 0.5) = sign(0.5 - \mathbb{E}[s_{(i+K)y}])$. Under these conditions, Eqn. 6, serving as a penalty term for ERM, achieves unique and optimal solution, i.e., $\widehat{\mathbf{h}}_c = \mathbf{h}_c$ and $\widehat{\mathbf{h}}_c \perp\!\!\!\perp \mathbf{h}_s$.*

A formal proof is provided in Appendix D.2, where we prove for both necessity and sufficiency for the optimal solution for Eqn. 6.

Mitigating data imbalance. Theorem 5.1 assumes an equal distribution of samples across all bins for every label Y. However, in real-world scenarios, certain spurious patterns, which are highly correlated with Y, often dominate, leading to a disproportionate accumulation of samples in specific bins. This data imbalance can hurt the optimality of Eqn. 6. To address this, we propose a novel sample reweighting approach that increases the weight of the minority group to achieve a more balanced distribution of data across different levels of correlation under the same target label Y, thereby facilitating the decorrelation of $\widehat{\mathbf{h}}_c$ and $\mathbf{h}_s$. Specifically, we adopt the following reweighting function: $w_{iy} := w\left(s_{iy};\gamma\right) = \frac{1-s_{iy}^{\gamma}}{\gamma}$, $s.t.\ \forall i \in [N], Y(i) = y$, where γ is a hyperparameter to control the smoothness of

the function. For all samples associated with the same y, $w(\cdot)$ will assign greater weight to samples where s_{iy} is closer to zero, indicating that these are minority samples whose spurious features less frequently co-occur with the prediction. Moreover, considering that a single prediction logits matrix $\mathbf{S} \subset \mathcal{S}$ might only capture a subset of the spurious features, thus limiting the identification of $\mathbf{h}_s$, we draw multiple logits matrices from $\mathcal{S}$ for better prediction of correlation degree of spurious features. Finally, we introduce a model-specific reweighting strategy to direct the loss term towards higher-quality target logits vectors $\mathbf{s}_i^k, \forall k \in [K]$, where K represents the number of prediction logits matrices. The quality of $\mathbf{S}_k$ is assessed based on the validation metric: *A lower validation metric indicates a reduced effectiveness of invariant features, thereby more accurately reflecting the correlation degree of spurious patterns.* Specifically, we adopt temperature-scaled softmax function for model-specific reweighting, i.e., $w_k := m(v_k;\tau) = \frac{\exp\left(\frac{-v_k}{\tau}\right)}{\sum_{j=1}^{K}\exp\left(\frac{-v_j}{\tau}\right)}$, where τ is the temperature, and v_k is the validation metric from model k. Finally, we arrive at the following refined objective:

$$\min_{\rho,h} -\frac{1}{NK}\sum_i\sum_j\sum_k w_{ij}w_k s_{ij}^k \log\left(\widehat{s}_{ij}\right),$$
$$\text{s.t. } \widehat{\mathbf{s}}_i = \rho\left(\widehat{\mathbf{h}}_c^{(i)}\right) = \rho\left(h\left(G_i\right)\right), i \in [N], j \in [|C|], k \in [K]. \tag{7}$$

Here w_{ij} and w_k are the sample-specific and model-specific reweighting coefficients respectively. Finally, to obtain the K prediction logits matrices from $\mathcal{S}$, we can also use the validation metric (e.g., validation accuracy) as a measure to gauge the extent of spurious features in the representations. Let $\mathcal{L}_{Inv}(\rho, h)$ denote Eqn. 7, and let $\mathcal{L}_{GT}$ denote the supervised training loss, i.e., $\mathcal{L}_{GT}(\rho', h) = \sum_{i=1}^{N} CE(y_i, \widehat{y}_i)$, where CE denote the cross-entropy loss for ground-truth label y_i and predicted class label $\widehat{y}_i$. In the specific implementation, $\rho(\cdot)$ and $\rho'(\cdot)$ share the same model parameters. Finally, the loss function for EQuAD is:

$$\mathcal{L} = \mathcal{L}_{GT} + \lambda\mathcal{L}_{Inv}, \tag{8}$$

where λ controls the strength of the decorrelation loss $\mathcal{L}_{Inv}$.

6. Experiments

In this section, we conduct extensive experiments to answer the following research questions.

- **RQ1)** Does EQuAD achieve better or comparable predictive performance than state-of-the-art methods?
- **RQ2)** How can we examine and interpret the latent representation induced by the GNN encoder in step 3?
- **RQ3)** How does each component in EQuAD contributes to the final performance?

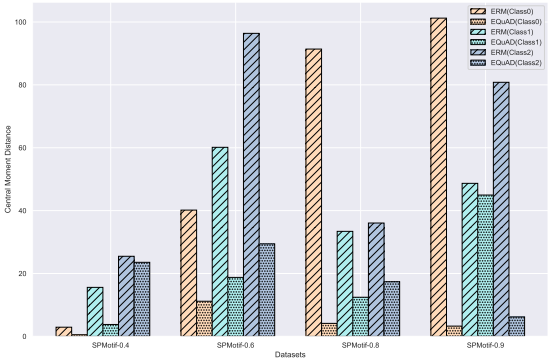


Figure 5. Investigation of the central moment distance of latent embeddings from training and validation set respectively using ERM and EQuAD. For all three classes across various SPMotif datasets, the distances for representations obtained via EQuAD are notably smaller compared to those from ERM.

6.1. Experimental Setting

Datasets. To comprehensively evaluate our proposed method under two data generating assumptions, namely FIIF and PIIF, we utilize the SPMotif datasets (Wu et al., 2022b) and Two-piece graph datasets (Chen et al., 2023) to verify its effectiveness. Additionally, for real-world datasets, we employ the DrugOOD datasets (Ji et al., 2022), which focus on the challenging task of AI-aided drug affinity prediction. We adopt 6 DrugOOD subsets, including splits using Assay, Scaffold, and Size from the IC50 and EC50 category respectively. Moreover, we also consider two molecule datasets, MolBACE and MolBBBP, from Open Graph Benchmark (Hu et al., 2021), where different molecules are structurally separated into different subsets, which provides a more realistic estimate of model performance in experiments (Wu et al., 2018b). More details about datasets are included in Appendix I.1.

Baseline methods. Besides ERM (Vapnik, 1995), we compare our method with state-of-the-art OOD methods from the Euclidean regime, including IRM (Arjovsky et al., 2020), VREx (Krueger et al., 2021), EIIL (Creager et al., 2021), IB-IRM (Ahuja et al., 2022), Coral (Sun & Saenko, 2016) and MixUp (Zhang et al., 2018). For graph-specific algorithms, we include GREA (Liu et al., 2022), DIR (Wu et al., 2022b), GSAT (Miao et al., 2022), CAL (Sui et al., 2022), DisC (Fan et al., 2022), MoleOOD (Yang et al., 2022), GIL (Li et al., 2022), CIGA (Chen et al., 2022), GALA (Chen et al., 2023) and iMoLD (Zhuang et al., 2023) as strong competitive baseline methods. For all baseline methods and EQuAD, we use GIN (Xu et al., 2019) as backbone encoder, and use Adam (Kingma & Ba, 2017) as the optimizer, for fair comparisons.

Evaluation. For SPMotif datasets and Two-piece graph datasets, the task is a 3-class classification, we adopt *accuracy* as the evaluation metric. For DrugOOD datasets

Table 1. Experiment results on synthetic datasets.

Methods	SPMotif				Two-piece graph		
	$b = 0.33$	$b = 0.40$	$b = 0.60$	$b = 0.90$	$(0.80, 0.70)$	$(0.80, 0.90)$	$(0.70, 0.90)$
ERM	53.40±2.20	62.19±3.26	55.24±2.43	49.41±3.78	75.65±1.62	51.37±1.20	42.73±3.82
IRM	58.31±2.59	55.71±6.37	58.76±1.98	42.11±4.14	75.13±0.77	50.76±2.56	41.32±2.50
V-Rex	56.12±4.76	60.08±4.11	58.91±4.45	42.32±3.48	74.96±1.40	49.47±3.36	41.65±2.78
IB-IRM	60.96±3.19	57.52±4.84	58.51±3.57	47.01±4.07	73.93±0.79	50.93±1.87	42.05±0.79
EIIL	59.87±2.19	57.73±5.09	53.42±3.84	42.58±5.42	74.25±1.74	51.45±4.92	39.71±2.64
GREA	59.27±3.45	62.46±4.28	61.04±5.21	58.63±1.52	82.72±0.50	50.34±1.74	39.01±1.21
GSAT	52.48±6.55	60.17±3.42	60.42±3.08	56.22±5.84	78.11±1.23	48.63±2.18	36.62±0.87
GIL	57.92±5.03	65.34±3.24	58.86±7.25	57.09±7.33	82.67±1.18	51.76±4.32	40.07±2.61
DisC	49.79±6.01	55.22±4.75	47.22±8.97	50.51±4.39	54.29±15.0	45.06±7.82	39.42±8.59
CIGA	72.91±1.92	67.96±5.27	67.31±6.84	58.87±5.93	83.21±0.30	57.87±3.38	43.62±3.20
GALA	66.96±5.18	65.38±3.68	63.25±3.11	62.07±2.20	**83.65±0.44**	62.25±3.71	49.65±3.93
EQuAD	**74.61±1.23**	**73.13±1.56**	**71.93±1.94**	**69.47±2.06**	82.76±0.71	**75.81±0.51**	**71.95±1.41**

and the two molecular datasets, we perform binary classification using *AUC* as the evaluation metric. To investigate the distribution discrepancy of two sets of embeddings, we adopt *central moment distance* (Zellinger et al., 2019) as a quantitative measure.

6.2. Main Results (RQ1)

Synthetic datasets. Our experimental results on two synthetic datasets are reported in Table 1. EQuAD demonstrates superior performance on these datasets across varying degrees of bias. The results from SPMotif indicate that EQuAD maintains stable performance under different levels of spurious correlation, consistently outperforming other baseline methods. In the context of PIIF, particularly when $H(S|Y) < H(C|Y)$, e.g., $(\alpha = 0.8, \beta = 0.9)$ and $(\alpha = 0.7, \beta = 0.9)$, environment inference algorithms(e.g., MoleOOD (Yang et al., 2022) and GIL (Li et al., 2022)) and environment augmentation algorithms (e.g., GREA (Liu et al., 2022) and DIR (Wu et al., 2022b)) all fail catastrophically. When the correlation between S and Y strengthens, these algorithms inevitably learn C in their representations, failing to accurately isolate S, which adversely affects subsequent invariance learning. Moreover, when $H(S|Y) < H(C|Y)$, CIGA's (Chen et al., 2022) objective fails to correctly identify G_c, and GSAT (Miao et al., 2022), based on the Information Bottleneck principle (Tishby & Zaslavsky, 2015), discards information about C as S reveals more information about Y. The key to EQuAD's success is its ability to identify S through self-supervised global-local mutual information maximization, which does not depend on label Y. Therefore, the strong association between S and Y does not impede the identification of S. It is noteworthy that GALA (Chen et al., 2023) employs data sampling for spurious subgraphs to address data imbalance. In our experiments, we exclude the data sampling technique for GALA, and similarly, EQuAD is tested without this strategy to ensure fair comparisons.

Real-world datasets. The effectiveness of EQuAD is further demonstrated through its performance on real-world datasets, as in Table 2. In these practical scenarios, EQuAD consistently outperforms other methods, achieving state-of-the-art results across most datasets. Notably, EQuAD shows significant improvements over both environment inference and environment augmentation algorithms in almost all datasets. This indicates that our approach, leveraging infomax based self-supervision, is more adept at identifying S, thereby facilitating the learning of invariant features. Furthermore, EQuAD consistently outperforms CIGA, which is known to reliably identify G_c when $H(C|Y) < H(S|Y)$. This highlights the effectiveness of EQuAD in directly learning invariant representations from the latent space. DisC also adopt a similar idea as EQuAD, i.e., to first identity spurious features (biased graphs), and then utilize the spurious features for the desired invariant features (learn invariant features on debiased graphs), however, DisC (Fan et al., 2022) relies on the strong correlation between S and Y to learn the biased graphs, which hinders its performance, while EQuAD resort to self-supervision to reliably extract spurious patterns which is orthogonal to $\mathbb{P}(S, Y)$. Although iMoLD (Zhuang et al., 2023) also employs a self-supervised objective for invariance learning to encourage the separation of S and C, their approach still relies on a parametric model with labeled supervision to obtain $\widehat{S}$. Consequently, $\widehat{S}$ may retain a subset of C, potentially compromising the efficacy of invariant feature learning.

6.3. Visualization (RQ2)

This section delves into the quality of the latent representation derived from the encoder $h(\cdot)$ in step 3. We explore the distribution discrepancy between latent embeddings from

Table 2. Experiment results on real-world datasets. `EQuAD` outperforms state-of-the-art methods in most of the datasets with various types of distribution shifts. First and second best methods are denoted by **bold** and underline respectively.

Method	IC50			EC50			BACE	BBBP
	Assay	Scaffold	Size	Assay	Scaffold	Size		
ERM	71.63±0.76	68.79±0.47	67.50±0.38	67.39±2.90	64.98±1.29	65.10±0.38	77.83±3.49	66.93±2.31
IRM	71.15±0.57	67.22±0.62	61.58±0.58	67.77±2.71	63.86±1.36	59.19±0.83	79.47±1.86	68.92±0.53
Coral	71.28±0.91	68.36±0.61	64.53±0.32	72.08±2.80	64.83±1.64	58.47±0.43	-	-
MixUp	71.49±1.08	68.59±0.27	67.79±0.39	67.81±4.06	65.77±1.83	65.77±0.60	-	-
V-Rex	71.32±1.17	67.30±1.27	64.46±0.79	75.57±2.17	64.73±0.53	62.80±0.89	-	-
IB-IRM	68.22±0.54	59.38±0.49	58.25±2.40	64.70±2.50	62.62±2.05	58.28±0.99	-	-
EIIL	70.58±1.56	67.02±0.46	61.58±0.58	64.20±5.40	62.88±2.75	59.58±0.96	-	-
DIR	69.84±1.41	66.33±0.65	62.92±1.89	65.81±2.93	63.76±3.22	61.56±4.23	79.93±2.03	69.63±1.54
GSAT	70.59±0.43	66.45±0.50	66.70±0.37	73.82±2.62	64.25±0.63	62.65±1.79	79.63±1.87	68.48±2.01
GREA	70.23±1.17	67.02±0.28	66.59±0.56	74.17±1.47	64.50±0.78	62.81±1.54	**82.37±2.37**	69.70±1.28
CAL	70.09±1.03	65.90±1.04	66.42±0.50	74.54±4.18	65.19±0.87	61.21±1.76	-	-
DisC	61.40±2.56	62.70±2.11	61.43±1.06	63.71±5.56	60.57±2.27	57.38±2.48	-	-
MoleOOD	71.62±0.52	68.58±1.14	65.62±0.77	72.69±1.46	65.74±1.47	65.51±1.24	81.09±2.03	69.84±1.84
CIGA	71.86±1.37	69.14±0.70	66.92±0.54	69.15±5.79	67.32±1.35	65.65±0.82	80.98±1.25	69.65±1.32
iMoLD	72.11±0.51	68.84±0.58	67.92±0.43	77.48±1.70	67.79±0.88	**67.09±0.91**	-	-
EQuAD	**73.26±0.47**	**69.78±0.41**	**68.19±0.24**	**79.36±0.73**	**68.12±0.48**	66.37±0.64	81.83±2.67	**71.22±1.47**

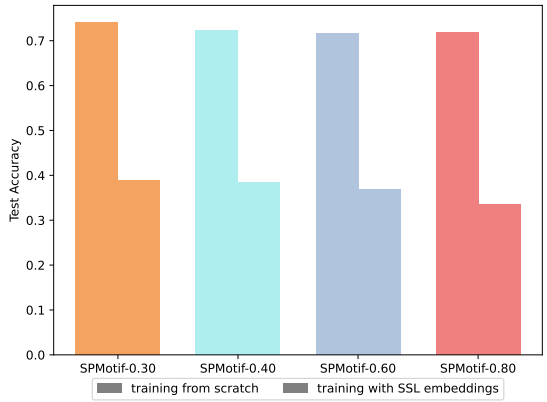


Figure 6. Ablation study on latent representations, which demonstrates the necessity of training a model $f(\cdot)$ from scratch for effective invariance learning.

the training set and those from the validation set. More visualizations on latent representation with t-SNE(van der Maaten & Hinton, 2008) plots in a 2-dimensional space are provided in Appendix I.4.

Analysis of Distribution Discrepancy in Latent Representations. We investigate the discrepancy in the distribution of latent representations between the training and validation sets. To quantify this discrepancy, we employ *the central moment distance* (CMD), a metric proposed in (Zellinger et al., 2019) for domain-invariant representation learning in neural networks. As illustrated in Figure 5, across all three ground-truth labels, the representations obtained through EQuAD exhibited relatively smaller distribution discrepancies between training and validation sets compared to those obtained through ERM. This suggests EQuAD can achieve superior OOD generalization capabilities.

Table 3. Ablation study on model-specfic reweighting.

	EC50-Assay	EC50-Scaffold	EC50-Size
w/ weighted	**79.36±0.63**	**68.12±0.48**	**66.37±0.64**
w/ averaging	78.03±2.24	67.45±0.32	66.13±0.38
w/ single	77.46±0.29	67.68±0.17	66.02±0.41

6.4. Ablation Study (RQ3)

In this section, we conduct two ablation study to further validate our design choice in EQuAD.

Adaptive Reweighting vs. Averaging. We perform ablation study to investigate the effectiveness of using multiple prediction logits matrices from $\mathcal{S}$ compared to a single logits matrix. Additionally, we explore the benefits of model-specific adaptive reweighting over simple averaging. As reported in Table 3, employing multiple logits matrices consistently outperform the use of a single logits matrix, which may be due to the different spurious patterns learned by distinct logits matrices by $g(\cdot)$, enabling a more accurate estimation of the correlation degree between S and label Y. Furthermore, leveraging feedback based on validation metrics allows for a preference towards logits matrices of higher quality, thereby yielding improved empirical results.

The Necessity of Training an Encoder from Scratch. As observed in Figure 6, embeddings derived from infomax-based SSL struggles to extract high-quality invariant features and results in low test accuracy, even trained with

decorrelation loss. In contrast, training an encoder $f(\cdot)$ from scratch can extract higher-quality invariant representations and lead to better graph invariance learning. This again indicates that infomax-based SSL primarily learns spurious features, and highlights the effectiveness of our proposed learning paradigm, which learns both invariant features and spurious features via ERM (Kirichenko et al., 2023), and remove spurious features in step 3 which are learned from SSL.

7. Conclusions

We have shown that the infomax principle effectively extracts spurious features with provable guarantees, which motivates us to design a novel learning paradigm and the development of a flexible framework, `EQuAD`. `EQuAD` induces a robust inductive bias by eliminating the reliance on strong assumptions about the correlation strengths between spurious features and class labels. Our approach significantly outperforms state-of-the-art methods in most synthetic and real-world datasets, demonstrating the effectiveness and robustness of `EQuAD` for OOD generalization on graphs. This learning paradigm holds great potential for adaptation to other data modalities, such as vision and natural language, which we leave for our future work.

Impact Statement

This paper is dedicated to advancing research in OOD Generalization, with a specific focus on graph data. The learning paradigm we propose holds the potential for broad application beyond graph data, extending to other data forms such as vision and natural language.

In practical applications, distribution shifts between testing and training data are inevitable, and traditional graph machine learning approaches often suffer from significant performance degradation under such shifts. Developing methods with robust OOD generalization capabilities is thus critically important, especially for high-stakes graph applications, including but not limited to molecule prediction (Hu et al., 2021), financial analysis (Yang et al., 2020), criminal justice (Agarwal et al., 2021), autonomous driving (Liang et al., 2020), and drug discovery (Gaudelet et al., 2021). The ability to maintain performance consistency across varying distributions can have profound implications in these areas, potentially leading to more reliable predictions and analyses that can inform decision-making processes and contribute to advancements in these fields. Besides, this paper does not raise any ethical concern or potentially harmful insight.

Acknowledgements

KZ would like to acknowledge the support from NSF Grant 2229881, the National Institutes of Health (NIH) under Contract R01HL159805, and grants from Apple Inc., KDDI Research Inc., Quris AI, and Florin Court Capital. We sincerely appreciate the insightful discussions with the anonymous reviewers, which greatly helped to improve our work.

References

Addepalli, S., Nasery, A., Radhakrishnan, V. B., Netrapalli, P., and Jain, P. Feature reconstruction from outputs can mitigate simplicity bias in neural networks. In *International Conference on Learning Representations*, 2023.

Agarwal, C., Lakkaraju, H., and Zitnik, M. Towards a unified framework for fair and stable graph representation learning, 2021.

Ahuja, K., Shanmugam, K., Varshney, K. R., and Dhurandhar, A. Invariant risk minimization games, 2020.

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, 2022.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization, 2020.

Bell, A. J. and Sejnowski, T. J. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995. URL https://api.semanticscholar.org/CorpusID:1701422.

Ben-Tal, A., den Hertog, D., Waegenaere, A. D., Melenberg, B., and Rennen, G. Robust solutions of optimization problems affected by uncertain probabilities. *Advanced Risk & Portfolio Management® Research Paper Series*, 2011. URL https://api.semanticscholar.org/CorpusID:761793.

Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. S. Invariant rationalization, 2020.

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, 2022.

Chen, Y., Bian, Y., Zhou, K., Xie, B., Han, B., and Cheng, J. Does invariant graph learning via environment augmentation learn invariance? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=EqpR9Vtt13.

Creager, E., Jacobsen, J.-H., and Zemel, R. Environment inference for invariant learning, 2021.

Duchi, J. and Namkoong, H. Learning models with uniform performance via distributionally robust optimization, 2020.

Fan, S., Wang, X., Mo, Y., Shi, C., and Tang, J. Debiasing graph neural networks via learning disentangled causal substructure, 2022.

Feng, J., Chen, Y., Li, F., Sarkar, A., and Zhang, M. How powerful are k-hop message passing graph neural networks. *arXiv preprint arXiv:2205.13328*, 2022.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric, 2019.

Gao, R., Chen, X., and Kleywegt, A. J. Wasserstein distributionally robust optimization and variation regularization, 2020.

Gaudelet, T., Day, B., Jamasb, A. R., Soman, J., Regep, C., Liu, G., Hayter, J. B., Vickers, R., Roberts, C., Tang, J., et al. Utilizing graph machine learning within drug discovery and development. *Briefings in bioinformatics*, 22(6):bbab159, 2021.

Hassani, K. and Khasahmadi, A. H. Contrastive multi-view representation learning on graphs, 2020.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization, 2019.

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, 2021.

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., 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, 2021.

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, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations, 2023.

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. A., Haque, I. S., Beery, S., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. Wilds: A benchmark of in-the-wild distribution shifts, 2021.

Kondor, R. Diffusion kernels on graphs and other discrete structures. In *International Conference on Machine Learning*, 2002. URL https://api.semanticscholar.org/CorpusID:8606662.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex), 2021.

Kügelgen, J. V., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., and Locatello, F. Self-supervised learning with data augmentations provably isolates content from style. In *Advances in Neural Information Processing Systems*, 2021.

Lee, J. and Raginsky, M. Minimax statistical learning with wasserstein distances, 2018.

Li, H., Zhang, Z., Wang, X., and Zhu, W. Learning invariant graph representations for out-of-distribution generalization. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=acKK8MQe2xc.

Li, Z., Xu, Z., Cai, R., Yang, Z., Yan, Y., Hao, Z., Chen, G., and Zhang, K. Identifying semantic component for robust molecular property prediction. *arXiv preprint*, arXiv:2311.04837, 2023.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks, 2020.

Linsker, R. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988. doi: 10.1109/2.36.

Liu, G., Zhao, T., Xu, J., Luo, T., and Jiang, M. Graph rationalization with environment-based augmentations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22. ACM, Au-

gust 2022. doi: 10.1145/3534678.3539347. URL http://dx.doi.org/10.1145/3534678.3539347.

Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. Heterogeneous risk minimization, 2021.

Maron, H., Litany, O., Chechik, G., and Fetaya, E. On learning sets of symmetric elements. In *International Conference on Machine Learning*, pp. 6734–6744. PMLR, 2020.

Miao, S., Liu, M., and Li, P. Interpretable and generalizable graph learning via stochastic attention mechanism, 2022.

Murray, C. and Rees, D. The rise of fragment-based drug discovery. *Nature chemistry*, 1:187–92, 06 2009.

Nikolentzos, G., Dasoulas, G., and Vazirgiannis, M. k-hop graph neural networks. *Neural Networks*, 130:195–205, 2020.

Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking : Bringing order to the web. In *The Web Conference*, 1999. URL https://api.semanticscholar.org/CorpusID:1508503.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, 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, 2019.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference using invariant prediction: identification and confidence intervals, 2015.

Platt, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Sui, Y., Wang, X., Wu, J., Lin, M., He, X., and Chua, T.-S. Causal attention for interpretable and generalizable graph classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22. ACM, August 2022. doi: 10.1145/3534678.3539366. URL http://dx.doi.org/10.1145/3534678.3539366.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation, 2016.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle, 2015.

van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Vapnik, V. N. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks, 2018a.

Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. Deep graph infomax, 2018b.

Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. Deep graph infomax. In *International Conference on Learning Representations*, 2019.

Weisfeiler, B. and Leman, A. The reduction of a graph to canonical form and the algebra which appears therein. *nti, Series*, 2(9):12–16, 1968.

Wu, Q., Zhang, H., Yan, J., and Wipf, D. Handling distribution shifts on graphs: An invariance perspective, 2022a.

Wu, Y.-X., Wang, X., Zhang, A., He, X., and Chua, T.-S. Discovering invariant rationales for graph neural networks, 2022b.

Wu, Z., Ramsundar, B., Feinberg, E., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.*, 9:513–530, 2018a. doi: 10.1039/C7SC02664A. URL http://dx.doi.org/10.1039/C7SC02664A.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: A benchmark for molecular machine learning, 2018b.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks?, 2019.

Xu, Y. and Jaakkola, T. Learning representations that support robust transfer of predictors, 2021.

Yang, N., Zeng, K., Wu, Q., Jia, X., and Yan, J. Learning substructure invariance for out-of-distribution molecular representations. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=2nWUNTnFijm.

Yang, S., Zhang, Z., Zhou, J., Wang, Y., Sun, W., Zhong, X., Fang, Y., Yu, Q., and Qi, Y. Financial risk analysis for smes with graph-based supply chain mining. In Bessiere, C. (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 4661–4667. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/643. URL https://doi.org/10.24963/ijcai.2020/643. Special Track on AI in FinTech.

Yao, T., Wang, Y., Zhang, K., and Liang, S. Improving the expressiveness of k-hop message-passing gnns by injecting contextualized substructure information. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3070–3081, 2023.

Ying, R., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks, 2019.

You, J., Gomes-Selman, J. M., Ying, R., and Leskovec, J. Identity-aware graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10737–10745, 2021.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, 2020.

Yu, J., Xu, T., Rong, Y., Bian, Y., Huang, J., and He, R. Graph information bottleneck for subgraph recognition, 2020.

Zadrozny, B. and Elkan, C. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pp. 609–616, 2001.

Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., and Saminger-Platz, S. Central moment discrepancy (cmd) for domain-invariant representation learning, 2019.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization, 2018.

Zhang, J. and Bottou, L. Learning useful representations for shifting tasks and distributions. In *International Conference on Machine Learning*, pp. 40830–40850, 2023.

Zhang, M. and Li, P. Nested graph neural networks, 2021.

Zhang, Z. and Sabuncu, M. R. Generalized cross entropy loss for training deep neural networks with noisy labels, 2018.

Zhao, L., Jin, W., Akoglu, L., and Shah, N. From stars to subgraphs: Uplifting any gnn with local structure awareness. *arXiv preprint arXiv:2110.03753*, 2021.

Zhu, Y., Xu, Y., Liu, Q., and Wu, S. An Empirical Study of Graph Contrastive Learning. *arXiv.org*, September 2021.

Zhuang, X., Zhang, Q., Ding, K., Bian, Y., Wang, X., Lv, J., Chen, H., and Chen, H. Learning invariant molecular representation in latent discrete space, 2023.

A. Notations

We present a set of notations used throughout our paper for clarity. Below are the main notations along with their definitions.

Table 4. Notation Table

Symbols	Definitions
$\mathcal{G}$	a set of graphs
$G = (A, X)$	a graph with the adjacency matrix $A \in \{0,1\}^{n\times n}$ and node feature matrix $X \in \mathbb{R}^{n\times d}$
Y	random variable for labels
C	content factor
S	style factor
E	environment
$\mathbf{h}_c$	invariant representations, interchangeably with C in our paper
$\mathbf{h}_s$	spurious representations, interchangeably with S in our paper
G_c	the invariant subgraph with respect to G
G_s	the spurious subgraph with respect to G
$\widehat{G}_c$	the estimated invariant subgraph
$\widehat{G}_s$	the estimated spurious subgraph
$\widehat{\mathbf{h}}_G$	the estimated graph representation for graph G
$\widehat{\mathbf{h}}_i$	the estimated node representation for node $i \in G$
$\widehat{\mathbf{h}}_c$	the estimated invariant graph representation, interchangeably with $\widehat{C}$ in our paper
$\widehat{\mathbf{h}}_s$	the estimated spurious graph representation, interchangeably with $\widehat{S}$ in our paper
$\widehat{C}$	the estimated content variable
$\widehat{S}$	the estimated style variable
$\mathbf{H}$	latent representations derived from global-local mutual information maximization
$\mathbf{S}$	prediction logits matrix derived from $\mathbf{H}$ and Y
$\mathcal{H}$	a set of $\mathbf{H}$
$\mathcal{S}$	a set of $\mathbf{S}$
$g(\cdot)$	classifier used in step 2 to generate spurious logits matrix $\mathbf{S}$
$h(\cdot)$	encoder
$\rho(\cdot)$	classifier
$[K] := \{1, 2, \cdots, K\}$	index set with K elements
w	a scalar value
$\mathbf{w}$	a vector
$\mathbf{W}$	a matrix

B. More Details on Data Generating Process

We detail about the underlying assumption of the data generating process in our work as illustrated in Figure 1. The graph generation process is illustrated using a latent-variable model perspective to elucidate, positing that a graph is generated through a mapping $f_{\text{gen}} : \mathcal{Z} \to \mathcal{G}$, where $\mathcal{Z} \subseteq \mathbb{R}^n$ represents the latent space and $\mathcal{G}$ denotes the graph space. Within this framework, we distinguish the latent variables into an content variable $C \in \mathcal{C}$ and a style variable $S \in \mathcal{S}$, based on their susceptibility to environmental influences E. The invariant and spurious components, C and S, respectively, govern the observed graphs' generation, with their interactions in the latent space leading to the emergence of Fully Informative Invariant Features (FIIF) and Partially Informative Invariant Features (PIIF), depends on the completeness of information C provides about the label Y.

The graph generation model is formalized through a Structural Causal Model (SCM) that decomposes f_{gen} into distinct functions controlling the generation of G_c, G_s, and G, as outlined in Eqn. 9. This decomposition allows for the isolation of invariant information within G_c, unaffected by environmental interventions, from the spurious and environment-sensitive information within G_s and G. Such a model reflects the reality that graphs from different domains may exhibit diverse structural and feature-level properties, all potentially spuriously correlated with labels depending on the underlying latent interactions. The SCM framework, compatible with a broad range of graph generation models, aims to characterize potential

distribution shifts without committing to specific graph families, thereby maintaining generality and applicability across various contexts.

$$G_c := f_{\text{gen}}^{G_c}(C), \quad G_s := f_{\text{gen}}^{G_s}(S), \quad G := f_{\text{gen}}^{G}\ (G_c, G_s) \tag{9}$$

Following previous work (Arjovsky et al., 2020; Ahuja et al., 2022; Chen et al., 2022), the latent interactions between C and S are categorized into Fully Informative Invariant Features (FIIF) and Partially Informative Invariant Features (PIIF) based on whether C fully informs about Y. In the context of FIIF (Eqn. 10), the invariant component C provides a complete and direct mapping to the label Y, making it fully informative for label prediction. This implies that, regardless of environmental variations or the presence of spurious correlations, a model can rely solely on the invariant features within C for accurate predictions. This scenario is ideal for OOD generalization, as it suggests that the core attributes necessary for classification are consistent across different domains. Previous OOD methods, such as GIB (Yu et al., 2020) and DIR (Wu et al., 2022b) and GSAT (Miao et al., 2022), have primarily focused on leveraging FIIF to ensure that models remain robust against distribution shifts by concentrating on invariant features that are not affected by environmental changes.

$$Y := f_{\text{inv}}(C), S := f_{\text{spu}}(C, E), G := f_{\text{gen}}(C, S). \tag{10}$$

PIIF (Eqn. 11), on the other hand, acknowledge that C only partially informs about Y. This necessitates the consideration of additional variables, potentially including the spurious component S or the environment E, to achieve accurate label prediction. In these cases, the model must discern how to integrate information from both invariant and spurious features, navigating the complex interactions that may indirectly influence the prediction of Y. The challenge here lies in identifying and mitigating the impact of spurious correlations that may be informative in a specific context but detrimental to generalization across environments. Methods like IRM (Arjovsky et al., 2020) focus on PIIF scenarios by attempting to learn representations that are predictive of Y across different environments, despite the presence of varying spurious correlations.

$$Y := f_{\text{inv}}(C), S := f_{\text{spu}}(Y, E), G := f_{\text{gen}}(C, S). \tag{11}$$

In this study, we provide a new perspective on how to learn graph invariance under both of these scenarios, and our proposed framework exhibits robust and superior performance under both scenarios with varying degrees of correlation strengths between S and Y.

C. More Background and Related Work

Graph Neural Networks. In this work, we adopt message-passing GNNs for graph classification due to their expressiveness. Given a simple and undirected graph $G = (A, X)$ 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 d feature dimensions, the graph encoder $h : \mathbb{G} \to \mathbb{R}^h$ aims to learn a meaningful graph-level representation h_G, and the classifier $\rho : \mathbb{R}^h \to \mathbb{Y}$ is used to predict the graph label $\widehat{Y}_G = \rho(h_G)$. To obtain the graph representation h_G, the representation $\mathbf{h}_v^{(l)}$ of each node v in a graph G is iteratively updated by aggregating information from its neighbors $\mathcal{N}(v)$. For the l-th layer, the updated representation is obtained via an AGGREGATE operation followed by an UPDATE operation:

$$\mathbf{m}_v^{(l)} = \text{AGGREGATE}^{(l)}\left(\left\{\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v)\right\}\right), \tag{12}$$

$$\mathbf{h}_v^{(l)} = \text{UPDATE}^{(l)}\left(\mathbf{h}_v^{(l-1)}, \mathbf{m}_v^{(l)}\right), \tag{13}$$

where $\mathbf{h}_v^{(0)} = \mathbf{x}_v$ is the initial node feature of node v in graph G. Then GNNs employ a READOUT function to aggregate the final layer node features $\left\{\mathbf{h}_v^{(L)} : v \in \mathcal{V}\right\}$ into a graph-level representation $\mathbf{h}_G$:

$$\mathbf{h}_G = \text{READOUT}\left(\left\{\mathbf{h}_v^{(L)} : v \in \mathcal{V}\right\}\right). \tag{14}$$

In this work, we adopt GIN (Xu et al., 2019) as backbone encoder, which is 1-WL (Weisfeiler & Leman, 1968) expressive. For more expressive GNN architectures, such as subgraph-based GNNs (Zhao et al., 2021; Zhang & Li, 2021; You et al.,

2021; Maron et al., 2020) and K-hop message-passing GNNs (Feng et al., 2022; Yao et al., 2023; Nikolentzos et al., 2020) as encoders, the OOD generalizability may further enhances as the expressive power increases.

Central Moment Discrepancy. Central Moment Discrepancy (CMD) is a probalistic metric for domain-invariant representation learning, particularly used in the context of domain adaptation with neural networks. The main goal of CMD is to minimize the discrepancy between domain-specific latent feature representations directly in the hidden activation space. Unlike traditional methods such as Maximum Mean Discrepancy (MMD), CMD explicitly aligns higher-order moments of probability distributions in an order-wise manner. The CMD method utilizes central moments to quantify the differences between probability distributions. Let P and Q be two probability distributions with their respective central moments $\mu_k(P)$ and $\mu_k(Q)$ for order k. The CMD between these distributions can be defined as: $\mathrm{CMD}(P, Q) = \sum_{k=1}^{K} |\mu_k(P) - \mu_k(Q)|$, where K is the highest order of moments considered. The central moments μ_k are defined as: $\mu_k(P) = \mathbb{E}_{x\sim P}\left[(x - \mathbb{E}_{x\sim P}[x])^k\right]$.

OOD Generalization. In the field of machine learning, especially in OOD settings, deep neural networks are known to exploit spurious features, leading to failures in generalization. Recently, there is an emerging line of work proposed to address this challenge. IRM (Arjovsky et al., 2020) aims to learn an optimal invariant association across diverse environment segments within the training data. Extending this concept, some studies have integrated multi-objective optimization with game theory (Ahuja et al., 2020; Chen et al., 2022) or adopted invariant representation learning through adversarial training using deep neural networks (Chang et al., 2020; Xu & Jaakkola, 2021); In parallel, Distributionally Robust Optimization (DRO) (Ben-Tal et al., 2011; Lee & Raginsky, 2018; Gao et al., 2020; Duchi & Namkoong, 2020) methods have been developed to enhance OOD generalization. These methods focus on training models to perform robustly against the worst-case loss among diverse data groups; However, both IRM and DRO typically necessitate explicit environment partitions within the dataset, a requirement that is often impractical in real-world scenarios. This limitation has motivated research into invariance learning where such explicit environment partitions are not available. Environment Inference for Invariant Learning (EIIL) (Creager et al., 2021) employs a two-stage training process involving biased model environment inference and subsequent invariance learning. Similarly, Heterogeneous Risk Minimization (HRM) (Liu et al., 2021) addresses this issue by simultaneously learning latent heterogeneity and invariance. However, most of these methods focus on Euclidean data, and cannot be trivially adapted to graph-structured data.

Graph Invariance Learning. In recent years, there has been an increasing focus on learning graph-level representations that are robust to distribution shifts, particularly from the perspective of invariant learning. This growing interest has led to the development of two lines of research in graph invariance learning algorithms. The first line of research involves environment inference (Yang et al., 2022; Li et al., 2022) or environment augmentation (Wu et al., 2022b; Liu et al., 2022) algorithms, which infer environmental labels, or perform environment augmentation, and then use this information to learn graph invariant features. Another line of work do not explicitly address the issue of unobserved environment labels. Instead, these methods adopt alternative strategies to achieve invariant learning. For instance, CIGA (Chen et al., 2022) utilizes contrastive learning within the same class labels, with the underlying assumption that samples with the same label share invariant substructures; DisC (Fan et al., 2022), conversely, leverages biased information to initially learn a biased graph, subsequently focusing on unbiased graphs for learning invariant features. However, these methods often rely on strong assumptions about the joint distribution $\mathbb{P}(S, Y)$, which can lead to potential failures in real-world scenarios. For example, DisC assume a strong correlation between S and Y to identify biased graph. While CIGA's ability to identify invariant subgraphs is contingent on a stronger correlation between C and Y. Similarly, environment inference and augmentation algorithms typically assume a weak correlation between S and Y; otherwise, C might be erroneously included in $\widehat{\mathbf{h}}_s$, leading to the failure of environment inference or augmentation.

In this work, we adopt a self-supervised learning approach based on the infomax principle to extract spurious features with provable guarantee. This method alleviates the dependence on the label Y, and reduces the reliance on the correlation between S and Y, establishing a new inductive bias that remains robust under varying degrees of correlation in $\mathbb{P}(S, Y)$.

Identifiability in Self-Supervised Learning. Self-supervised learning with augmentations has gained huge success in learning useful graph representations (Veličković et al., 2019; You et al., 2020). Existing analysis of self-supervised learning focuses on showing the desired property such as identifying the content from style (Kügelgen et al., 2021), or invariant subgraph from spurious one (Chen et al., 2022; 2023; Li et al., 2023). In contrast, we show that infomax principle tends to learn the spurious features under suitable conditions.

D. Proofs for Theorems and Propositions

D.1. Proof for Theorem 4.1

Theorem D.1. *[Restatement of Theorem 4.1] Given the same data generation process as in Fig. 1 with $H(S) = H(C) = \delta_f$, assuming the node representations encode proper information of the underlying latent factors (i.e., $\delta_r \geq I(\widehat{\boldsymbol{h}}_i; C) - I(\widehat{\boldsymbol{h}}_i; S) \geq \delta_l, \forall i \in G_c$ and $\delta_r \geq I(\widehat{\boldsymbol{h}}_i; S) - I(\widehat{\boldsymbol{h}}_i; C) \geq \delta_l, \forall i \in G_s$), the graph representation $\widehat{\boldsymbol{h}}_G \in \mathbb{R}^k$ have sufficient capacity to encode k independent features $\{\widehat{\boldsymbol{h}}_G[j]\}_{j=1}^k$ with $H(\widehat{\boldsymbol{h}}_G[j]) \leq \delta_f$, then, if $|G_s|/|G_c| > \delta_r/\delta_l$, then the graph representation elicited by the infomax principle (Eqn. 2) exclusively contain spurious features S, i.e.,*

$$\mathbf{h}_s = \arg\max_{\theta} \frac{1}{|G||\mathcal{G}|} \sum_{G \in \mathcal{G}} \sum_{i \in |G|} I\left(\widehat{\mathbf{h}}_i; \widehat{\mathbf{h}}_G\right).$$

Proof. Our proof is established by contradiction. To begin with, without loss of generality, let us assume that in the final solution $\widehat{\boldsymbol{h}}_G$, there exist some features $i_c \in I_C \subseteq \{1, ..., k\}$ in $\widehat{\boldsymbol{h}}_G$ exclusively encoding information of C, such that,

$$I(\widehat{\boldsymbol{h}}_{i_c}; C) = H(C),$$

where we denote i_c-th feature of $\widehat{\boldsymbol{h}}_G$ as $\widehat{\boldsymbol{h}}_{i_c}$ for the clarity of notation. Correspondingly, we could also find the complementary dimensions as $i_s \in I_S$ such that

$$I(\widehat{\boldsymbol{h}}_{i_s}; S) = H(S).$$

Consider a single graph G, to solve $\max_\theta \; I\left(\widehat{\mathbf{h}}_i, \widehat{\mathbf{h}}_G\right)$ (Eqn. 2), we can expand $\widehat{\boldsymbol{h}}_G$ and write out the mutual information with respect to each node as follows:

$$\begin{aligned}
I_{\text{DI}} &= I(\widehat{\boldsymbol{h}}_i; \cup_{i_c \in I_C} \widehat{\boldsymbol{h}}_{i_c}, \cup_{i_s \in I_S} \widehat{\boldsymbol{h}}_{i_s}) \\
&= \sum_{i \in G} \Big(\sum_{i_c \in I_C} I(\widehat{\boldsymbol{h}}_i; \widehat{\boldsymbol{h}}_{i_c}) + \sum_{i_s \in I_S} I(\widehat{\boldsymbol{h}}_i; \widehat{\boldsymbol{h}}_{i_s}) \Big) \\
&= \sum_{i \in G_c} \Big(\sum_{i_c \in I_C} I(\widehat{\boldsymbol{h}}_i; \widehat{\boldsymbol{h}}_{i_c}) + \sum_{i_s \in I_S} I(\widehat{\boldsymbol{h}}_i; \widehat{\boldsymbol{h}}_{i_s}) \Big) + \sum_{i \in G_s} \Big(\sum_{i_c \in I_C} I(\widehat{\boldsymbol{h}}_i; \widehat{\boldsymbol{h}}_{i_c}) + \sum_{i_s \in I_S} I(\widehat{\boldsymbol{h}}_i; \widehat{\boldsymbol{h}}_{i_s}) \Big) \\
&= \sum_{i \in G_c} \Big(\sum_{i_c \in I_C} I(\widehat{\boldsymbol{h}}_i; C) + \sum_{i_s \in I_S} I(\widehat{\boldsymbol{h}}_i; S) \Big) + \sum_{i \in G_s} \Big(\sum_{i_c \in I_C} I(\widehat{\boldsymbol{h}}_i; C) + \sum_{i_s \in I_S} I(\widehat{\boldsymbol{h}}_i; S) \Big)
\end{aligned} \tag{15}$$

Then, considering switching an index $i'_c \in C$ to encode S, we will have the information changes to I_{DI} as the following

$$\begin{aligned}
\Delta I_{\text{DI}} &= \sum_{i \in G_c} \left(-I(\widehat{\boldsymbol{h}}_i; C) + I(\widehat{\boldsymbol{h}}_i; S) \right) + \sum_{i \in G_s} \left(-I(\widehat{\boldsymbol{h}}_i; C) + I(\widehat{\boldsymbol{h}}_i; S) \right) \\
&= -\sum_{i \in G_c} \left(I(\widehat{\boldsymbol{h}}_i; C) - I(\widehat{\boldsymbol{h}}_i; S) \right) + \sum_{i \in G_s} \left(I(\widehat{\boldsymbol{h}}_i; S) - I(\widehat{\boldsymbol{h}}_i; C) \right),
\end{aligned} \tag{16}$$

where we can upper bound the first item

$$\sum_{i \in G_c} \left(I(\widehat{\boldsymbol{h}}_i; C) - I(\widehat{\boldsymbol{h}}_i; S) \right) \leq |G_c| \left(I(\widehat{\boldsymbol{h}}_i; C) - I(\widehat{\boldsymbol{h}}_i; S) \right) = |G_c| \delta_r,$$

and lower bound the second item

$$\sum_{i \in G_s} \left(I(\widehat{\boldsymbol{h}}_i; S) - I(\widehat{\boldsymbol{h}}_i; C) \right) \geq |G_s| \left(I(\widehat{\boldsymbol{h}}_i; S) - I(\widehat{\boldsymbol{h}}_i; C) \right) = |G_s| \delta_l.$$

Then, it suffices to know that

$$\Delta I_{\text{DI}} \geq |G_s| \delta_l - |G_c| \delta_r > 0,$$

and hence switching any node $i_c \in I_C$ to one for S increases the infomax objective. Therefore, we conclude that $\boldsymbol{h}_s$ maximizes the objective of the infomax principle. We conclude the proof for Theorem 4.1. □

D.2. Proof for Theorem 5.1

To prove Theorem 5.1, we show both necessity and sufficiency that Eqn. 6, serving as the penalty term for $\mathcal{L}_{GT}$, will encourage the decorrelation of $\widehat{\mathbf{h}}_c$ and $\mathbf{h}_s$, and elicit invariant representations $\mathbf{h}_c$. First we propose a proposition to aid the proof.

Proposition D.2. *Let $s_i \in \mathbb{R}_+$ and $s_j \in \mathbb{R}_+$ denote the empirical random variable for the logits values of sample that belongs to B_i and B_j respectively. For any two bins B_i and B_j with equal size samples (denoted by K), and located symmetrically around 0.5, i.e., $|s_i| = |s_j|$, $d\left(\mathbb{E}\left[s_{iy}\right], 0.5\right) = d\left(\mathbb{E}\left[s_{jy}\right], 0.5\right) = \epsilon$, and where $sign(\mathbb{E}[s_i] - 0.5) = sign(0.5 - \mathbb{E}[s_j])$. The optimal and unique solution for $\hat{s}$ of the following cross-entropy function $l(\cdot)$ is 0.5:*

$$\begin{aligned} l(\widehat{s}) &= K\left(-\mathbb{E}\left[s_i\right]\log\widehat{s} - \mathbb{E}\left[1-s_i\right]\log(1-\widehat{s})\right) + K\left(-\mathbb{E}\left[s_j\right]\log\widehat{s} - \mathbb{E}\left[1-s_j\right]\log(1-\widehat{s})\right) \\ &= K(-(0.5+\varepsilon)\log\widehat{s} - (1-0.5-\varepsilon)\log(1-\widehat{s})) + K(-(0.5-\varepsilon)\log\widehat{s} - (1-0.5+\varepsilon)\log(1-\widehat{s})) \end{aligned} \tag{17}$$

Proof. Simplifying $l(\widehat{s})$, we get:

$$l(\widehat{s}) = -K\log(\widehat{s}) - K\log(1-\widehat{s}).$$

Calculating $\frac{d}{d\hat{s}}l(\hat{s})$, we get:

$$\begin{aligned} \frac{d}{d\widehat{s}}f(\widehat{s}) &= \frac{d}{d\widehat{s}}(-K\log(\widehat{s}) - K\log(1-\widehat{s})) \\ &= -K\cdot\left(\frac{1}{\hat{s}} - \frac{1}{1-\hat{s}}\right) \\ &= \frac{K(1-2\widehat{s})}{\widehat{s}(1-\widehat{s})}. \end{aligned} \tag{18}$$

Setting $\frac{K(1-2\widehat{s})}{\widehat{s}(1-\widehat{s})} = 0$, we get $\hat{s} = 0.5$. We conclude the proof.

□

With Prop. D.2, we first prove the necessity, i.e., given $\widehat{\mathbf{h}}_c = \mathbf{h}_c$, Eqn. 6 satisfies $\max\ H\left(\mathbf{h}_s \mid \widehat{\mathbf{h}}_c\right)$.

Proof. Given $\widehat{\mathbf{h}}_c = \mathbf{h}_c$, and under the same class label $Y = y$, all samples that belong to class y are encoded into $\mathbf{h}_c$, as C is causally related to Y. Under the assumption of Theorem 5.1, we have K pairs of symmetric bins for any label Y. Using Prop. D.2, we know that for any pair of symmetrical bins B_i and B_j, the optimal solution for the estimated logits $\widehat{s_y} = \rho(\mathbf{h}_c) \in \mathbb{R}^+$ is 0.5, which maximizes the prediction entropy. As the K pairs of bins simultaneously achieve the same optimal solution, and $l(\cdot)$ is a convex function, $\widehat{s_y} = 0.5$ is both optimal and unique. We conclude the proof. □

The necessity condition shows that $\widehat{\mathbf{h}}_c = \mathbf{h}_c$ is indeed one of the feasible solutions for Eqn. 6, which decorrelates $\mathbf{h}_s$ and $\widehat{\mathbf{h}}_c$. We remain to show that $\widehat{\mathbf{h}}_c = \mathbf{h}_c$ is the unique solution for Eqn. 6, when we optimize $\mathcal{L} = \mathcal{L}_{GT} + \lambda\mathcal{L}_{Inv}$, where Eqn. 6 serves as a penalty term $\mathcal{L}_{Inv}$. To demonstrate the sufficiency, we will divide our analysis into three distinct scenarios. In each scenario, we will compare and discuss the empirical risks associated with $\mathcal{L}_{GT}$ and $\mathcal{L}_{Inv}$.

Proof. The empirical risk for $l(\cdot)$ is defined as: $\hat{R}_n(l) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\left(l\left(X_i\right) \neq Y_i\right)$. We consider three scenarios: 1) The solution only contains C, i.e., $\widehat{\mathbf{h}}_c = \mathbf{h}_c$, which we have proved that it is one of the feasible solution. 2) The solution only contains S, i.e., $\widehat{\mathbf{h}}_c = \mathbf{h}_s$. 3) The solution contains a mixing ratio of C and S. We need to show that for case 1), the empirical risk for $\mathcal{L}_{GT} + \lambda\mathcal{L}_{Inv}$ is the smallest. We consider the first two cases. For any pair of symmetrical bins with equal sample size n ($2n$ in total):

- For case 1, $\hat{R}_n(\mathcal{L}_{GT})$ would be 0 as $\widehat{\mathbf{h}}_c = \mathbf{h}_c$. $\hat{R}_n(\mathcal{L}_{Inv}) = n$ as in this case the estimated probability $p = 0.5$, according to Prop. D.2.

- For case 2, $\hat{R}_n(\mathcal{L}_{GT}) = n$ and $\hat{R}_n(\mathcal{L}_{Inv}) = 0$ since we can encode different $\widehat{\mathbf{h}}_c$ for different samples to fit the spurious patterns. However, only half of them are correlated to Y.

From the above discussion, we conclude that when $\lambda < 1.0$, the solution for $\mathcal{L}$ would prefer $\widehat{\mathbf{h}}_c = \mathbf{h}_c$, whereas when $\lambda > 1.0$, the solution would prefer $\widehat{\mathbf{h}}_c = \mathbf{h}_s$. This is consistent with our experiment results in Section I.4: when $\lambda > 1.0$, the model performance degrade dramatically.

Finally, we also need to rule out case 3. For simplicity, we assume that the encoder $h(\cdot) \sim \text{Bernoulli}(r)$ is stochastic which follows a Bernoulli distribution with parameter r. Then in each bin, there are r samples that are encoded using C and $1-r$ encoded using S, then with a similar technique, we can derive that: $\hat{R}_n(\mathcal{L}_{GT}) = (1-r)n$, and $\hat{R}_n(\mathcal{L}_{Inv}) = rn$, which can be obtained as a convex combination of case 1 and case 2 for $\mathcal{L}_{GT}$ and $\mathcal{L}_{Inv}$ respectively. As for case 3, $\hat{R}_n(\mathcal{L}_{GT}) = (1-r)n > 0$ when $r < 1$, for any value r, we can find a suitable λ that make the empirical risk for case 1 to be smallest among the 3 scenarios. In other words, $\widehat{\mathbf{h}}_c = \mathbf{h}_c$ would be optimal and unique solution when $\mathcal{L} = \mathcal{L}_{GT} + \lambda\mathcal{L}_{Inv}$. We conclude the proof for Theorem 5.1.

□

E. More Discussions on EQuAD Framework

Encoding-**QuA**ntifying-**D**ecorrelation (EQuAD) is a flexible learning framework, where off-the-shelf algorithms can serve as plug-ins for specific implementations for each step. In the first two steps, *Encoding* and *Quantifying*, we adhere to infomax-based self-supervised learning and model-based quantification due to their effectiveness. For the third step, *Decorrelation*, multiple options are available. We note that the learning objective in Eqn. 7 demonstrates superior performance across multiple datasets. However, in cases of significant bias, this objective may encounter optimization challenges, despite its theoretical soundness. This issue often arises when S and Y are highly correlated, leading minibatch gradient descent to inadvertently reinforce gradients towards spurious patterns, as samples highly correlated with Y predominate in each minibatch. A potential solution involves full-batch gradient descent with sample reweighting, but this approach is memory-intensive and impractical for large datasets. To overcome this limitation, we propose an alternative learning objective for decorrelation, drawing inspiration from DisC (Fan et al., 2022). This approach utilizes ERM and assigns greater weight to sample i from class $Y = y$ with logits value $\widehat{s}_{iy}$ close to zero, indicating a lack of spurious patterns. By employing a reweighted cross-entropy loss, this method focuses on learning invariant features. The formulation of the alternative loss function $\mathcal{L}_{GT}$ is outlined as follows:

$$\mathcal{L}_{GT} = w_i CE\big(y_i, \widehat{y_i}\big), \forall i \in [N], \text{ s.t. } w_i = \frac{1 - s_{iy}^{\gamma}}{\gamma}, \tag{19}$$

where $CE(\cdot,\cdot)$ is the cross-entropy loss. In Eqn. 19, each sample i with ground-truth label y get reweighted such that the model training can focus on samples exhibit weaker spurious correlation, thus implicitly achieve the optimality of Eqn. 5. While DisC (Fan et al., 2022) requires learning the biased graphs at first using generalized cross entropy (GCE) (Zhang & Sabuncu, 2018) loss, EQuAD can directly obtain the spurious (biased) information from step 1 and 2, without presumptive assumptions on $\mathbb{P}(S, Y)$. We adopt Eqn 19 for two-piece graph datasets without the need of $\mathcal{L}_{Inv}$, and observe better predictive performance and faster convergence speed.

F. More Details on Experiments about Representation Quality Analysis

Recent studies have indicated that for Euclidean data, even when neural networks heavily rely on spurious features and perform poorly on minority groups where the spurious correlation is broken, they still learn the invariant, or core features sufficiently well (Kirichenko et al., 2023). In our work, we conducted a similar experiment with the SPMotif datasets to investigate whether ERM has already learned effective invariant features. Specifically, we first trained an encoder $h(\cdot)$ on the training set using ERM, then froze this encoder and obtain representations for each sample. Subsequently, we added a 2-layer MLP on top of $h(\cdot)$ and re-trained this MLP for feature reweighting on half of the validation set, where the spurious correlation does not hold. If ERM truly learned causally-related features, then the feature reweighting via the validation set should be able to exploit the invariant features to achieve OOD generalization. The experiment results are illustrated in left part of Figure 2. As we can see, for both SPMotif-0.60 and SPMotif-0.90, the test accuracy is already on par with or

even better than many state-of-the-art invariance learning methods. This suggests that *ERM has already been able to learn effective invariant features*, which is contrary to our objective. In contrast, when utilizing the representations learned from infomax-based SSL, the accuracy after feature reweighting exhibits a significant gap compared with ERM, indicating that the representations derived from infomax-based SSL predominantly contains spurious features.

G. Algorithmic Pseudocode

We provide the following pseudocode to facilitate a better understanding of our algorithm and framework. The code will be open-sourced upon acceptance of our work.

Algorithm 1 The `EQuAD` Framework

0: **Input:** graph dataset $\mathcal{G}$, set of epochs $\mathcal{P}$, number of classifiers M, total number of training epochs P for decorrelation;
0: **Output:** Model $f = \rho \circ h$, composed of encoder $h(\cdot)$ and classifier $\rho(\cdot)$;
0: **Step 1: Encoding**
0: Initialize encoder $h(\cdot)$;
0: **for** each model architecture $i \in [K]$ **do**
0: **for** each epoch $j \in \mathcal{P}$ **do**
0: Train $h(\cdot)$ using Eqn. 2;
0: Generate latent representations $\mathbf{H}^{(i,j)}$;
0: **end for**
0: **end for**
0: Collect all representations $\mathcal{H} := \{\mathbf{H}^{(i,j)}\}$;
0: **Step 2: Quantifying**
0: Initialize classifiers $g(\cdot)$;
0: **for** each representation $\mathbf{H}^{(i,j)} \in \mathcal{H}$ **do**
0: Train classifier $g(\cdot)$ using ERM, taking $\mathbf{H}^{(i,j)}$ and Y as inputs;
0: Compute logits matrix $\mathbf{S}^{(i,j)} = g(\mathbf{H}^{(i,j)}, Y)$;
0: **end for**
0: Collect all logits matrices $\mathcal{S} := \{\mathbf{S}^{(i,j)}\}$;
0: Select top-M classifiers $g(\cdot)$ according to the lowest validation metric;
0: **Step 3: Decorrelation**
0: Reinitialize model $f(\cdot)$;
0: **for** epoch $e \in \{1, 2, \ldots, P\}$ **do**
0: **for** each minibatch B in $\mathcal{G}$ **do**
0: Compute logits from top-M logits matrices in $\mathcal{S}$;
0: Minimize Eqn. 8 to update $\rho(\cdot)$ and $h(\cdot)$;
0: **end for**
0: **end for**
=0

H. Complexity Analysis

Time Complexity. The time complexity of the `EQuAD` framework depends on the specific GNN encoder employed. In this work, we utilize GIN (Xu et al., 2019), which is a 1-WL GNN. Consequently, the time complexity is $\mathcal{O}(CkmF)$, where k is the number of GNN layers, F is the feature dimensions and m denotes the number of edges in a graph G. `EQuAD` incurs an additional constant $C > 1$ due to the multiple-stage learning paradigm.

Space Complexity. EQuAD also incurs additional memory overhead as in the encoding step, it needs to store multiple matrices, and in the quantifying stage the matrices are transformed into logits matrices (which is much lower dimensional). In the implementation, we save the graph representation matrices (and corresponding labels) obtained from encoding stage in the disk, and load them one by one, followed by transforming them into logits matrices, hence the memory complexity is $\mathcal{O}(H|\mathcal{C}||\mathcal{B}| + |\mathcal{B}|m)$, here $|\mathcal{C}|$ is the number of class labels, H is the number of the graph representation matrices selected from the quantifying step, and $|\mathcal{B}|$ denotes the batch size. As H, $|\mathcal{C}|$ and $|\mathcal{B}|$ are usually small integers, the memory cost is affordable.

Table 5. Running time analysis in seconds for various baseline methods and EQuAD on EC50 datasets.

Method	EC50-Sca	EC50-Assay	EC50-Size
ERM	113.97±1.56	169.83±21.76	224.54±63.38
IRM	1102.51±17.31	1719.48±215.50	1035.22±30.71
V-Rex	932.81±1.16	1498±137.18	886.07±0.27
CIGA	2179.94±540.88	2676.04±897.95	1822.99±551.09
GREA	1812.90±273.94	2461.01±876.19	1661.67±375.18
GALA	1699.81±235.46	2176.58±638.76	1365.27±240.32
GSAT	1140.90±285.10	1609.15±322.02	1135.12±326.98
EQuAD	878.34±12.31	1240.88±36.32	1326.61±21.79

We provide a empirical running time analysis in Table 5. As illustrated, the running time of EQuAD is comparable to or less than that of most invariance learning methods, also exhibiting lower variance, since we do not adopt early stop and only need to run for 50 epochs for all datasets. Specifically, the first step accounts for around 65% of the total time, while steps 2 and 3 together constitute the remaining 35%.

I. Experimental Details

I.1. Datasets

We provide a more detailed introduction of the datasets adopted in the experiments as follows.

SPMotif datasets. Following (Wu et al., 2022b), we generate a 3-class synthetic datasets based on BAMotif (Ying et al., 2019). In these datasets, each graph comprises a combination of invariant and spurious subgraphs, denoted by I and S. The spurious subgraphs include three structures (Tree, Ladder, and Wheel), while the invariant subgraphs consist of Cycle, House, and Crane. The task for a model is to determine which one of the three motifs (Cycle, House, and Crane) is present in a graph. A controllable distribution shift can be achieved via a pre-defined parameter b. This parameter manipulates the spurious correlation between the spurious subgraph G_s and the ground-truth label Y, which depends solely on the invariant subgraph G_c. Specifically, given the predefined bias b, the probability of a specific motif (e.g., House) and a specific base graph (Tree) will co-occur is b while for the others is $(1-b)/2$ (e.g., House-Ladder, House-Wheel). When $b = \frac{1}{3}$, the invariant subgraph is equally correlated to the three spurious subgraphs in the dataset. In SPMotif datasets, S is directly influenced by C, and C is causally related with Y, thus satisfies our data generating assumption as FIIF.

Two-piece graph datasets. To validate the effectiveness of EQuAD under the PIIF data generating process, we adopt the two-piece graph datasets(Chen et al., 2023). These datasets employ parameters α^e and β^e to control the correlations between C and Y, and between S and Y respectively in different environments. A formal definition of the two-piece graph is presented as follows:

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

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

(b) Generate G_c and G_s via: $G_c := f_{\text{gen}}^{G_c}\left(Y^e \cdot \text{Rad}\left(\alpha_e\right)\right), G_s := f_{\text{gen}}^{G_s}\left(Y^e \cdot \text{Rad}\left(\beta_e\right)\right)$, where $f_{\text{gen}}^{G_c}, f_{\text{gen}}^{G_s}$ map the input $\{-1, 1\}$ to a corresponding graph selected from a given set, and $\text{Rad}(\alpha)$ is a random variable taking value -1 with probability α and +1 with $1-\alpha$;

(c) Synthesize G^e by randomly assembling G_c and G_s : $G^e := f_{gen}^{G}\left(G_c, G_s\right)$.

Specifically, we adopt BAMotif (Ying et al., 2019) to generate 3 variants of 3-class two-piece graph datasets, with different correlation degrees of $H(S|Y)$ parametrized by (α^e, β^e), where we can examine both scenarios where $H(C|Y) > H(S|Y)$ and $H(C|Y) < H(S|Y)$, to verify the effectiveness of the baseline methods and our approach.

DrugOOD. The DrugOOD benchmark(Ji et al., 2022) is specifically designed for OOD challenges in AI-aided drug discovery. It comprises a diverse collection of datasets focusing on the prediction of drug-target interactions. Each dataset

within DrugOOD is derived from real-world bioactivity data, encompassing a range of drug compounds and their target proteins. The datasets are categorized based on two bioactivity measures: IC50 and EC50, representing half-maximal inhibitory concentration and half-maximal effective concentration, respectively. DrugOOD datasets are split into subsets using three distinct criteria: Assay, Scaffold, and Size. The Assay-based split groups data according to biological assays, reflecting variations in experimental conditions. The Scaffold-based split focuses on the chemical structure, categorizing compounds by their core molecular scaffolding. The Size-based split, on the other hand, divides the data based on the size of the molecular compounds, offering insights into size-dependent drug properties. With the two measurements (IC50 and EC50), and three environment-splitting strategies (assay, scaffold and size), we obtain 6 datasets, and each dataset contains a binary classification task for drug target binding affinity prediction.

Open Graph Benchmark(OGB). We also consider two molecule datasets, MOLBACE and MOLBBBP, from Open Graph Benchmark (Hu et al., 2021). We use the scaffold splitting procedure as OGB adopted, where different molecules are structurally separated into different subsets, which provides a more realistic estimate of model performance in experiments (Wu et al., 2018a).

I.2. Experiment Settings

Encoding. For encoding stage, we adopt MVGRL(Hassani & Khasahmadi, 2020) to obtain the latent representations. MVGRL (Hassani & Khasahmadi, 2020) utilize global-local MI maximization as learning objective, In addition, it utilizes Personalized PageRank(Page et al., 1999) and heat kernel(Kondor, 2002) as data augmentation methods to generate correlated structural views for contrastive learning. Intuitively, the structural augmentation corrupts the invariant substructure by randomly adding edges to the neighboring nodes, thus further facilitates the learning of spurious patterns under OOD settings. Across all the datasets, we set teleport probability $\alpha = 0.2$, and diffusion time $t = 5$. To obtain a collection of latent representation $\mathcal{H}$, we set pre-defined training epochs $\mathcal{P} = \{50, 100, 150\}$, number of layers in GNN encoder $L = \{2, 3, 5\}$, hidden dimensions $H = \{32, 64\}$, leading to 18 embedding matrices for each dataset.

Quantifying. To quantify the latent representations in $\mathcal{H}$ and obtain $\mathcal{S}$, we adopt linear svm(Hearst et al., 1998) for its computational efficiency, followed by probability calibration (Platt et al., 1999; Zadrozny & Elkan, 2001) to get logits values. For all datasets, the regularization parameters C of linear svms are searched over $\{10, 1000\}$. We set cross validation to be 5 for probability calibration classifier.

Decorrelation. For the decorrelation step, we utilize Eqn. 8 and search over the following hyperparameters:

- $\gamma : \{0.7, 0.5, 0.3, 0.1, 0.05\}$
- $\tau : \{1.0, 0.5, 0.25, 0.1, 0.01\}$
- $\lambda : \{1e-3, 1e-2, 1e-1\}$

As discussed in Section E, when the dataset exhibits high bias, employing Eqn. 7 can lead to gradient conflicts between $\mathcal{L}_{GT}$ and $\mathcal{L}_{Inv}$, posing challenges in optimization. A potential solution is to adopt full-batch training; however, this approach is not memory efficient and is impractical for large-scale datasets. To address this issue, we propose an alternative loss that encourages decorrelation between $\widehat{\mathbf{h}}_c$ and $\mathbf{h}_s$ which is efficient and effective when the bias is high, as outlined in Eqn. 19. For two-piece graph datasets, we adopt Eqn. 19 for model training, which simplifies the optimization process, and also enhances performance. The hyperparameter q is searched over: $\{0.9, 0.7, 0.5, 0.3, 0.1, 0.01\}$.

Backbone encoder. We adpot GIN(Xu et al., 2019) as backbone encoder for all the datasets, and the number of layers L are searched over $\{3, 5\}$ across all the datasets for EQuAD and all other baseline methods. For synthetic datasets, the hidden dimension is uniformly set to 32 across all methods, while for the DrugOOD and OGBG datasets, it is set to 128.

Other experiment setting details. For all the methods and datasets, the experiments are ran for three times, with random seeds $\{1, 2, 3\}$. The batch size is 32 for all experiments, with a learning rate of $1e-3$ is applied across all experiments. We adopt Adam optimizer (Kingma & Ba, 2017) for model training. For `EQuAD`, the experiments for SPMotif and DrugOOD datasets are ran for 50 epochs, and for OGBG datasets, the experiments are ran for 100 epochs. We do not use early stopping and learning rate scheduler for all the experiments.

Table 6. Ablation study on different SSL algorithms

Method	SPMotif-0.40	SPMotif-0.60	SPMotif-0.90	EC50-Assay	EC50-Sca	EC50-Size
ERM	62.19±3.26	55.24±2.43	49.41±3.78	67.39±2.90	64.98±1.29	65.10±0.38
EQuAD w/GraphCL	72.78±0.81	70.66±2.67	67.79±1.29	75.66±1.27	66.24±0.61	64.13±0.65
EQuAD w/Infomax	**73.13±1.56**	**71.93±1.94**	**69.47±2.06**	**79.36±0.73**	**68.12±0.48**	**66.37±0.64**

Table 7. Ablation study on different decorrelation loss objectives

Method	(0.80,0.70)	(0.80,0.90)	(0.70,0.90)	EC50-Assay	EC50-Sca	EC50-Size
EQuAD w/Eqn. 19	82.76±0.71	**75.81±0.51**	**71.95±1.14**	79.14±0.51	67.36±0.43	64.48±1.14
EQuAD w/Eqn. 8	**82.03±0.38**	54.36±3.03	44.69±1.06	**79.36±0.73**	**68.12±0.48**	**66.37±0.64**

I.3. Implementation Details

We implement our method and all baseline methods using PyTorch(Paszke et al., 2019) and PyTorch Geometric(Fey & Lenssen, 2019). To generate latent embeddings with MVGRL(Hassani & Khasahmadi, 2020), we adopt PyGCL(Zhu et al., 2021) package for the implementation. We utilize linearSVC and CalibratedClassifierCV in scikit-learn(Pedregosa et al., 2011) for the implementation of linear SVM and probability calibration classifier respectively. We ran all our experiments on Linux Servers with GeForce RTX 4090 with CUDA 11.8.

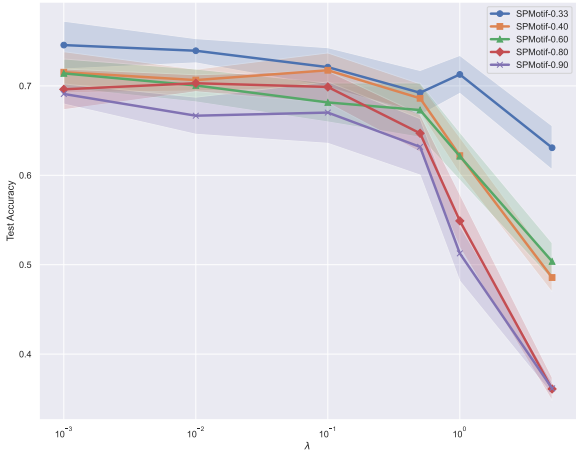


Figure 7. Ablation study on λ. Notably, when λ is greater than 1, the performance degrades dramatically, which is consistent with our theoretical results in Seciton D.2.

I.4. More Experiment Results

More Ablation Studies We provide more ablation studies to study the impact of different SSL algorithms in the *Encoding* stage, and the impact of different decorrelation loss objectives. In our study, we adopt the infomax principle to learn spurious features, here we provide experiment results with GraphCL (You et al., 2020) as an alternative for learning spurious features in the *Encoding* stage, as illustrated in Table 6. As shown in Table 6, the representations derived from GraphCL, compared to those obtained using Infomax, exhibit a notable decline in performance on both synthetic and real-world datasets. This phenomenon could likely be attributed to the scenario when $|G_c| < |G_s|$, randomly dropping nodes or edges are more likely to perturb the spurious subgraph G_s, thereby enhancing the invariant feature learning for G_c. This underscores the importance of accurately identifying spurious features in the *encoding* step. We also conduct experiments to study the decorrelation loss objectives (i.e., Eqn. 8 and Eqn. 19). The experiment results are illustrated in Table 7. As discussed in Appendix E, Eqn. 8 poses optimization challenges due to gradient conflicts for Two-piece graph datasets, while using Eqn. 19 brings significant performance improvement, which also demonstrates the flexibility of EQuAD.

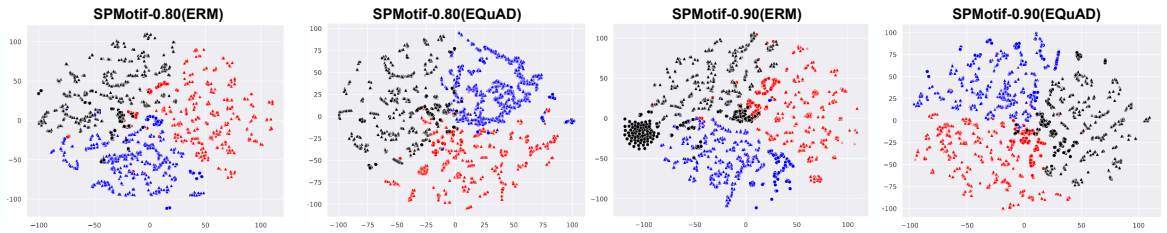


Figure 8. Latent representations of data samples in 2d space for SPMotif datasets, where samples from different labels are represented by different colors, within the same class label, we use *cycles* and *triangles* to represent data samples with high and low spurious correlations to Y. Representations obtained from EQuAD exhibit higher overlapping for the *cycles* and *triangles* points within the same classes, thus leading to better graph invariant features.

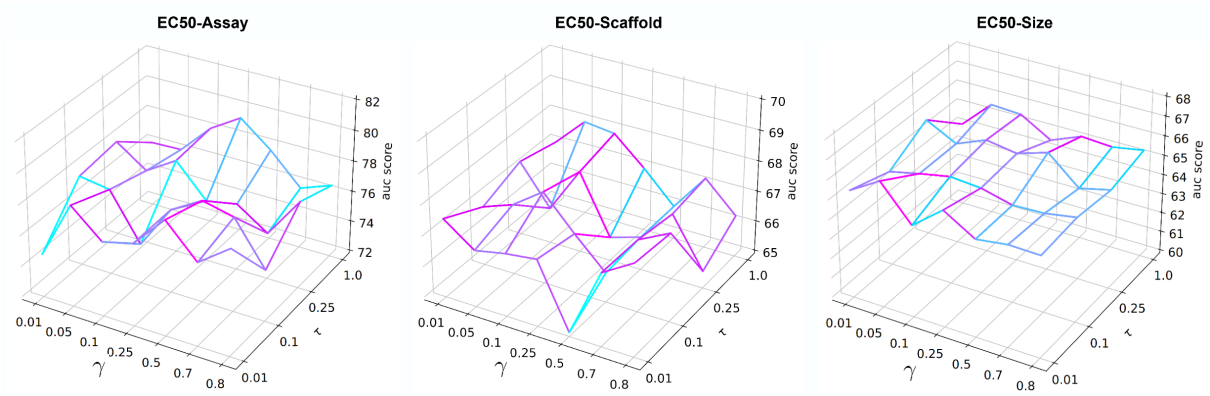


Figure 9. Sensivitity on hyperparameters of γ and τ.

Sensitivity Analysis We first investigate the impact of various hyperparameters. First, we examine λ in Eqn. 8, which controls the regularization strength of $\mathcal{L}_{Inv}$. As demonstrated in Figure 7, across 5 different datasets, the optimal λ generally remains below 0.1. When $\lambda \geq 1$, there is a substantial decline in model performance, as now the empirical risk focus on $\mathcal{L}_{Inv}$ which maximizes the conditional entropy $H\left(\mathbf{h}_s \mid \widehat{\mathbf{h}}_c\right)$, hence degrade to random guessing ultimately. The experiment results are also consistent with our theoretical results in Sec. D.2, i.e., λ should be less than 1.0 to guide the model focusing on learning invariant features.

Next, we examine γ and τ, the hyperparameters that control the sample-specific reweighting and model-specific reweighting scheme respectively. The experiments are performed on three EC50 datasets. As illustrated in Figure 9, under a range of hyperparameter settings, EQuAD consistently matches or outperform the performance of state-of-the-art methods in all three EC50 datasets. Moreover, the careful tuning of γ and τ are shown to be effective empirically. For instance, on the EC50-Scaffold dataset, the best-performing hyperparameters yield a performance increase of 3.67% compared to the least effective settings.

More Visualizations on Latent Representations in 2d Space We provide more visualization results of latent representations in 2d space using t-SNE (van der Maaten & Hinton, 2008). Figure 8 illustrates the latent representation in 2d space for the training set, where samples from different labels (represented by various colored points) trained with both ERM and EQuAD exhibit a clear decision boundary, achieving training accuracy above 90% in both cases. On the other hand, within the same classes, we use *cycles* and *triangles* to represent data samples with high and low spurious correlations to Y respectively. The visualizations indicate that the representations obtained from ERM exhibit a notable separation between the two groups within the latent space, implying minimal overlap. In contrast, the representations from EQuAD show greater overlap, or they are more closely positioned. This observation implies that the GNN model $f(\cdot)$ trained from scratch learns high-quality invariant features.

More Visualizations on Prediction Logits from $g(\cdot)$ We present the logits distributions for all three classes in SPMotif-0.40 and SPMotif-0.80, as derived from the top-4 classifiers $g(\cdot)$ from Step 2. Figure 10 illustrates that, under varying

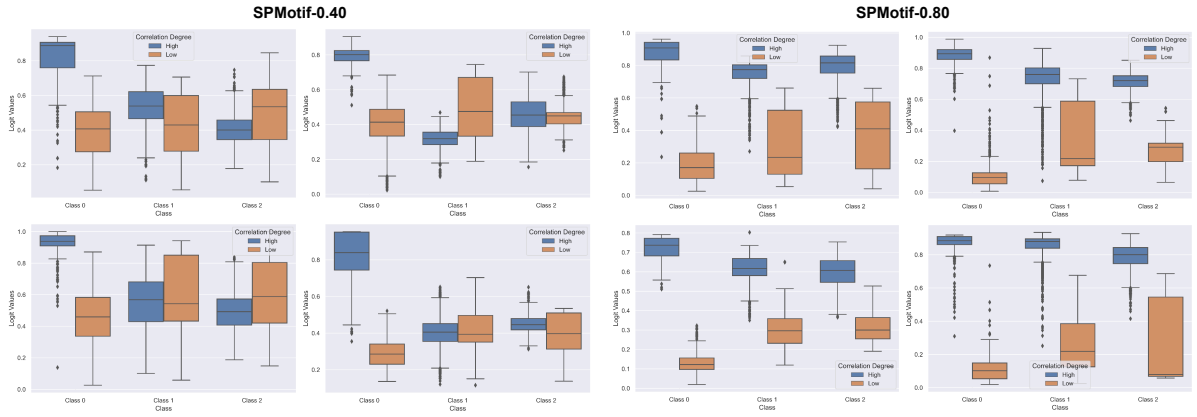


Figure 10. Logits distributions for all three classes of SPMotif-0.40 and SPMotif-0.80.

degrees of spurious correlation, both high and low correlation samples exhibit discernible differences. Notably, at a higher bias level (e.g., $bias = 0.8$), the distinction between these samples is more pronounced, with a certain symmetry around 0.5, providing favorable conditions for subsequent decorrelation. However, at $bias = 0.4$ some classes show less distinct separation and lack symmetry. This observation underscores the need to implement sample-specific and model-specific reweighting strategies to determine the optimal prediction logits matrices.

Does infomax-based SSL representations capture spurious substructures in real-world datasets? Our analysis, as illustrated in Figure 2, demonstrates that infomax-based SSL embeddings predominantly capture spurious features in SPMotif datasets. This is a crucial prerequisite for successful decorrelation and invariance learning in subsequent steps. To further investigate the propensity of infomax-based SSL embeddings to learn spurious features in real-world datasets, we conducted experiments on IC50 and EC50 datasets. We evaluated the train/validation AUC scores for the top-5 classifiers $g(\cdot)$ in step 2, specifically selecting those with the lowest validation AUC scores. Ideally, we would expect a high train AUC score with a low validation AUC score.

Observations from Figures 11 and 12 reveal that, across datasets with different distribution shifts, there is typically a significant gap between train and validation AUC scores for most $g(\cdot)$, indicating a primary focus on learning spurious features. However, there are exceptions, such as in the IC50-Scaffold dataset, where a smaller gap due to lower train AUC scores suggests that the representation may not predominantly contain spurious features. This finding leads us to consider implementing a model-specific reweighting scheme to address such scenarios. Finally, the observed discrepancy between the training and validation AUC scores also supports the assumption that $|G_c| < |G_s|$ is valid in these datasets, which is crucial for the successful capture of spurious features by the infomax-based SSL approach.

Infomax principle learns invariant features when $|G_c| > |G_s|$ To further evaluate our theoretical presumption that infomax-based SSL can extract spurious features when $|G_c| < |G_s|$, we perform another empirical study by modifying the SPMotif dataset to let $|G_c| > |G_s|$, and examine whether infomax-based SSL learns causal features. Specifically, we modify SPMotif dataset to shrink the size of base spurious subgraph (from 40 nodes in average to 12) and increase the number of ground-truth subgraphs (from 1 to 10). By doing this, $|G_c| > |G_s|$ is satidfied, and we conduct a similar experiment as discussed in Section F. Figure 13 demonstrates both scenarios of $|G_c| < |G_s|$ (in the middle) and $|G_c| > |G_s|$ (in the right). When $|G_c| > |G_s|$, the validation accuracy (after feature reweighting) exceeds 90%, implying that SSL with infomax principle learns causal invariant features. We can also see that the test accuracy is also much higher compared to the case when $|G_c| < |G_s|$ (62.1% vs 40.3%), although still lower than ERM.

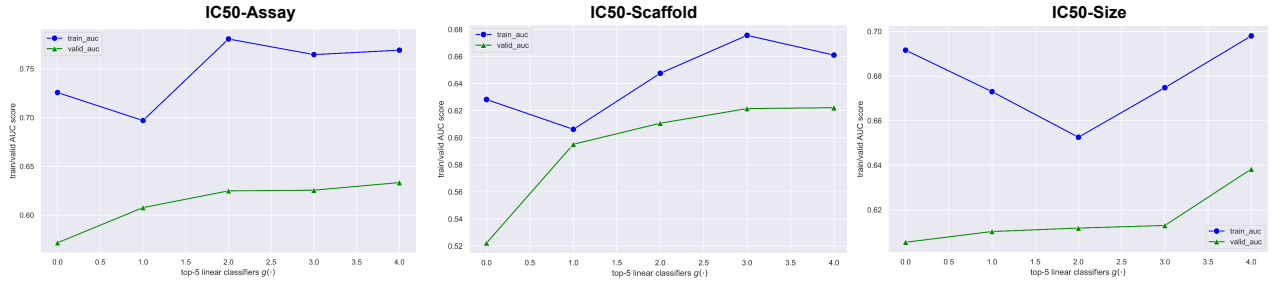


Figure 11. Train and validation AUC score for top-5 linear classifiers $g(\cdot)$ in step 2 for IC50 datasets.

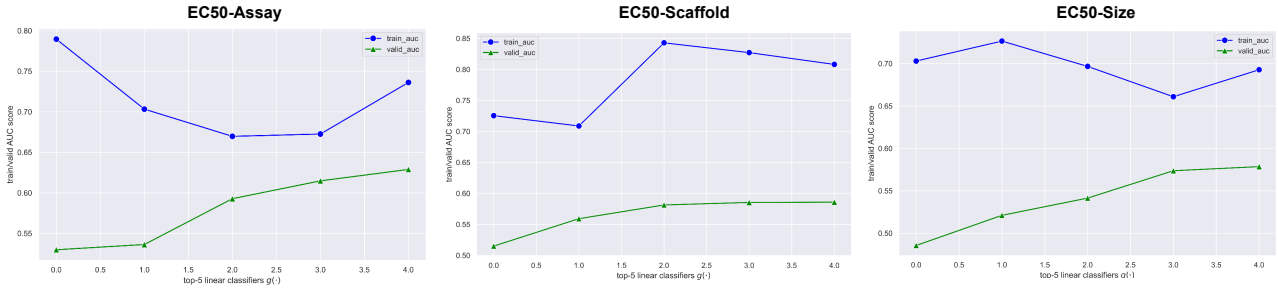


Figure 12. Train and validation AUC score for top-5 linear classifiers $g(\cdot)$ in step 2 for EC50 datasets.

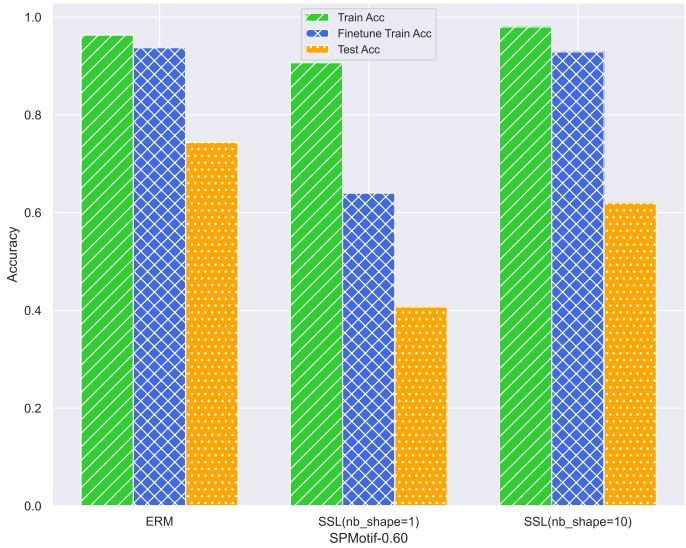


Figure 13. Representation quality in terms of how much invariant features are included. When $|G_c| > |G_s|$, infomax-based SSL effectively include causal invariant features, which validates assumptions of theorem 4.1 empirically from another perspective.