Improving Out-of-Distribution Generalization in Graphs via Hierarchical Semantic Environments

Yinhua Piao[1], Sangseon Lee[2], Yijingxiu Lu[1], Sun Kim[1,3,4]

Department of Computer Science and Engineering, Seoul National University[1]
Institute of Computer Technology, Seoul National University[2], AIGENDRUG Co., Ltd.[3]
Interdisciplinary Program in Artificial Intelligence, Seoul National University[4]
{2018-27910, sangseon486, solanoon0113, sunkim.bioinfo}@snu.ac.kr

Abstract

*Out-of-distribution (OOD) generalization in the graph domain is challenging due to complex distribution shifts and a lack of environmental contexts. Recent methods attempt to enhance graph OOD generalization by generating **flat** environments. However, such flat environments come with inherent limitations to capture more complex data distributions. Considering the DrugOOD dataset, which contains diverse training environments (e.g., scaffold, size, etc.), flat contexts cannot sufficiently address its high heterogeneity. Thus, a new challenge is posed to generate more semantically enriched environments to enhance graph invariant learning for handling distribution shifts. In this paper, we propose a novel approach to generate **hierarchical semantic environments** for each graph. Firstly, given an input graph, we explicitly extract variant subgraphs from the input graph to generate proxy predictions on local environments. Then, stochastic attention mechanisms are employed to re-extract the subgraphs for regenerating global environments in a hierarchical manner. In addition, we introduce a new learning objective that guides our model to learn the diversity of environments within the same hierarchy while maintaining consistency across different hierarchies. This approach enables our model to consider the relationships between environments and facilitates robust graph invariant learning. Extensive experiments on real-world graph data have demonstrated the effectiveness of our framework. Particularly, in the challenging dataset DrugOOD, our method achieves up to 1.29% and 2.83% improvement over the best baselines on IC50 and EC50 prediction tasks, respectively.*

1. Introduction

Graph-structured data is ubiquitous in real-world applications, from social networks to biological networks and chemical molecules [14, 16, 21, 48]. One notable advancement in this area is the emergence of Graph Neural Networks (GNNs). GNN-based models have pioneered end-to-end learning strategies to extract valuable information from graphs and have demonstrated remarkable success across various applications [13, 24, 42]. However, the success of GNNs encounters challenges in out-of-distribution (OOD) scenarios primarily due to the intricate nature of graph distribution shifts [3, 9, 28]. Graph data, characterized by nodes, edges, and potential attributes, poses additional challenges compared to other domains such as natural language processing (NLP) or computer vision (CV). Unlike those domains where context is often provided by sentences, paragraphs, or spatial information in images [4, 15, 36], graph lacks a built-in contextual framework, making it inherently challenging to discern the relevance and context of individual graph elements in OOD scenarios [12, 49].

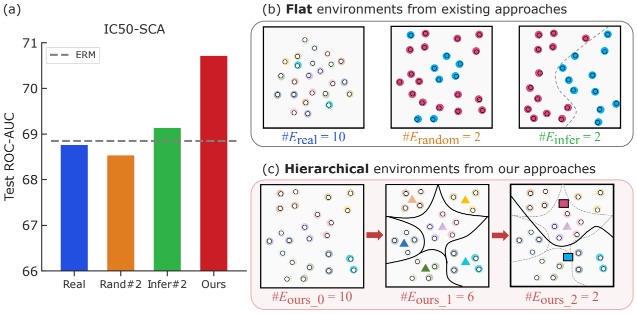

Figure 1. (a) Results on IC50-SCA dataset from DrugOOD [20]. (b) Flat environments from existing approaches. (c) Hierarchical environments from our methods. For visualization, we set #real environments as 10.

Invariant Risk Minimization (IRM) [2] is a widely used strategy in the Euclidean domain, relying on the assumption that training data is sourced from distinct environments with varied data distributions. Motivated by the success of invariant learning in the Euclidean domain [1, 2, 8, 17, 26, 29, 31, 32], invariant learning methods for non-Euclidean graphs [6, 7, 10, 27, 27, 30, 43] extend the concept to address the unique challenges posed by graph-structured data. Despite commendable progress in addressing OOD scenarios within the graph domain, existing studies have predominantly adhered to the assumption of a flat environmental situation [5, 33]. For example, in the DrugOOD dataset, there are nearly 7000 diverse training environments in IC50-SCA subset, including a wide variety of distinct substructures. Fig. 1 illustrates three existing approaches to acquire environments when performing invariant learning on this data: *"Real"* means directly utilizing provided environments [27], *"Rand#2"* means randomly splitting samples into two environments [2, 6], and *"Infer#2"* means inferring samples into two environments [8, 17]. As shown in Fig. 1, all existing approaches exhibit poor or comparable performance to empirical risk minimization (ERM), which is consistent with the findings of Yang [43]. These phenomena indicate the shortcomings of flat environments: (1) Limited consideration of local environment similarity degrades performance in numerous environments (e.g., '*Real*' case in Fig. 1) (2) Inferring from a small number of environments may fail to capture global environment similarities and interrelationships. (e.g., '*Infer#2*' case in Fig. 1)

Considering that graphs exhibit a hierarchical structure, with semantic information organized across various levels. This inherent hierarchy is fundamental to understanding structures and properties within graphs [44]. In addition, inspired by the recent works [1, 17, 29], which show that the diversity of environments is recognized as the key to effectively handling graph OOD scenarios , we aim to bridge the gap of flat environments while leveraging the inherent hierarchy of graphs. In this paper, we propose a hierarchical approach to generate semantic environments of each graph for effective graph invariant learning. Initially, we extract variant subgraphs from the input graph, enabling the generation of proxy predictions on local environments. Employing stochastic attention mechanisms, we iteratively re-extract subgraphs, building global environments hierarchically. To guide the robustness of hierarchical environment inference, we introduce a hierarchical environment diversification loss which encourages our model to diversify environments within the same hierarchy while maintaining consistency across different hierarchical levels. This approach not only aids our model in considering relationships between environments but also strengthens graph invariant learning robustness. By modeling relationships between different environments within a hierarchical framework, our approach acquires the rich source of environmental information embedded in the hierarchical structure of graphs. Extensive experiments demonstrate our approach to graph OOD classification datasets. Our contributions can be summarized as follows:

- We propose a hierarchical approach to generate semantic environments for effective graph invariant learning. To the best of our knowledge, our proposed method is the first attempt to generate the environments in a hierarchical way in graph OOD generalization.
- We introduce a new learning objective that guides our model to learn the diversity of environments within the same hierarchy while maintaining consistency across different hierarchies.
- Extensive experiments have demonstrated our model yields significant improvements over various domains. In molecule graph benchmarks DrugOOD, our method achieves up to 1.29% and 2.83% higher ROC-AUC compared to SOTA graph invariant learning approaches.

2. Related Works

Out-of-Distribution Generalization In the Euclidean domain, such as computer vision, OOD generalization has been studied extensively using a variety of strategies, such as Invariant Risk Minimization (IRM) [2], data augmentation [47], and domain adaptation methods [39]. Unlike deep neural networks with ERM, which suffer performance degradation under data distributional shifts, IRM-based studies [1, 26] have shown powerful and robust performance when providing environmental information. Recently, a two-step optimization framework EIIL [8] has been proposed to train the invariant learning model without explicit environment labels. EIIL trains the environment inference model to learn and infer the environment labels in the first step, and in the second step, EIIL uses inferred environments to conduct invariant learning. Other recent approaches, such as ZIN [29], HRM [31], KerHRM[32], and EDNIL [17] improve the OOD generalization by augmenting the environment to address its high heterogeneity. These augmentation strategies, such as the assistant model, clustering-based model, or effective diversification objective function all enhance the learning stage to infer diverse environments for effective invariant learning.

Graph Invariant Learning While general invariant learning methods are effective for the Euclidean domain, challenges arise when applying them to non-Euclidean domains like graph structures. In the realm of graph invariant learning, since graph data have an inherent nature of complex distribution shifts, often lacks explicit environment labels. To address this issue, methodologies have been widely developed and implemented. Approaches such as EERM [41], LiSA [45], and GREA [30] proposed to generate new environments, offering effective strategies to discover invariant patterns without environment label knowledge. While methods like MoleOOD [43] and GIL[27] focus on the inference of environments, providing solutions to deal with challenging or unexplored environments. Another dimension of exploration involves learning with auxiliary assistance to ensure sufficient variation, with typical techniques found in CIGA [6], GALA[7]. Collectively, these innovative methods offer detailed strategies to navigate the challenges inherent in graph invariant learning.

3. Problem Definition and Preliminaries

In Sec. 3.1, we introduce notations about out-of-distribution generalization and the background knowledge of graph invariant learning. In Sec. 3.2, we introduce environment inference to explain how recent research works on the graph dataset without environmental information.

3.1. Problem Definition

Following general settings of OOD generalization, we assume that the training datasets are collected from multiple training environments. Let $\mathcal{G}_{tr} = \{G^e\}_{e \in \mathtt{supp}(\mathcal{E}_{tr})}$ with different graphs $G^e = \{(G_i^e, y_i^e) \mid 1 \leq i \leq N^e\}$ be the training datasets, where G^e represents the graphs drawn from environment e, and $\mathcal{E}_{tr}$ indicates the environment labels in the training datasets. We denote the environments $\mathcal{E}_{all}$ as all possible environment labels in the real world. Then, the test datasets contain graphs $\mathcal{G}_{test} = \{G^{e'}\}$ with unknown environment labels e', where $e' \in \mathtt{supp}(\mathcal{E}_{all}) \setminus \mathtt{supp}(\mathcal{E}_{tr})$. Our approach addresses out-of-distribution challenges in graph-level classification by incorporating invariant learning within environments derived from complex heterogeneous data hierarchically.

3.2. Preliminaries

Graph Invariant Learning. Recently, several studies have identified the cause of performance deterioration in out-of-distribution (OOD) graphs, attributing it to the introduction of spurious features. It is widely acknowledged that spurious features exhibit sensitivity to environmental changes, often referred to as environmental factors. The incorporation of spurious features diminishes the significance of invariant features, resulting in a decline in OOD performance. As a result, recent research [6, 7, 27, 45] often involves graph decomposition to extract both variant subgraphs G_v and invariant subgraphs G_{inv} from the original graphs G, thereby facilitating the invariant GNN encoder f [24] to learn relationships between invariant features h_{inv} and labels y. Following the invariant risk minimization [2], the encoder f is trained with a regularization term enforcing jointly optimize the encoder f in training environments e, which can generalize to unseen testing environments e':

$$\sum_{e \in \mathtt{supp}\{\mathcal{E}_{tr}\}} \mathcal{R}^e(f) + \lambda ||\nabla \mathcal{R}^e(f)||^2, \tag{1}$$

where $\mathcal{R}^e(f) = \mathbb{E}^e_{G,y}[\mathcal{L}(f(G), y)]$ denotes the risk of f in environment e, and $\mathcal{L}(\cdot,\cdot)$ denotes the loss function.

Environment Inference for GIL. In most cases, environmental information is difficult to obtain, especially in the context of graph generalization. While some datasets may provide metadata that could serve as environmental information, it is not reliable to fully use such information as the ground truth of the environment is not guaranteed. Therefore, existing methods [8, 17, 29, 31, 32] often infer the environmental labels $\mathcal{E}_{infer}$ by using variant subgraphs G_v that are related to the environment. A variant GNN encoder f^e is used to learn the variant features h_v from G_v and infer the label $\hat{e}$ of the environment e. This approach allows joint optimization of environment inference and graph invariant learning through a learning process.

4. Methods

Our method consists of three components: hierarchical stochastic subgraph generation Sec. 4.1, hierarchical semantic environment inference Sec. 4.2, and robust graph invariant learning (GIL) with inferred environments Sec. 4.3. In Sec. 4.1, we generate invariant and variant subgraphs hierarchically for hierarchical semantic environment inference. In Sec. 4.2, we learn hierarchical semantic environments using the proposed hierarchical loss. Finally, in Sec. 4.3, we leverage the inferred hierarchical semantic environments to facilitate robust graph invariant learning, enabling us to uncover the invariant relationships between input graphs and their corresponding labels. These three steps together form a comprehensive approach for enhanced graph analysis and understanding.

4.1. Hierarchical Stochastic Subgraph Generation

Given a graph $G = (V, E)$, where V represents the set of nodes and E represents the set of edges, the goal is to generate the invariant subgraph G_v^k and variant subgraph G_{inv}^k from the original graph G at each hierarchy $k \in K$, where K denotes the number of environmental hierarchies. At each hierarchy k, we employ a graph neural network denoted as GNN_k to update the hidden embedding $h_i^k = \text{GNN}_k(h_i^{k-1}, A)$ for each node $v_i \in V$. We use graph neural networks as GNN_k [42] to aggregate information from neighboring nodes for node embedding updating. In the process of learning variant and invariant subgraphs at each hierarchy, we define a probability distribution function $S^k(\cdot)$ for edge selection as follows. We omit the hierarchy notations, while the computation of probability distribution is conducted within each hierarchy:

$$s_{ij} = S(h_{ij}) = \sigma(\text{MLP}([h_i, h_j])) \tag{2}$$

Here, s_{ij} is the probability of selecting edge e_{ij}, σ is the sigmoid function, and $\text{MLP}(h_i, h_j)$ is a multi-layer perceptron (MLP) that takes node embeddings v_i and v_j as inputs.

To introduce stochasticity in edge selection, we generate a sampler $\mathbf{p}_{ij} \in \{0, 1\}$ from the Bernoulli distribution

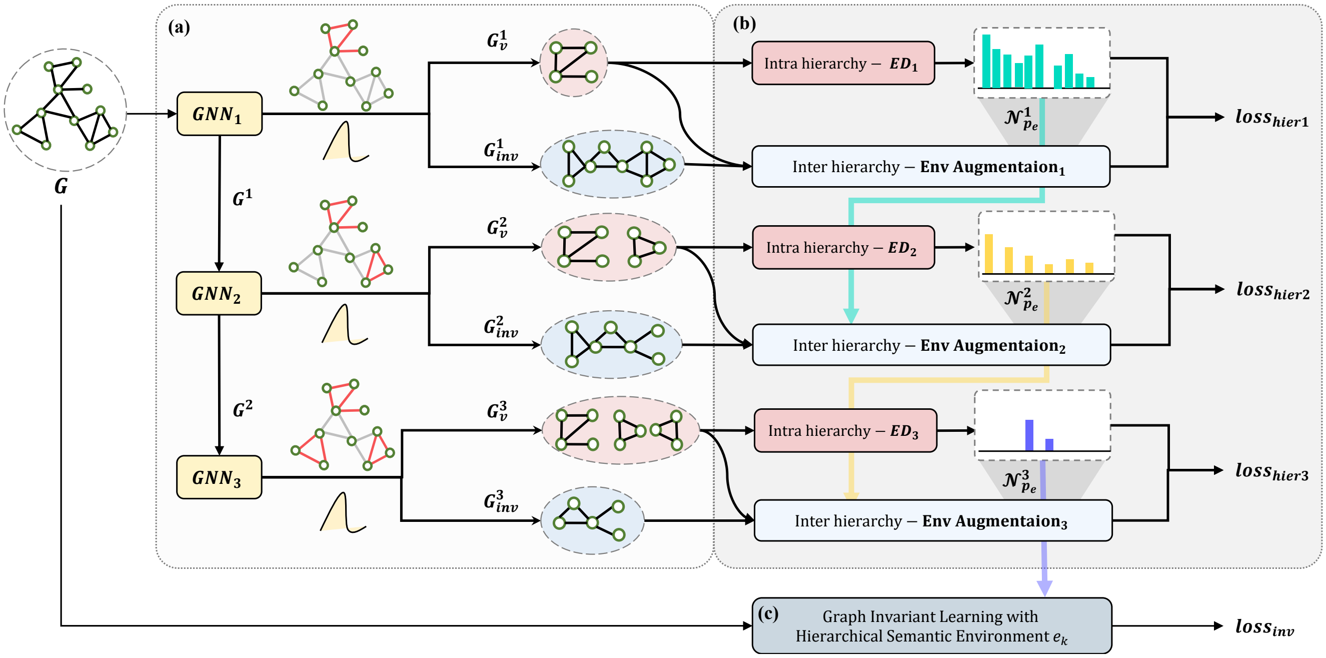

Figure 2. Our Framework consists of (a) Hierarchical Stochastic Subgraph Generation in Sec. 4.1, (b) Hierarchical Semantic Environments in Sec. 4.2, (c) Robust GIL with Hierarchical Semantic Environments in Sec. 4.3.

$\mathbf{p}_{ij} \sim \{\pi_1 := s_{ij}, \pi_0 := 1 - s_{ij}\}$. Gumbel-Softmax [19] is then applied to obtain differentiable edge selection probabilities for each edge:

$$\hat{\mathbf{p}}_{ij} = \frac{\exp\left((\log \pi_1 + \mathbf{g}_1)/\tau\right)}{\sum_{i\in\{0,1\}} \exp\left((\log \pi_i + \mathbf{g}_i)/\tau\right)} \tag{3}$$

Here, $\mathbf{g}_1$ and $\mathbf{g}_0$ are i.i.d variables sampled from the Gumbel distribution, and τ is a temperature parameter that controls the smoothness of the sampling process. A lower τ leads to more categorical (hard) selections, while a higher τ results in smoother (soft) probabilities. As τ approaches 0, $\hat{\mathbf{p}}_{ij}$ can be annealed to a categorical distribution.

To ensure hierarchical neighborhood consistency [38] in the generation of stochastic subgraphs, we introduce an additional hierarchical neighbor masking function denoted as N^k. This masking function compares the neighbor matrix from the previous hierarchy $(k-1)$ with the currently selected neighbors, ensuring that nodes within the hierarchical neighborhood are consistently either included or excluded. The hierarchical neighbor mask N_{ij}^k can be defined as:

$$N_{ij}^k = N_{ij}^{k-1} + \mathbf{1}\{N_{ij}^{k-1} = 0 \text{ and } \mathbf{p}_{ij}^k > T\} \tag{4}$$

where T denotes the threshold for selecting the edges based on $\mathbf{p}_{ij}^{(k)}$, and $\mathbf{1}$ denotes indicator function. Then we obtain the variant and invariant subgraph at each hierarchy k as:

$$A_v^k \leftarrow A \odot N^k, A_{inv}^k \leftarrow A - A_v^k \tag{5}$$

where the adjacency matrix of the variant subgraph, denoted as A_v^k, is derived by performing element-wise multiplication between the original edge matrix A and the hierarchical neighbor mask matrix N^k. Conversely, the adjacency matrix of the invariant subgraph, denoted as A_{inv}^k, is obtained by subtracting the edges included in the variant subgraph A_v^k from the original edge matrix A [5, 7, 27]. By following this process iteratively across hierarchies, we construct a series of variant and invariant subgraphs that capture the hierarchical and structural nuances within the original graph, enabling a multi-level analysis of downstream tasks and analysis.

4.2. Hierarchical Semantic Environments

Since the existing environmental information is not reliable or available, we design an environment inference model to assign the graphs to relatively reliable environments. Given prior $p(e|G)$, we maximize the log-likelihood of $p(y|G)$ to obtain the posterior $p(e|G, y)$. As there is no solution to the true posterior, existing methods use clustering-based or variational distribution to approximate and infer the posterior. Inspired by a multi-class classification problem, we estimate the posterior using the softmax function as:

$$P(\hat{e} \in \mathcal{E}_{tr}|X, Y) = \frac{\exp\left(-l(f^e(\Phi(X), Y))\right)}{\sum_{e'\in \text{supp}(\mathcal{E}_{tr})} \exp(-l(f^{e'}(\Phi(X), Y)))}$$

where f^e denotes neural network with y^e-class classifier and Φ indicates the variant subgraph generator.

4.2.1 Intra-Hierarchy Environment Diversification

In the previous section, we discussed the extraction of variant subgraphs $\{G_v^k | k \in K\}$ utilizing GNN encoders in conjunction with corresponding stochastic masking $\{p_{ij}^k | k \in K\}$ at each hierarchy k. To guide the learning process effectively, we incorporate an intra-hierarchy environment diversification loss with variant subgraph and estimated posteriors, denoted as L_{ED}:

$$L_{ED}^k = -\sum_i \max_{e^k} \log P(e^k | f^k(G_{vi}^k, y_i)) \tag{6}$$

Specifically, L_{ED} is designed to guide the network f^k to maximize the dependency between e^k and Y given the variant subgraph G_v^k at each hierarchy k. By optimizing this loss, the network f^k is ensured to distinguish relationships between the generated environments, fostering diversity within hierarchies.

4.2.2 Inter-Hierarchy Environment Augmentation

The goal of inter-hierarchy environment augmentation is to maximize the intra-class mutual information of the estimated invariant subgraphs while maximizing the intra-environment mutual information of the estimated variant subgraphs. Before introducing our objective function, a standard InfoNCE loss [37] can maximize the similarity between positive pairs and minimize the similarities between some randomly sampled negative pairs, which is defined as:

$$\mathcal{L}_{\text{InfoNCE}}(z, \mathcal{N}_p, \tau) = -\log\left(\frac{\exp(z \cdot \mathcal{N}_p(z)/\tau)}{\sum_{n \in \mathcal{N}(z)} \exp(z \cdot n/\tau)}\right)$$

where $\mathcal{N}_p(z)$ denotes positive samples for z, and $\mathcal{N}(z)$ denotes the batch samples for z, and τ denotes a temperature parameter. To ensure the reliability and diversification of learned hierarchical environments, we introduce two inter-hierarchy semantic invariants: label-invariant and neighborhood-invariant. Inspired by [5, 7], we introduce two contrastive learning losses: (1) Environment-based Contrastive learning loss $\mathcal{L}_{\text{EnvCon}}$ and (2) Label-based Contrastive learning loss $\mathcal{L}_{\text{LabelCon}}$. For environment-based contrastive learning, we define an environment-based neighborhood function $\mathcal{N}_{p_e}^k$ that constructs the positive set for the anchor sample, where samples have the same inferred environment. To preserve neighborhood consistency across hierarchies, the hierarchical neighborhood function $\mathcal{N}_{p_e}^k$ can be defined as :

$$\mathcal{N}_{p_e}^k(z) = \mathcal{N}_{p_e}^{k-1}(z) \cup \{z_i \mid e_i^k = e_z^k, z_i \in \mathcal{N}(z)\} \tag{7}$$

Using $\mathcal{N}_{p_e}^k$, we formulate the environment consistency loss $\mathcal{L}_{\text{EnvCon}}$ as follows:

$$\mathcal{L}_{\text{EnvCon}}^k = \mathbb{E}_{G_v^k \sim p_d}\left[\mathcal{L}_{\text{InfoNCE}}(z_v^k, \mathcal{N}_{p_e}^k(z_v^k), \tau)\right] \tag{8}$$

where p_d denotes the batch of data, and z_v^k denotes the embedding of variant subgraph G_v^k from a multi-layer perception MLP at k-th hierarchy. Environment consistency loss $\mathcal{L}_{\text{EnvCon}}^k$ attracts the variant subgraphs with the same environment labels and pushes the variant subgraphs with different environments, which can augment the power of learning the sufficiency of the environment.

For label-based contrastive loss, we define a label-based neighborhood function that represents the positive samples given ground truth label y, which can be defined as $\mathcal{N}_{p_y}(z) = \{z_i \mid y_i = y_z, z_i \in \mathcal{N}(z)\}$. To measure the label consistency loss, we formulate $\mathcal{L}_{\text{LabelCon}}$ as follows:

$$\mathcal{L}_{\text{LabelCon}}^k = \mathbb{E}_{G_{\text{inv}}^k \sim p_d}\left[\mathcal{L}_{\text{InfoNCE}}(z_{\text{inv}}^k, \mathcal{N}_{p_y}(z_{\text{inv}}^k), \tau)\right] \tag{9}$$

where z_{inv} denotes the embedding of invariant subgraph G_{inv}^k from MLP at k-th hierarchy.

4.2.3 Overall Objective Function

Considering the diversity of intra-hierarchy and consistency of inter-hierarchy, we define a total loss of each hierarchy k as follows:

$$\mathcal{L}_{\text{hier}_k} = \mathcal{L}_{\text{ED}}^k + \alpha \cdot \mathcal{L}_{\text{EnvCon}}^k + \beta \cdot \mathcal{L}_{\text{LabelCon}}^k \tag{10}$$

where α and β denote coefficient parameters. And the overall objective loss can be defined as $\mathcal{L}_{\text{HEI}} = \sum_{k=1}^{K} \mathcal{L}_{\text{hier}_k}$ This formulation ensures a unified optimization objective, allowing the model to learn invariant subgraphs that capture both label and environment information effectively.

4.3. Robust GIL with Hierarchical Semantic Environments

After the hierarchical learning stage of environment generation, we extract the generated environment e^{K-1} from the last hierarchy $K-1$. Our objective is to minimize the invariant risk given by Eq. (1), employing the hierarchically generated semantic environments e^{K-1}. The objective function to calculate $\mathcal{L}_{inv}$ is as follows:

$$\min_f \mathcal{L}_{cls}(f) + \nabla_{\hat{e}}(\mathcal{L}_{cls}(f)) \text{ s.t. } \hat{e} = \arg\min_e \mathcal{L}_{\text{HEI}} \tag{11}$$

This equation represents an optimization problem where the goal is to find a function f that minimizes the sum of two terms. The first term $\mathcal{L}_{cls}(f)$ corresponds to the cross entropy loss of the classification task. The second term $\nabla_{\hat{e}}(\mathcal{L}_{cls}(f))$ represents the gradient with respect to $\hat{e}$, which is a specific environment. In essence, the objective is to discover a function f that performs well on a classification task while maintaining stability in performance across different environments, adhering to the specified environmental minimization condition.

METHODS	IC50-ASSAY	IC50-SCA	IC50-SIZE	EC50-ASSAY	EC50-SCA	EC50-SIZE
#ENV	311	6881	190	47	850	167
ERM[40]	71.79±0.27	68.85±0.62	66.70±1.08	76.42±1.59	64.56±1.25	62.79±1.15
IRM[2]	72.12±0.49	68.69±0.65	66.54±0.42	76.51±1.89	64.82±0.55	63.23±0.56
V-REX[26]	72.05±1.25	68.92±0.98	66.33±0.74	76.73±2.26	62.83±1.20	59.27±1.65
EIIL[8]	72.60±0.47	68.45±0.53	66.38±0.66	76.96±0.25	64.95±1.12	62.65±1.88
IB-IRM[1]	72.50±0.49	68.50±0.40	66.64±0.28	76.72±0.98	64.43±0.85	64.10±0.61
GREA [30]	72.77±1.25	68.33±0.32	66.16±0.46	72.44±2.55	67.98±1.00	63.93±3.01
CIGAV1 [6]	72.71±0.52	69.04±0.86	67.24±0.88	78.46±0.45	66.05±1.29	66.01±0.84
CIGAV2 [6]	73.17±0.39	69.70±0.27	67.78±0.76	-	-	-
MOLEOOD [43]	71.38±0.68	68.02±0.55	66.51±0.55	73.25±1.24	66.69±0.34	65.09±0.90
GALA [7]	-	-	-	79.24±1.36	66.00±1.86	66.01±0.84
OURS	**74.01±0.11**	**70.72±0.30**	**68.64±0.23**	**80.82±0.21**	**69.73±0.21**	**66.87±0.38**

Table 1. Test ROC-AUC of various models on DrugOOD benchmark datasets. The mean ± standard deviation of all models is reported as an average of 5 executions of each model. The best methods are highlighted in **bold** and the second best methods are underlined.

5. Experiments

Datasets. Extensive experiments were conducted on real-world graph data from diverse domains.

- **CMNIST-75sp.** The task is to classify each graph that is converted from an image in the ColoredMNIST dataset [2] into the corresponding handwritten digit using the superpixels algorithm [25]. Distribution shifts exist on node attributes by adding random noises in the testing set.
- **Graph-SST datasets.** We utilize sentiment graph data from SST5 and SST-Twitter datasets from [46]. For the Graph-SST5 dataset, graphs are split into different sets based on averaged node degrees to create distribution shifts. For the Graph-Twitter dataset, we invert the split order to assess the out-of-distribution generalization capability of GNNs trained on large-degree graphs to smaller ones.
- **DrugOOD.** We use six datasets in DrugOOD [20] that are provided with manually specified environment labels. DrugOOD provides more diverse splitting indicators, including assay, scaffold, and size. To comprehensively evaluate the performance of our method under different environment definitions, we adopt these three different splitting schemes on categories IC50 and EC50 provided in DrugOOD. Then we obtain six datasets, **EC50-*** and **IC50-***, where the suffix * specifies the splitting scheme i.e. IC50/EC50-ASSAY/SCAFFOLD/SIZE.

Baselines. We comprehensively compare our methods with the following categories of baselines: (1) ERM denotes supervised learning with empirical risk minimization [40]. (2) Euclidean OOD methods: We compare with SOTA invariant learning methods from the Euclidean regime, including IRM [2], V-REX [26], EIIL [8], IB-IRM [1]. (3) Graph OOD methods: We also compare with SOTA invariant learning methods from the graph regime. Graph OOD methods can be further split into three groups: environment generation-based baseline methods including GREA [30] and LiSA [45], environment augmentation-based baseline methods including DisC [10] and CIGA [6] and environment inference-based baseline methods including GIL [27], MoleOOD [43], and GALA [7].

Environmental Setup. All methods use the same GIN backbone [42] and the same optimization protocol for fair comparisons. Each of the methods is configured using the same parameters reported in the original paper or selected by grid search. For a fair comparison, we use the same embedding size for all methods. We tune the hyperparameters following the common practice. All details are given in Appendix. We report the ROC-AUC score in the DrugOOD dataset and the accuracy score for the rest of the datasets.

5.1. Performance Comparison

We first provide a detailed report on the DrugOOD benchmark dataset in Tab. 1. We conduct IC50 and EC50 predictions with different split settings. As shown in Tab. 1 and mentioned in Sec. 1, the DrugOOD dataset includes molecular datasets with complex distributions, such as assay, scaffold, and size split with various numbers of environments. Our approach consistently outperforms existing methods, demonstrating the importance of hierarchical environment learning in addressing complex drug OOD generalization applications. As shown in Tab. 1, compared to Empirical Risk Minimization (ERM), Euclidean-based invariant learning methods, such as IRM [2], V-REX [26], EIIL [8], etc, show comparable or even degraded performance in most experimental settings. This suggests that di-

METHODS	IC50-SCA	IC50-SIZE
w/ env$_{\text{\#non-infer(rand)=2}}$	68.54±0.64	67.63±0.33
w/ env$_{\text{\#non-infer=real}}$	68.77±0.72	67.60±0.32
w/ env$_{\#\text{infer=2}}$	69.14±0.80	67.55±0.34
w/ env$_{\#\text{infer=real}}$	69.08±0.64	67.74±0.13
w/ env$_{\#\textbf{hier-infer}}$ (OURS)	**70.72±0.30**	**68.64±0.23**

Table 2. The ablation study on IC50-SCA and IC50-SIZE datasets.

CONFIGURATIONS	IC50-SCA	IC50-SIZE
#real = [#e_p]	69.08±0.64	67.60±0.32
#env = [5]	69.35±0.67	67.70±0.50
#env = [2]	69.14±0.80	67.73±0.64
#env=[#$e_p \rightarrow \#e_p/2 \rightarrow 5$]	70.12±0.14	68.62±0.34
#env=[#$e_p \rightarrow \#e_p/2 \rightarrow$ 2]	**70.72±0.30**	**68.64±0.23**

Table 3. Sensitivity analysis on generated environments. #e_p denotes the number of provided environments in datasets.

rectly applying general invariant learning struggles to handle shifts in graph distributions. In contrast, graph-based OOD methods exhibit better performance, as they learn invariant subgraph patterns for graph OOD generalization.

In addition, in the performance comparison of graph-based OOD methods, the MoleOOD [43] pointed out that simple ERM sometimes outperforms several existing methods when faced with a large number of provided environments in graph OOD datasets, which is aligned with the results of Tab. 1. For example, IC50-SCA dataset is provided with 6881 environments, where 6881 scaffolds or substructures construct a vast amount of environmental information for the dataset. In IC50-SCA dataset, GREA and MoleOOD show poor performance compared to ERM, as shown in Tab. 1. We analyze this phenomenon for two potential reasons: (1) The first reason is that the provided environments are overly fine-grained, thereby similar substructure environments in the training and testing sets. ERM can capture similar substructures to learn similar data distributions, showing better performance. (2) The second reason is that GREA and MoleOOD only generate a small number of environments, which overly simplifies the environment and neglects the relationships between environments. This limits their ability to learn heterogeneous and interrelated environments, resulting in noticeable performance degradation.

Our method performs a hierarchical approach to learn relationships between redundant environments and maximize the diversity of environments. Leveraging the high-level semantic environments from the last hierarchy, we minimize invariant prediction risk, thereby achieving better graph OOD generalization. In Appendix, we report performance comparisons showing that our performance is better or comparable to existing methods in general domains.

5.2. Effect of Hierarchical Semantic Environments

In this section, similar to existing environment inference-based works for OOD generalization [7, 8], we first analyze the role of environment inference, and then we discuss the effects of hierarchical environments learned by our model, showing the necessity of learning the hierarchical environments for graph invariant learning.

Environment Inference. As shown in Tab. 2, we analyze the role of environment inference by comparing the direct usage of *non-inferred* and *inferred* environments. Direct usage of the *non-inferred* environments can be divided into two scenarios: (1) When environments are unavailable, existing methods randomly assign two environments for graph invariant learning, which is shown as 'w/ env$_{\text{\#non-infer (rand)}}$' from Tab. 2. (2) DrugOOD dataset provides environmental information, e.g., scaffold information in IC50-SCA dataset. As shown in Tab. 1, we can know the number of environments for both dataset IC50-SCA and dataset IC50-SIZE. In such cases, we use environments for invariant learning, as illustrated by the 'w/ env$_{\text{\#non-infer (real)}}$' in Tab. 2.

Secondly, we consider the usage of *inferred environments* for invariant learning, as indicated in Tab. 2. Through the model of environment inference, we directly predict the flat environment and use it in invariant learning, as illustrated by 'w/ env$_{\#\text{infer=*}}$'. From Tab. 2, it can be observed that the model with inferred environments exhibits better performance, indicating that learned environments from data are more effective than direct usage of *non-inferred* environments. Moreover, in dataset IC50-SCA, the effects of inference are more pronounced, emphasizing the necessity of performing environment inference operations in situations with complex real-world environments.

Hierarchical Environment Inference. We further analyze the effects of *hierarchical* environments by comparing them with *flat* environments. Our model adopts a hierarchical approach to learning environmental information, regulating semantic environmental content between hierarchies through both intra-hierarchy and inter-hierarchy mechanisms. Upon comparing experimental results in IC50-SCA and IC50-SIZE datasets, as shown in Tab. 2, our hierarchical model consistently outperforms invariant learning models with flat environments. This indicates that in real-world datasets, the environmental factors influencing label predictions are complex and interdependent. Consequently, investigating graph invariant learning with hierarchical environment inference is essential for attaining more sophisticated and effective in graph OOD generalization.

5.3. Sensitive analysis on hierarchy

We also conduct sensitive analysis on the number of hierarchies and the number of environments at each hierarchy. Specifically, in hierarchical settings, we start from the number of environments as the number of environments provided in the first hierarchy for a fair comparison. By comparing the results shown in Tab. 3, we find inferred environments with hierarchical settings help OOD generalization in the graph domain. Inferring flat environments is worse than environment inference with three hierarchies. In addition, inferring a large number of environments $\#e_p$ in a single hierarchy shows poor performance compared to inferring a small number of environments. This is because the model with a large number of environments learns too local and redundant to capture the relationships between environments. However, in a hierarchical setting, even if inference starts from a large number of environments to a small number of environments, different hierarchies can capture the local relationships among a huge number of environments and dynamically maximize the diversity of inferred high-level environments. Therefore, despite the final number of inferred environments is still small, these environments learn the semantic information from complex data distribution hierarchically and can improve the Graph OOD generalization using the high-level semantic environments.

5.4. Discussions

We discuss the diversity of environments generated by different methods as shown in Fig. 3. we compare two environment assignment methods with our method, which are random sampling-based method *rand#2*, and flat environment inference-based method *infer#2*, respectively. For a fair comparison, we set two environments for both methods and set the hierarchical environment as $[6881 \rightarrow 3440 \rightarrow 2]$ for our method using the IC50-SCA dataset from DrugOOD.

As mentioned in [8], the diversity of environments is important to obtain effective IRM models, and large discrepancy of spurious correlations between environments benefits the IRM model. We measure the diversity of environments by calculating the distance of distributions among the acquired environments for each method. More specifically, we measure the Kolomogorov-Smirnov (K-S) statistic [34] as a distance between the two inferred scaffold distributions for each method, respectively. The K-S test can capture the dissimilarity between cumulative distribution functions, making it suitable for our analysis. As shown in Fig. 3, the two environments generated by *rand#2* methods are almost the same since they assign the sample to two environments using random sampling, showing the lowest diversity of environments. *infer#2* shows better diversity of generated environment distributions with significant p-values ($p = 1.86e - 26$). Our method generates more significantly diverse environment distributions ($p = 4.52e - 73$) compared to other methods, surpassing them in *inter-env-distance* by a large margin. Moreover, the results are aligned with Fig. 1, our methods outperform other methods in IC50-SCA dataset, indicating that diverse environments can give a clear indication of distribution shifts, therefore IRM can easily identify and eliminate variant features.

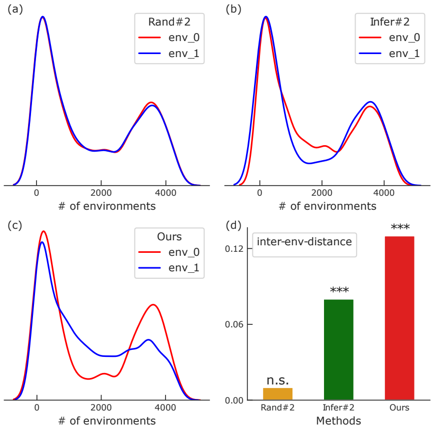

Figure 3. Discussions on the diversity of generated environments. We show distributions of two generated environments env_0 and env_1 for (a) random sampling methods, (b) flat environment inference methods, and (c) our hierarchical environment inference methods. (d) We employ the Kolmogorov-Smirnov test [34] to calculate the diversity of three methods.

6. Conclusion

In our research, we take on the formidable challenge of improving out-of-distribution (OOD) generalization in the graph domain. We observe that recent works often overlook the crucial aspect of studying hierarchical environments in graph invariant learning. To address this gap, we introduce a novel method that generates hierarchical dynamic environments for each graph. Our approach involves hierarchical stochastic subgraph generation, hierarchical environment inference, and a carefully designed learning objective. By incorporating graph invariant learning with inferred high-level environments, our model not only achieves meaningful and diverse environments within the same hierarchy but also ensures consistency across different hierarchies. The effectiveness of our method is particularly pronounced in the DrugOOD dataset, shedding light on the potential for further exploration in hierarchical graph learning within OOD scenarios.

7. Acknowledgement

This work was supported by Bio & Medical Technology Development Program of the NRF funded by the Ministry of Science & ICT(NRF-2022M3E5F3085677, RS-2023-00257479), and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2023-00246586).

References

[1] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34: 3438–3450, 2021. 2, 6, 12

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1, 2, 3, 6, 12, 14

[3] Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. Size-invariant graph representations for graph classification extrapolations. In *International Conference on Machine Learning*, pages 837–851. PMLR, 2021. 1

[4] Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing*, 9:e15, 2020. 1

[5] Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, MA KAILI, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Invariance principle meets out-of-distribution generalization on graphs. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*, 2022. 2, 4, 5

[6] Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, MA Kaili, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Learning causally invariant representations for out-of-distribution generalization on graphs. *Advances in Neural Information Processing Systems*, 35:22131–22148, 2022. 2, 3, 6, 11, 12

[7] Yongqiang Chen, Yatao Bian, Kaiwen Zhou, Binghui Xie, Bo Han, and James Cheng. Does invariant graph learning via environment augmentation learn invariance? *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3, 4, 5, 6, 7, 11, 12

[8] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021. 2, 3, 6, 7, 8, 12

[9] Laurianne David, Amol Thakkar, Rocío Mercado, and Ola Engkvist. Molecular representations in ai-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, 12(1):1–22, 2020. 1

[10] Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang. Debiasing graph neural networks via learning disentangled causal substructure. *Advances in Neural Information Processing Systems*, 35:24934–24946, 2022. 2, 6, 12, 14

[11] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *ACL 2018*, page 1, 2018. 12

[12] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. Good: A graph out-of-distribution benchmark. *Advances in Neural Information Processing Systems*, 35:2059–2073, 2022. 1

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 1

[14] Peng Han, Peng Yang, Peilin Zhao, Shuo Shang, Yong Liu, Jiayu Zhou, Xin Gao, and Panos Kalnis. Gcn-mf: disease-gene association identification by graph convolutional networks and matrix factorization. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 705–713, 2019. 1

[15] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960, 2020. 1

[16] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020. 1

[17] Bo-Wei Huang, Keng-Te Liao, Chang-Sheng Kao, and Shou-De Lin. Environment diversification with multi-head neural network for invariant learning. *Advances in Neural Information Processing Systems*, 35:915–927, 2022. 2, 3, 11, 14

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 12

[19] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[20] Yuanfeng Ji, Lu Zhang, Jiaxiang Wu, Bingzhe Wu, Lanqing Li, Long-Kai Huang, Tingyang Xu, Yu Rong, Jie Ren, Ding Xue, et al. Drugood: Out-of-distribution dataset curator and benchmark for ai-aided drug discovery–a focus on affinity prediction problems with noise annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8023–8031, 2023. 1, 6, 11

[21] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR, 2018. 1

[22] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. 12

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12

[24] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016. 1, 3

[25] Boris Knyazev, Graham W Taylor, and Mohamed Amer. Understanding attention and generalization in graph neural networks. *Advances in neural information processing systems*, 32, 2019. 6

[26] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021. 2, 6, 12

[27] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning invariant graph representations for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35:11828–11841, 2022. 2, 3, 4, 6, 12, 14

[28] Sangsoo Lim, Sangseon Lee, Yinhua Piao, MinGyu Choi, Dongmin Bang, Jeonghyeon Gu, and Sun Kim. On modeling and utilizing chemical compound information with deep learning technologies: A task-oriented approach. *Computational and Structural Biotechnology Journal*, 2022. 1

[29] Yong Lin, Shengyu Zhu, Lu Tan, and Peng Cui. Zin: When and how to learn invariance without environment partition? *Advances in Neural Information Processing Systems*, 35: 24529–24542, 2022. 2, 3

[30] Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph rationalization with environment-based augmentations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1069–1078, 2022. 2, 6

[31] Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. Heterogeneous risk minimization. In *International Conference on Machine Learning*, pages 6804–6814. PMLR, 2021. 2, 3

[32] Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. Kernelized heterogeneous risk minimization. *arXiv preprint arXiv:2110.12425*, 2021. 2, 3

[33] Yang Liu, Xiang Ao, Fuli Feng, Yunshan Ma, Kuan Li, Tat-Seng Chua, and Qing He. Flood: A flexible invariant learning framework for out-of-distribution generalization on graphs. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1548–1558, 2023. 2

[34] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951. 8

[35] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, et al. Chembl: towards direct deposition of bioassay data. *Nucleic acids research*, 47(D1):D930–D940, 2019. 11

[36] Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. Delving into out-of-distribution detection with vision-language representations. *Advances in Neural Information Processing Systems*, 35:35087–35102, 2022. 1

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5

[38] Yinhua Piao, Sangseon Lee, Dohoon Lee, and Sun Kim. Sparse structure learning via graph neural networks for inductive document classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11165–11173, 2022. 4

[39] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 443–450. Springer, 2016. 2

[40] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991. 6

[41] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. *arXiv preprint arXiv:2202.02466*, 2022. 2

[42] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018. 1, 3, 6, 11, 12

[43] Nianzu Yang, Kaipeng Zeng, Qitian Wu, Xiaosong Jia, and Junchi Yan. Learning substructure invariance for out-of-distribution molecular representations. *Advances in Neural Information Processing Systems*, 35:12964–12978, 2022. 2, 6, 7

[44] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018. 2

[45] Junchi Yu, Jian Liang, and Ran He. Mind the label shift of augmentation-based graph ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11620–11630, 2023. 2, 3, 6, 12, 14

[46] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):5782–5799, 2022. 6

[47] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[48] Yanfu Zhang, Hongchang Gao, Jian Pei, and Heng Huang. Robust self-supervised structural graph neural network for social network prediction. In *Proceedings of the ACM Web Conference 2022*, pages 1352–1361, 2022. 1

[49] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020. 1