Graph Invariant Learning with Subgraph Co-mixup for Out-of-Distribution Generalization

Tianrui Jia[1], **Haoyang Li**[2], **Cheng Yang**[1*], **Tao Tao**[3], **Chuan Shi**[1*]
[1]Beijing University of Posts and Telecommunications
[2]Tsinghua University
[3]China Mobile Information Technology Co. Ltd.
{jiatianrui, yangcheng, shichuan}@bupt.edu.cn, lihy18@mails.tsinghua.edu.cn, taotao@chinamobile.com

Abstract

Graph neural networks (GNNs) have been demonstrated to perform well in graph representation learning, but always lacking in generalization capability when tackling out-of-distribution (OOD) data. Graph invariant learning methods, backed by the invariance principle among defined multiple environments, have shown effectiveness in dealing with this issue. However, existing methods heavily rely on well-predefined or accurately generated environment partitions, which are hard to be obtained in practice, leading to sub-optimal OOD generalization performances. In this paper, we propose a novel graph invariant learning method based on invariant and variant patterns co-mixup strategy, which is capable of jointly generating mixed multiple environments and capturing invariant patterns from the mixed graph data. Specifically, we first adopt a subgraph extractor to identify invariant subgraphs. Subsequently, we design one novel co-mixup strategy, i.e., jointly conducting environment mixup and invariant mixup. For the environment mixup, we mix the variant environment-related subgraphs so as to generate sufficiently diverse multiple environments, which is important to guarantee the quality of the graph invariant learning. For the invariant mixup, we mix the invariant subgraphs, further encouraging to capture invariant patterns behind graphs while getting rid of spurious correlations for OOD generalization. We demonstrate that the proposed environment mixup and invariant mixup can mutually promote each other. Extensive experiments on both synthetic and real-world datasets demonstrate that our method significantly outperforms state-of-the-art under various distribution shifts.

Introduction

Graph data is ubiquitous in the real world, such as molecular networks, protein networks, social networks. Graph representation learning (Chen et al. 2020; Hamilton, Ying, and Leskovec 2017b) achieves deep learning on graphs by encoding them into vectors in a latent space. Graph neural networks (GNNs) (Kipf and Welling 2016; Xu et al. 2018; Veličković et al. 2018; Hamilton, Ying, and Leskovec 2017a), as one of the most popular graph representation learning methods, have attracted wide attention in the last decade. (Lee, Rossi, and Kong 2018; Xu et al. 2018).

*Corresponding authors.

Despite their noticeable success, existing GNNs heavily rely on the identically distributed (I.D.) assumption (Vapnik 1999), i.e., the training and test data are sampled from an identical distribution. However, various forms of distribution shifts between the training and testing datasets widely exist in the real world, since the uncontrollable data generation mechanisms, resulting in OOD (Hu et al. 2020a; Ji et al. 2022; Koh et al. 2021) scenarios. For instance, in graph classification tasks, there could be significant distribution shifts existing in graph size (Bevilacqua, Zhou, and Ribeiro 2021; Yehudai et al. 2021), node degree (Yoo et al. 2023), and structure (e.g., molecule scaffold) (Ji et al. 2022) between the training and testing graphs. Existing GNNs that perform well on the training data by capturing the spurious correlations significantly fail to generalize to OOD testing graph data. Therefore, it is of paramount importance to capture the invariant relationships between predictive graph patterns and labels.

Invariant learning (Arjovsky et al. 2019; Krueger et al. 2021; Creager, Jacobsen, and Zemel 2021; Ahuja et al. 2021) emerges as a prevalent strategy for tackling the challenge of generalization to OOD data. The basic assumption of invariant learning method is the invariance principle among defined multiple environments, namely there existing a proportion of input data capturing invariant relations with the labels across distinct environments (Arjovsky et al. 2019). Consequently, a predictor that performs well across multiple pre-defined environments is guaranteed to possess generalization capabilities for unseen data distributions (Arjovsky et al. 2019). In the field of graphs, existing graph invariant learning methods (Wu et al. 2021; Miao, Liu, and Li 2022; Li et al. 2022b; Chen et al. 2022a; Yang et al. 2022) consider that within each environment the graph data can be decomposed into two components, including invariant subgraphs that have deterministic and truly predictive relations with the labels, and environment subgraphs that could exhibit spurious correlations with the labels. The main goal of them is focused on obtaining diverse training environments. For example, DIR (Wu et al. 2021) generates multiple training environments for invariant learning by implementing distribution interventions on graphs, while GIL (Li et al. 2022b) clusters environment subgraphs and treats each cluster as an environment. The performance of invariant learning heavily relies on the diversity of environment partitioning. In other words,

if different environments are not diverse, these methods will not sufficiently get rid of spurious correlations, showing poor OOD generalization ability. For example, when there exists size distribution shift between the training set and the test set, suppose the size of the graphs in the training set is 6-8, while the size of the graphs in the test set is 30-50, then no matter how environments are partitioned within the training set, it would be challenging for the model to demonstrate satisfactory generalization capabilities in the test set. However, the environments generated by existing graph invariant learning methods cannot possess sufficient distribution shifts among different environments. For DIR (Wu et al. 2021), although it is theoretically possible to mitigate the influence of the environment, all environments are relatively similar in the initial stages of training, performing suboptimally in practice since environments can not be proved to be diverse (Chen et al. 2022b). Similarly, for GIL (Li et al. 2022b), if the training set itself does not have significantly diverse latent environments, the generated environments during the training process will also not be enough to learn invariant patterns. In addition to these two representative methods, existing methods are generally hard to achieve a diverse environment partitioning, resulting in suboptimal performance and largely hindering the OOD generalization.

To tackle these problems, in this paper, we are the first to study mixup-based graph invariant learning for graph OOD generalization, to the best of our knowledge. Although Mixup (Zhang et al. 2018) and their variations (Verma et al. 2019; Chou et al. 2020; Kim et al. 2020; Yun et al. 2019), as one type of interpolation-based data augmentation methods that amalgamate two training instances and the labels to generate new instances are proposed in the literature, the existing graph mixup methods (Han et al. 2022; Wang et al. 2021; Park, Shim, and Yang 2022; Guo and Mao 2021) are only based on mixing up entire graphs, which can definitely introduce spurious correlations since they do not explicitly distinguish invariant and environment subgraphs during conducting mixup, so as to degrade the model's generalization performance on OOD graph data. Incorporating mixup with invariant learning for graph out-of-distribution generalization is promising but poses great challenges as follows and has not been explored:

- How to design mixup to generate diverse enough environments which have enough distribution shifts for invariant learning.
- How to improve the mixup method so that the mixed-up graph data only retains invariant information while excluding environmental-related spurious correlations for OOD generalization.

To address the aforementioned challenges, we propose a novel graph invariant learning method based on invariant and variant co-mixup strategy, herein referred to as **I**nvairnt learning on **G**raph with co-**M**ixup (IGM)[1]. Firstly, we design an invariant subgraph extractor to identify the invariant subgraphs and consider their complements as the environment-related environment subgraphs. Then, we design an environment Mixup module based on the environment subgraphs to encourage the generated environments that are sufficiently diverse for graph invariant learning. We generate a variety of environments by concatenating invariant and environment subgraphs with different labels. The environments generated in this tailored way will have sufficient distribution shifts so as to be diverse enough. Next, in order to ensure that the mixed graph data only retains invariant information, we design an invariant Mixup module to perform mixup only on invariant subgraphs rather than the whole graphs. Performing invariant and environment subgraph co-mixup with these two modules above can effectively get rid of spurious correlations from the entire graph. More importantly, we also show that environment Mixup and invariant Mixup modules of the co-mixup strategy can mutually promote each other, for the promising performances of the OOD generalization capabilities.

We conducted extensive experiments on three artificially synthesized datasets and nine real-world datasets to verify the effectiveness of our proposed method for various types of distribution shifts. Compared to the state-of-the-art baselines, our method shows significant improvements, e.g., an average of 7.4% improvement on real-world datasets. Furthermore, we have verified the effectiveness of each module and performed visualization experiments on the learned invariant subgraphs to conduct deeper analyses. Our contributions can be summarized as follows: (1) We design an invariant and environment subgraph co-mixup based graph invariant learning method for OOD generalization. To the best of our knowledge, this is the first work to automatically generate enough diverse environments for graph invariant learning. (2) We design an environment Mixup module to generate environments which have enough distribution shifts, leading to better invariant learning on graphs. (3) We propose an invariant Mixup method to encourage the mixed-up data only retain invariant graph patterns. This novel design mitigates the impact of spurious correlations in the whole graph. We demonstrate that our designed environment Mixup and invariant Mixup can mutually promote each other in practice, thereby enhancing the generalization capability on OOD graph data. (4) We conduct extensive experiments on both synthetic and real-world datasets to show that our proposed method has the most competitive OOD generalization ability via significantly outperforming state-of-the-art on various types of distribution shifts.

Related Works

Graph Neural Network. Graph Neural Networks (GNNs) (Kipf and Welling 2016; Hamilton, Ying, and Leskovec 2017a; Xu et al. 2018; Veličković et al. 2018) aggregate the neighbors of nodes through a message-passing mechanism to obtain individual node representations. Subsequently, a pooling function is employed to derive a global graph representation, which is then utilized for subsequent classification tasks. Inspired by the spectral method (Bruna et al. 2014; Defferrard, Bresson, and Vandergheynst 2016), GNN is designed to use convolutional neural networks to aggregate neighbors' features (Kipf and Welling 2016; Hamilton, Ying, and Leskovec 2017a). Due to the good performance of the attention mechanism, attention

[1]Code available at https://github.com/BUPT-GAMMA/IGM

is introduced to GNN, which is known for GAT (Veličković et al. 2018). However, traditional GNNs fail to achieve generalization on OOD data.

OOD Generalization on Graphs. Currently, graph OOD generalization methods (Xia et al. 2023; Miao, Liu, and Li 2022; Li et al. 2022b; Wu et al. 2021; Yang et al. 2022; Liu et al. 2022; Sui et al. 2022; Buffelli, Liò, and Vandin 2022; Zhang et al. 2022; Chen, Xiao, and Kuang 2022; Li et al. 2022a) can be primarily categorized into two approaches. The first, based on information bottleneck methods such as CIGA (Chen et al. 2022a) and GSAT (Miao, Liu, and Li 2022), achieves generalization by maximizing mutual information between labels and invariant subgraphs while minimizing mutual information between the subgraph and the entire graph. The second approach, based on invariant learning methods (Arjovsky et al. 2019; Krueger et al. 2021; Creager, Jacobsen, and Zemel 2021; Ahuja et al. 2021), like DIR (Wu et al. 2021) and GIL (Li et al. 2022b), defines environments within datasets and incorporates a regularization term between these environments. This method aims to learn cross-environment invariant information, thereby facilitating OOD generalization.

Notations and Preliminaries

Notations. Denote a graph dataset as $\mathcal{G} = \{G_i, Y_i\}_{i=1}^N$. Due to the uncontrollable data generation mechanism (Bengio et al. 2020), we follow the literature (Arjovsky et al. 2019; Ahuja et al. 2021) to consider realistic yet challenging scenarios that there exist unobservable distribution shifts between training and test sets $\mathrm{P}(\mathrm{G}_{train}) \neq \mathrm{P}(\mathrm{G}_{test})$ since the training and test graph data are sampled from different environments. The label space of graphs and labels are $\mathbb{G}$, $\mathbb{Y}$.

Problem Formulation. Following existing works (Wang et al. 2021; Li et al. 2022b), we assume each graph G_i consists of two parts, namely invariant subgraph G_i^I and environment subgraph G_i^E, where G_i^E is the complement of G_i^I. Denote the invariant subgraph set as $\mathcal{G}^I = \{G_i^I, Y_i\}_{i=1}^N$ and environment subgraph set as $\mathcal{G}^E = \{G_i^E, Y_i\}_{i=1}^N$. We use the subscript to denote the corresponding train and test set, i.e., $\mathcal{G}^I_{train}$,$\mathcal{G}^E_{train}$,$\mathcal{G}^I_{test}$,$\mathcal{G}^E_{test}$. G_i^I determine its label Y_i so that it has invariant relations with the label and should be captured for OOD generalization, i.e., $\mathrm{P}(\mathrm{Y}_{train}|\mathrm{G}^I_{train}) = \mathrm{P}(\mathrm{Y}_{test}|\mathrm{G}^I_{test}) = \mathrm{P}(\mathrm{Y}|\mathrm{G}^I)$, where G^I_{train}, G^I_{test}, G^I denotes the random variables for $\mathcal{G}^I_{train}$, $\mathcal{G}^I_{test}$, $\mathcal{G}^I$. In contrast, G_i^E contains information which has spurious relation with Y_i so that it has variant relations with the label and should be got rid of for stable performances among different environments.

Thus, our objective is to identify the invariant subgraphs within the graph and only use them to make OOD generalized predictions. By extracting the right invariant subgraph of each graph, our model will generalize well in testing environment (Li et al. 2022b; Chen et al. 2020; Wu et al. 2021).

Methodology

In this section, we first present the overall framework of our IGM. Then we will introduce our invariant subgraph extractor. Finally, we will describe the two mixup modules based on the extracted subgraphs, namely environment Mixup and invariant Mixup, An overview of IGM is shown in Figure 1.

Overall Framework

To tackle the limitations of the existing graph invariant learning methods' strong dependence on the predefined environment partitions, we propose to incorporate mixup (Zhang et al. 2018) and invariant learning to generate mixed environments and capture invariant patterns from mixed graphs simultaneously. Given the input data, we first use an invariant subgraph extractor to extract the invariant and environment subgraphs from each graph. Subsequently, we apply environment Mixup and invariant Mixup to update the parameters of the invariant subgraph extractor.

Specifically, the environment Mixup module is designed to generate environments with sufficient distribution shifts and the invariant Mixup module is proposed to prevent spurious correlations within the graph from affecting the mixup. Note that these two modules can mutually enhance each other's learning: on the one hand, the environment Mixup module is able to partition environments with sufficient distribution shifts, thereby facilitating the invariant Mixup to capture more invariant information. On the other hand, as the invariant Mixup captures more invariant information, it can further aid the environment Mixup in achieving a more refined environmental partition, subsequently promoting the invariant learning of the environment Mixup.

Invariant Subgraph Extractor

We use g to represent the subgraph extractor, $G_i^I = g(G_i)$, corresponding to the invariant feature extractor g in the previous section. The idealized invariant subgraph extractor $g^*(\cdot)$ should satisfy:

$$\begin{aligned} \mathrm{P}_{e_1}(\mathrm{Y}|g^*(\mathrm{G})) = \mathrm{P}_{e_2}(\mathrm{Y}|g^*(\mathrm{G})) \quad , \forall e_1, e_2 \in \mathcal{E}, \\ \mathcal{R}(f \circ g^*(\mathrm{G})) = \min \mathcal{R}(f \circ g(\mathrm{G})), \end{aligned} \tag{1}$$

where $\mathcal{E}$ is the set of environments. $\mathcal{R}(\cdot)$ is the risk function that can be a cross-entropy, and f represents the classifier.

Now we instantiate g with learnable parameters. For a given graph G, its nodes set and edges set are V_G and E_G respectively. The $p_{(u,v)}$ represents the probability that edge (u, v) is selected as an edge in the invariant subgraph G_I, and we get it via a GNN_{enc} and an MLP_{enc}:

$$\begin{aligned} \Omega &= \mathrm{GNN}_{enc}(G), \\ \phi_{(u,v)} &= \Omega_u \parallel \Omega_v, \\ p_{(u,v)} &= \mathrm{MLP}_{enc}(\phi_{(u,v)}), \end{aligned} \tag{2}$$

where Ω is the nodes representations, Ω_u, Ω_v is the representation of node u, v.

Next, we sample edges based on distribution $\xi_{uv} \sim Bern(p_{uv})$ to get G_I. Due to the non-differentiability of this sampling process, we employ the Gumbel Softmax (Jang, Gu, and Poole 2016) technique to make it differentiable.

In practice, we set a maximum edge ratio r to avoid the extracted subgraph being overly large. With our subgraph extractor, we can adaptively select edges instead of selecting a fixed ratio of nodes or edges. In the subsequent experimental section, we will report the values of r.

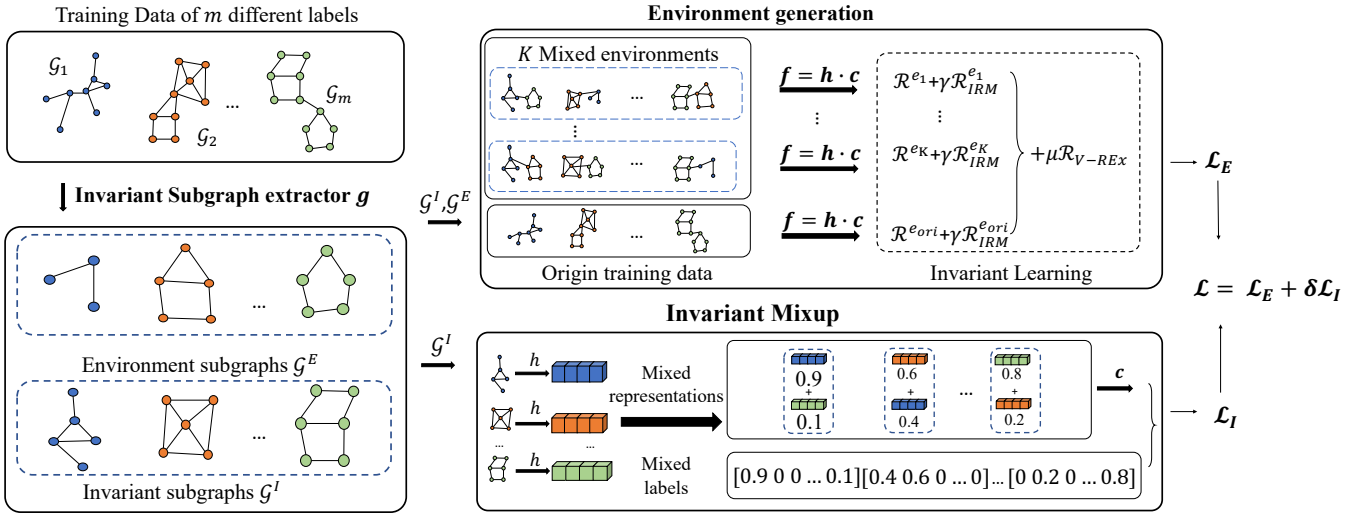


Figure 1: The overall framework of our proposed IGM. Invariant subgraph extractor g splits each graph into invariant subgraph and environment subgraph. Following this, we employ two mixup strategies: (1) Concatenating invariant subgraphs and environment subgraphs from different labels to generate K new environments, upon which we conduct invariant learning. (2) Mixing invariant subgraphs from different labels to augment the data. Here, h represents a GNN used for feature extraction, and c denotes an MLP utilized for classification.

Mixup for Out-of-Distribution Generalization

Following the extraction of the invariant and environment subgraphs, we proceed to apply two distinct types of mixup to them, namely Invariant Mixup and Environment Mixup.

Environment Mixup. Here we design an Environment Mixup method to generate diverse enough environments which have sufficient distribution shift with origin data, which makes invariant learning on graphs more effective. Furthermore, the Environment Mixup enhances the Invariant Mixup, which will be mentioned in the following section.

Let $L : \mathbb{Y} \to \mathbb{Y}$ be a mapping function that maps a label to a different one. For each $G_i^I \in \mathcal{G}_{train}^I$, we random select an $G_j^E \in \mathcal{G}_{train}^E$, whose label $Y_j = L(Y_i)$. Then we mixup the two subgraphs by randomly adding edges between the two graphs according to the node degree. The number of edges added is $n_{add}^{G_i^I G_j^E} = r_{add}(|E_{G_i^I}| + |E_{G_j^E}|)$, and r_{add} is a pre-defined ratio. Since the invariant subgraphs determine the label, we define the label of the augmented graph as G_i^I's label Y_i. We can obtain multiple label mapping functions and augmentations for data with multiple classes. For K augmentations, denotes the k-th label function as L_k. We define the k-th augmentation as $\mathcal{G}_{aug}^k$:

$$\mathcal{G}_{aug}^k = \{\text{Mix}(G_i^I, G_j^E) | G_i^I \in \mathcal{G}_{train}^I, G_j^E \in \mathcal{G}_{train}^E, \\ Y_j = L_k(Y_i), i = 1, 2, 3..., |\mathcal{G}_{train}^I|\}. \tag{3}$$

Since G_j^E is only spuriously related with Y_j, graph G_i^I concatenated with G_j^E is more different than that with its own spurious subgraph G_i^E or with a same substructure. So we consider each augmentation $\mathcal{G}_{aug}^k$ as an environment that has obvious distribution shifts with the origin training set.

After obtaining K augmented environments, we adopt invariant learning on them to enable our model to learn invariant information across environments and extract correct invariant subgraphs that satisfy previous assumptions.

Drawing from the invariant learning literature (Chen et al. 2022b), combining different invariant regularizers can improve the generalization ability of modes further. During our training of Environment Mixup, V-REx (Krueger et al. 2021) regularizer is less impactful initially due to similar environments, while the IRM (Arjovsky et al. 2019) contributes more to the optimization procedure. But as spurious correlations increase in later stages, the effectiveness of IRM regularizer reduces, while V-REx gains importance. Hence, using both regularizers together leads to a better ability of generalization. We then formulate the overall risk following the IRMX (Chen et al. 2022b) literature:

$$\mathcal{L}_E = \sum_{e=0}^{K} \left(\mathcal{R}^e(f) + \gamma \mathcal{R}^e_{\text{IRM}} \right) + \mu \mathcal{R}_{\text{V-REx}}, \tag{4}$$

where K is the number of environments, and $e = 0$ represents the orignal data. f is the classifier, and $\mathcal{R}^e(f)$ represents CrossEntropy loss on environment e. γ is the weight of IRM regularizer $\mathcal{R}^e_{\text{IRM}} = \| \nabla_{w|w=1.0} \mathcal{R}^e(w \cdot f) \|^2$, and μ is the weight of of V-REx regualrizer $\mathcal{R}_{\text{V-REx}} = \text{Var}_e(\mathcal{R}^e(f))$. Var$(\cdot)$ denotes the variance of risks over the environments. We instantiate f with GNN$_{fea}$ and MLP$_{cls}$ as follows:

$$\Psi = \text{GNN}_{fea}(G), \psi = \text{Pooling}(\Psi), \\ \hat{Y} = \text{SoftMax}(\text{MLP}_{cls}(\psi)), \tag{5}$$

where Ψ is the node representation of G and Pooling is the readout function. For clarity of presentation, we denote Pooling(GNN$_{fea}(G)$) as h, MLP$_{cls}$ as c in Figure 1.

In the previous discussions, mixup was primarily utilized as a data augmentation technique to promote invariant learning in different environments. However, mixup also serves as a good regularizer for improving model generalization. In the following section, we will present how to leverage the extracted invariant subgraphs to perform mixup, thereby further enhancing the generalization capability of the model on out-of-distribution data.

Invariant Mixup. Recent studies (Pinto et al. 2022) show that Mixup based method leads to learning models exhibiting high entropy throughout, and consequently, Mixup method can improve the model performance on out-of-distribution data. In other words, Mixup is a good regularizer for out-of-distribution generalization.

Existing methods of Mixup on graph (Han et al. 2022; Wang et al. 2021; Park, Shim, and Yang 2022; Guo and Mao 2021) do Mixup operation on the whole graph, while we only perform Mixup method on casual subgraphs. Mixup only on invariant subgraphs enhances the performance. Applying mixup across the entire graph could potentially disrupt the real relationships. However, implementing mixup on invariant subgraphs allows for a more precise preservation and learning of original invariant relationships, reducing the occurrence of erroneous learning. In other words, the mixed-up invariant subgraphs retain as much invariant information as possible and effectively prevent the impact of noise and spurious correlations from the entire graph on classification tasks.

We adopt Manifold Mixup on invariant subgraphs we extract in the previous part. We obtain the invariant subgraph representations ψ_i^I and ψ_j^I for G_i and G_j as:

$$\Psi^I = \text{GNN}_{fea}(G^I), \psi^I = \text{Pooling}(\Psi^I), \tag{6}$$

where Ψ^I is the node representation of G^I.

The labels for G_i and G_j are Y_j and Y_j respectively. Our definition of invariant Mixup is as follows:

$$\begin{aligned} \psi_{i,j}^I &= \lambda \psi_i^I + (1-\lambda)\psi_j^I, \\ Y_{i,j} &= \lambda Y_i + (1-\lambda) Y_j, \\ \lambda &\sim Beta(\alpha, \alpha), \end{aligned} \tag{7}$$

where $\psi_{i,j}^I$ is the mixed representation of G_i^I and G_j^I. $Y_{i,j}$ is the mixed label of Y_i and Y_j. λ is derived from the Beta distribution with the parameter α. The loss function of invariant Mixup can be defined as:

$$\begin{aligned} \mathcal{L}_I &= \text{CrossEntropy}(Y_{ij}, \hat{Y_I}), \\ \hat{Y_I} &= \text{SoftMax}(\text{MLP}_{cls}(\psi_{i,j}^I)), \end{aligned} \tag{8}$$

where $\hat{Y_I}$ is the predicted label with the mixed representation.

Overall, we can jointly optimize these components via the environment loss and invariant loss, i.e., $\mathcal{L} = \mathcal{L}_E + \delta \mathcal{L}_I$, where δ is the balance hyper-parameter.

Experiments

In this section, we conduct experiments on 11 datasets to answer the following research questions: · **RQ1**: Is IGMeffective on the graph OOD generalization problem? · **RQ2**: Is it necessary to use two kinds of Mixup? · **RQ3**: How about the hyper-parameter sensitivity of IGM? · **RQ4**: Does the learned invariant subgraph capture invariant information, and does it capture better invariant patterns compared to other methods?

Experimental Setup

Datasets. We conduct experiments on synthetic and real-world datasets. For synthetic datasets, following DIR (Wu et al. 2021), we use the SPMotif dataset to evaluate our method on structure and degree shift. For real-world datasets, we examine degree shift, size shift, and other distribution shifts. For the degree shift, we employ the Graph-SST5 and Graph-Twitter datasets (Chen et al. 2022a; Yuan et al. 2022; Dong et al. 2014; Socher et al. 2013). To evaluate size shift, we utilize PROTEINS and DD datasets from TU benchmarks (Morris et al. 2020), adhering to the data split as suggested by previous research (Chen et al. 2022a). We also consider the DrugOOD (Ji et al. 2022) and Open Graph Benchmark (OGB) (Hu et al. 2020b) for structural distribution shifts. More details are shown in the Appendix.

Evaluation. We employ different evaluation metrics tailored to specific datasets as previous works (Chen et al. 2022a; Yang et al. 2022). For the SPMotif, Graph-SST5 (Socher et al. 2013), and Graph-Twitter (Dong et al. 2014) datasets, we use accuracy as the evaluation metric. For the DrugOOD (Ji et al. 2022) and OGB (Hu et al. 2020b) datasets, we assess performance using the ROC-AUC metric. For the TU datasets (Morris et al. 2020), we measure the model with Matthews correlation coefficient. We report the mean results and standard deviations across five runs. The implementation details are given in the Appendix.

Baselines. In addition to Empirical Risk Minimization (ERM), we compare our approach with three categories of methods, including mixup-based, invariant learning based and graph OOD generalization methods. In the mixup-based methods, there are Manifold Mixup (Verma et al. 2019) and G-Mixup (Han et al. 2022). For invariant learning based cate-

Dataset	SPMotif-0.33	SPMotif-0.6	SPMotif-0.9
ERM	59.49 ± 3.50	55.48 ± 4.84	49.64 ± 4.63
G-mixup	60.31 ± 2.89	58.74 ± 5.58	53.60 ± 5.01
Manifold-mixup	58.33 ± 4.05	56.63 ± 2.96	49.81 ± 4.25
IRM	57.15 ± 3.98	61.74 ± 1.32	45.68 ± 4.88
V-REx	54.64 ± 3.05	53.60 ± 3.74	48.86 ± 9.69
EIIL	56.48 ± 2.56	60.07 ± 4.47	55.79 ± 6.54
DIR	58.73 ± 11.9	48.72 ± 14.8	41.90 ± 9.39
GSAT	56.21 ± 7.08	55.32 ± 6.35	52.11 ± 7.56
CIGA	<u>77.33 ± 9.13</u>	<u>69.29 ± 3.06</u>	<u>63.41 ± 7.38</u>
IGM	**82.36 ± 7.39**	**78.09 ± 5.63**	**76.11 ± 8.86**

Table 1: Graph classification results on synthetic datasets. We use the accuracy ACC (%) as the evaluation metric.

Shift Type	Degree		Size		Structure(Assay, Scaffold)			
Dataset	Graph-SST5	Graph-Twitter	PROTEINS	DD	$\text{DrugOOD}_{\text{Assay}}$	$\text{DrugOOD}_{\text{Scaffold}}$	BACE	BBBP
Metric	ACC (%)		MCC		AUC (%)			
ERM	43.89 ± 1.73	60.81 ± 2.05	0.22 ± 0.09	0.27 ± 0.09	76.41 ± 0.73	66.83 ± 0.93	77.83 ± 3.49	66.93 ± 2.31
G-Mixup	43.75 ± 1.34	63.91 ± 3.01	0.24 ± 0.03	0.29 ± 0.04	76.53 ± 2.20	66.01 ± 1.35	79.12 ± 2.75	68.44 ± 2.08
Manifold-Mixup	43.11 ± 0.65	62.60 ± 1.87	0.23 ± 0.04	0.28 ± 0.06	77.02 ± 1.15	65.56 ± 0.44	78.85 ± 1.26	68.67 ± 1.38
IRM	43.69 ± 1.26	63.50 ± 1.23	0.21 ± 0.09	0.22 ± 0.08	74.03 ± 0.58	66.32 ± 0.27	77.51 ± 2.46	69.13 ± 1.45
V-REx	43.28 ± 0.52	63.21 ± 1.57	0.22 ± 0.06	0.21 ± 0.07	75.85 ± 0.78	65.37 ± 0.42	76.96 ± 1.88	64.86 ± 2.13
EIIL	42.98 ± 1.03	62.76 ± 1.72	0.20 ± 0.05	0.23 ± 0.10	76.93 ± 1.44	64.13 ± 0.89	79.36 ± 2.72	65.77 ± 3.36
DIR	41.12 ± 1.96	59.85 ± 2.98	0.25 ± 0.14	0.20 ± 0.10	74.11 ± 3.10	64.45 ± 1.69	79.93 ± 2.03	69.73 ± 1.54
GSAT	43.72 ± 0.87	62.50 ± 1.44	0.21 ± 0.06	0.28 ± 0.04	76.64 ± 2.82	66.02 ± 1.13	79.63 ± 1.87	68.48 ± 2.01
CIGA	44.71 ± 1.14	64.45 ± 1.99	0.40 ± 0.06	0.29 ± 0.08	76.15 ± 1.21	67.11 ± 0.33	80.98 ± 1.25	69.65 ± 1.32
IGM	**46.69 ± 0.52**	**66.23 ± 1.58**	**0.43 ± 0.05**	**0.36 ± 0.04**	**78.16 ± 0.65**	**68.32 ± 0.48**	**82.65 ± 1.17**	**71.03 ± 0.79**

Table 2: Graph OOD generalization performance with on real-world datasets. We show the graph classification results on datasets with three types of distribution shifts: degree, size, and structure. We use ACC (%) as the evaluation metric on Graph-SST5 and Graph-Twitter datasets, MCC for DD and PROTEIN datasets, and ROC-AUC (%) for $\text{DrugOOD}_{\text{scaffold}}$, $\text{DrugOOD}_{\text{assay}}$, BACE, and BBBP datasets. Experimental results indicate that our method outperforms all the baselines.

gory, we consider the methods that use known environment partition such as Invariant Risk Minimization (IRM) (Arjovsky et al. 2019) and V-REx (Krueger et al. 2021), as well as methods that automatically partition environments, such as EIIL (Creager, Jacobsen, and Zemel 2021). The third category encompasses those information bottleneck based methods like CIGA (Chen et al. 2022a) and GSAT (Miao, Liu, and Li 2022), as well as methods based on environment divisions within the graph, such as DIR (Wu et al. 2021).

Main Results (RQ1)

Experiments on synthetic datasets. We report our results on synthetic datasets in Table 1. The bias which set as 0.33, 0.6, and 0.9 represents the degree of spurious correlation between labels and features. Through the experimental results, we can observe that our proposed method outperforms other baselines by a large margin under three different bias settings. Specifically, our model surpasses ERM by an average margin of 44.2% on average and outperforms the state-of-the-art (SOTA) method CIGA by 13.1% on average. This demonstrates that our IGM is more adept at capturing the invariant patterns under distribution shifts, thereby enabling the model to perform better on OOD data.

Experiments on real-world datasets. We explore three types of distribution shifts on real-world datasets: degree shift, size shift, and structure shift. The results are presented in Table 2. It can be observed that existing methods uniformly fail to achieve good OOD generalization across all datasets. For instance, G-Mixup underperforms compared to ERM on the Graph-SST5 and NCI109 datasets, while IRM is consistently outdone by ERM on most datasets, and the SOTA method CIGA is outperformed by ERM on DrugOOD-Scaffold dataset.

As can be observed across these eight datasets, our model consistently achieved the best performance. Our method demonstrates an overall improvement of 7.7% compared to the state-of-the-art (SOTA) methods. These results show that our model can effectively deal with the complex distribution shift in the real world. It also indicates the model's strong OOD generalization ability.

In detail, for datasets with size shift, our method achieves an average enhancement of 15.9% over SOTA. In instances involving degree shift, the average improvement stands at 3.9%. For datasets subjected to structure shift, our method records an average increase in performance of 1.7%. From these results, it can be inferred that our model excels in identifying invariant patterns in all these distribution shifts.

Ablation Study (RQ2)

We conduct two types of experiments for the ablation study. First, we explore the necessity of using two mixup methods to find invariant subgraphs. Second, we investigate the contributions of each component of the proposed IGM. For the first part, we initially utilize a previous OOD graph generalization method (CIGA, DIR) for training. We then use the subgraph extractor from the trained model as our model's invariant subgraph extractor and fix its parameters. Then we train with our two kinds of mixup. For the second part, we compare our model (two kinds of mixup) with two ablated models (only invariant Mixup and only environment Mixup) and ERM. We conduct experiments on the Graph-Twitter, DD, and BBBP datasets, obtaining results under three different distribution shifts. The results are demonstrated in Figure 2.

We can observe that the performance of using the subgraph extractor from the previous methods (CIGA, DIR) combined with our two mixup methods for training is superior to ERM but still falls short of our method. Specifically, the IGM outperformed the IGM using a pre-trained extractor by an av-

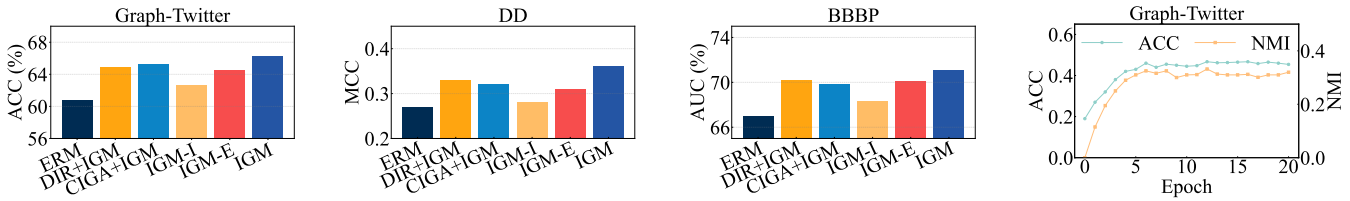


Figure 2: The first three figures present ablation studies on Graph-Twitter, DD, and BBBP datasets, emphasizing the importance of utilizing both environment and invariant Mixup. The last figure provides an analysis of environment subgraph clustering and the result of classification, demonstrating the mutual enhancement of the two mixup components.

erage of 4.6%. This validates the necessity of employing both mixup methods to obtain invariant subgraphs. Furthermore, we can observe that the performance of the two ablated models are somewhat diminished compared to simultaneously using both mixup methods, yet they still outperform ERM by 1.6% on average. This indicates that both types of mixup can achieve OOD generalization to a certain extent. Among them, the model trained only with invariant Mixup shows a more significant reduction in performance than the model using only environment Mixup.

Collaboration of Two Mixups. To show the environment Mixup module and invariant Mixup module can be mutually promoted by each other, we record the test accuracy and the Normalized Mutual Information (NMI) (Strehl and Ghosh 2002), which is a common clustering metric and can reflect the quality of generated environments for invariant learning. As shown in Figure 2, the results on Graph-Twitter demonstrate that such two metrics improve synchronously during the training process. One plausible reason is that, during environment Mixup, invariant learning across different environments captures invariant information, thereby promoting the invariant Mixup. Conversely, in the invariant Mixup phase, the invariant information captured amplifies the environment Mixup by delineating environments with larger distribution shifts, subsequently enhancing invariant learning in the environment Mixup segment.

Hyper-parameter Sensitivity Analysis (RQ3)

We conduct experiments on DrugOOD to examine our model's sensitivity to hyper-parameters. We select three critical parameters of the model, including the IRM weight γ, V-REx weight μ, invariant Mixup weight δ. We vary γ, μ and δ in $\{0.1, 0.5, 1, 2, 4\}$. The results are shown in Figure 3. Our method remains stable and effective across different values of these hyper-parameters.

Invariant Subgraph Visualization (RQ4)

To verify whether our method captures the invariant information, we first visualize the invariant subgraphs found by our model and other graph OOD methods on the Graph-Twitter, and we use this dataset because it is comprehensible to humans. It can be observed that our model adeptly identifies the specific subgraphs that are pivotal in determining the sentiment of the sentences. In contrast, DIR fails to capture all

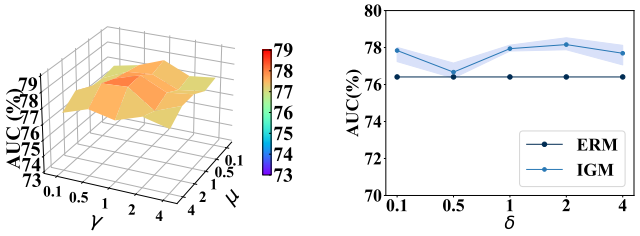
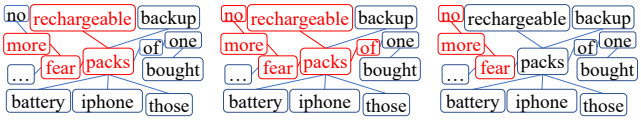


Figure 3: AUC sensitivity of hyper-parameters γ, μ, δ



(a) DIR (b) CIGA (c) IGM

Figure 4: Visualization of the invariant subgraphs extracted by different models on Graph-SST dataset. The original sentence is "Bought one of those rechargeable iPhone backup battery packs... no more fear".

the proper subgraphs, resulting in classification errors, while CIGA tends to capture relatively larger subgraphs.

Conclusion

In this work, we integrate mixup with invariant learning to address the problem of OOD generalization in graphs. We propose two modules, namely environment Mixup and invariant Mixup, to capture invariant information within graphs, thereby achieving OOD generalization. Extensive experiments demonstrate the efficacy of our methods under various distribution shifts on both synthetic and real-world datasets. In future work, we aim to extend our framework to node classification tasks and explore its applicability to dynamic graphs.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. U20B2045, 62192784, U22B2038, 62002029, 62322203, 62172052), Young Elite Scientists Sponsorship Program (No. 2023QNRC001) by CAST.

References

Ahuja, K.; Caballero, E.; Zhang, D.; Gagnon-Audet, J.-C.; Bengio, Y.; Mitliagkas, I.; and Rish, I. 2021. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34: 3438–3450.

Arjovsky, M.; Bottou, L.; Gulrajani, I.; and Lopez-Paz, D. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Bengio, Y.; Deleu, T.; Rahaman, N.; Ke, N. R.; Lachapelle, S.; Bilaniuk, O.; Goyal, A.; and Pal, C. 2020. A meta-transfer objective for learning to disentangle causal mechanisms. In *Eighth International Conference on Learning Representations*. OpenReview. net.

Bevilacqua, B.; Zhou, Y.; and Ribeiro, B. 2021. Size-invariant graph representations for graph classification extrapolations. In *International Conference on Machine Learning*, 837–851. PMLR.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral Networks and Locally Connected Networks on Graphs. In *ICLR*.

Buffelli, D.; Liò, P.; and Vandin, F. 2022. Sizeshiftreg: a regularization method for improving size-generalization in graph neural networks. *Advances in Neural Information Processing Systems*, 35: 31871–31885.

Chen, F.; Wang, Y.-C.; Wang, B.; and Kuo, C.-C. J. 2020. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing*, 9: e15.

Chen, Y.; Zhang, Y.; Bian, Y.; Yang, H.; Kaili, M.; Xie, B.; Liu, T.; Han, B.; and Cheng, J. 2022a. Learning causally invariant representations for out-of-distribution generalization on graphs. *Advances in Neural Information Processing Systems*, 35: 22131–22148.

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

Chen, Z.; Xiao, T.; and Kuang, K. 2022. Ba-gnn: On learning bias-aware graph neural network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 3012–3024. IEEE.

Chou, H.-P.; Chang, S.-C.; Pan, J.-Y.; Wei, W.; and Juan, D.-C. 2020. Remix: rebalanced mixup. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, 95–110. Springer.

Creager, E.; Jacobsen, J.-H.; and Zemel, R. 2021. Environment inference for invariant learning. In *International Conference on Machine Learning*, 2189–2200. PMLR.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *NeurIPS*, 3837–3845.

Dong, L.; Wei, F.; Tan, C.; Tang, D.; Zhou, M.; and Xu, K. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers)*, 49–54.

Guo, H.; and Mao, Y. 2021. ifmixup: Towards intrusion-free graph mixup for graph classification. *arXiv e-prints*, arXiv–2110.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017a. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017b. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Han, X.; Jiang, Z.; Liu, N.; and Hu, X. 2022. G-mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*, 8230–8248. PMLR.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020a. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33: 22118–22133.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020b. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33: 22118–22133.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

Ji, Y.; Zhang, L.; Wu, J.; Wu, B.; Huang, L.-K.; Xu, T.; Rong, Y.; Li, L.; Ren, J.; Xue, D.; et al. 2022. DrugOOD: Out-of-Distribution (OOD) Dataset Curator and Benchmark for AI-aided Drug Discovery–A Focus on Affinity Prediction Problems with Noise Annotations. *arXiv preprint arXiv:2201.09637*.

Kim, J.; Choo, W.; Jeong, H.; and Song, H. O. 2020. Co-Mixup: Saliency Guided Joint Mixup with Supermodular Diversity. In *International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Gao, I.; et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, 5637–5664. PMLR.

Krueger, D.; Caballero, E.; Jacobsen, J.-H.; Zhang, A.; Binas, J.; Zhang, D.; Le Priol, R.; and Courville, A. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, 5815–5826. PMLR.

Lee, J. B.; Rossi, R.; and Kong, X. 2018. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1666–1674.

Li, H.; Wang, X.; Zhang, Z.; and Zhu, W. 2022a. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering*.

Li, H.; Zhang, Z.; Wang, X.; and Zhu, W. 2022b. Learning invariant graph representations for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35: 11828–11841.

Liu, G.; Zhao, T.; Xu, J.; Luo, T.; and Jiang, M. 2022. Graph rationalization with environment-based augmentations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1069–1078.

Miao, S.; Liu, M.; and Li, P. 2022. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*, 15524–15543. PMLR.

Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*.

Park, J.; Shim, H.; and Yang, E. 2022. Graph transplant: Node saliency-guided graph mixup with local structure preservation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7966–7974.

Pinto, F.; Yang, H.; Lim, S. N.; Torr, P.; and Dokania, P. 2022. Using mixup as a regularizer can surprisingly improve accuracy & out-of-distribution robustness. *Advances in Neural Information Processing Systems*, 35: 14608–14622.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Strehl, A.; and Ghosh, J. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec): 583–617.

Sui, Y.; Wang, X.; Wu, J.; Lin, M.; He, X.; and Chua, T.-S. 2022. Causal attention for interpretable and generalizable graph classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1696–1705.

Vapnik, V. 1999. *The nature of statistical learning theory*. Springer science & business media.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, 6438–6447. PMLR.

Wang, Y.; Wang, W.; Liang, Y.; Cai, Y.; and Hooi, B. 2021. Mixup for node and graph classification. In *Proceedings of the Web Conference 2021*, 3663–3674.

Wu, Y.; Wang, X.; Zhang, A.; He, X.; and Chua, T.-S. 2021. Discovering Invariant Rationales for Graph Neural Networks. In *International Conference on Learning Representations*.

Xia, D.; Wang, X.; Liu, N.; and Shi, C. 2023. Learning Invariant Representations of Graph Neural Networks via Cluster Generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Yang, N.; Zeng, K.; Wu, Q.; Jia, X.; and Yan, J. 2022. Learning substructure invariance for out-of-distribution molecular representations. *Advances in Neural Information Processing Systems*, 35: 12964–12978.

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

Yoo, H.; Lee, Y.-C.; Shin, K.; and Kim, S.-W. 2023. Disentangling Degree-related Biases and Interest for Out-of-Distribution Generalized Directed Network Embedding. In *Proceedings of the ACM Web Conference 2023*, 231–239.

Yuan, H.; Yu, H.; Gui, S.; and Ji, S. 2022. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5): 5782–5799.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6023–6032.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, Z.; Wang, X.; Zhang, Z.; Li, H.; Qin, Z.; and Zhu, W. 2022. Dynamic graph neural networks under spatio-temporal distribution shift. *Advances in Neural Information Processing Systems*, 35: 6074–6089.