This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Mind the Label Shift of Augmentation-based Graph OOD Generalization

Junchi Yu[1,2], Jian Liang[1], Ran He[1,2,3*]

[1]MAIS&CRIPAC, Institute of Automation, Chinese Academy of Sciences, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, China
[3]School of Information Science and Technology, ShanghaiTech University, China
yujunchi2019@ia.ac.cn, liangjian92@gmail.com, rhe@nlpr.ia.ac.cn

Abstract

*Out-of-distribution (OOD) generalization is an important issue for Graph Neural Networks (GNNs). Recent works employ different graph editions to generate augmented environments and learn an invariant GNN for generalization. However, the label shift usually occurs in augmentation since graph structural edition inevitably alters the graph label. This brings inconsistent predictive relationships among augmented environments, which is harmful to generalization. To address this issue, we propose **LiSA**, which generates label-invariant augmentations to facilitate graph OOD generalization. Instead of resorting to graph editions, LiSA exploits **L**abel-**i**nvariant **S**ubgraphs of the training graphs to construct **A**ugmented environments. Specifically, LiSA first designs the variational subgraph generators to extract locally predictive patterns and construct multiple label-invariant subgraphs efficiently. Then, the subgraphs produced by different generators are collected to build different augmented environments. To promote diversity among augmented environments, LiSA further introduces a tractable energy-based regularization to enlarge pair-wise distances between the distributions of environments. In this manner, LiSA generates diverse augmented environments with a consistent predictive relationship and facilitates learning an invariant GNN. Extensive experiments on node-level and graph-level OOD benchmarks show that LiSA achieves impressive generalization performance with different GNN backbones. Code is available on https://github.com/Samyu0304/LiSA.*

1. Introduction

Learning from graph-structured data is a fundamental problem in various applications, such as 3D vision [50], knowledge graph reasoning [65], and social network analysis [32]. Recently, the Graph Neural Networks (GNNs) [33] have become a *de facto* standard in developing deep learning systems on graphs [10], showing superior performance on point cloud classification [18], recommendation system [56], biochemistry [29] and so on. Despite their remarkable success, these models heavily rely on the i.i.d. assumption that the training and testing data are independently drawn from an identical distribution [9, 39]. When tested on out-of-distribution (OOD) graphs (*i.e.* larger graphs), GNN usually suffers from unsatisfactory performances and unstable prediction results. Hence, handling the distribution shift for GNNs has received increasing attention.

Many solutions have been proposed to explore the OOD generalization problem in Euclidean space [47], such as invariant learning [3, 14, 38], group fairness [34], and distribution-robust optimization [49]. Recent works mainly resort to learning an invariant classifier that performs equally well in different training environments [3, 16, 37, 38]. However, the study of its counterpart problem for non-Euclidean graphs is comparatively lacking. One challenge is the environmental scarcity of graph-structured data [39, 53]. Inspired by the data augmentation literature [48, 52], some pioneering works propose to generate augmented training environments by applying different graph edition policies to the training graphs [55,57]. After training in these environments, the GNN is expected to have better OOD generalization ability. Nevertheless, the graph labels may change during the graph edition since they are sensitive to graph structural modifications. This causes the label shift problem of augmented graphs. For example, methods of graph adversarial attack usually seek to modify the graph structure to permute the model prediction [9]. Moreover, a small structural modification can drastically influence the biochemical property of molecule or protein graphs [30].

We formalize the impact of the label shift in augmentations on generalization using a unified structure equa-

*Corresponding Author

tion model [1]. Our analysis indicates that the label shift causes inconsistent predictive relationships among the augmented environments. This misguides the GNN to output a perturbed prediction rather than the invariant prediction, making the learned GNN hard to generalize (see Section 3 for more details). Thus, it is crucial to generate label-invariant augmentations for graph OOD generalization. However, designing label-invariant graph edition is nontrivial or even brings extensive computation, since it requires learning class-conditional distribution for discrete and irregular graphs. In this work, we propose a novel label-invariant subgraph augmentation method, dubbed *LiSA*, for the graph OOD generalization problem. For an input graph, LiSA first designs the variational subgraph generators to identify locally predictive patterns (*i.e.* important nodes or edges for the graph label) and generate multiple label-invariant subgraphs. These subgraphs capture prediction-relevant information with different structures, and thus construct augmented environments with a consistent predictive relationship. To promote diversity among the augmentations, we propose a tractable energy-based regularization to enlarge the pair-wise distances between the distributions of augmented environments. With the augmentations produced by LiSA, a GNN classifier is learned to be invariant across these augmented environments. The GNN predictor and variational subgraph generators are jointly optimized with a bi-level optimization scheme [61]. LiSA is model-agnostic and is flexible in handling both graph-level and node-level distribution shifts. Extensive experiments indicate that LiSA enjoys satisfactory performance gain over the baselines on 7 graph classification datasets and 4 node classification datasets. Our contributions are as follows:

- We propose a model-agnostic label-invariant subgraph augmentation (LiSA) framework to generate augmented environments with consistent predictive relationships for graph OOD generalization.
- We propose the variational subgraph generator to discover locally crucial patterns to construct the label-invariant subgraphs efficiently.
- To further promote diversity, we further propose an energy-based regularization to enlarge pair-wise distances between the distributions of different augmented environments.
- Extensive experiments on node-level and graph-level tasks indicate that LiSA enjoys satisfactory performance gain over the baselines on various backbones.

2. Related Work

Graph Neural Networks. The Graph Neural Network (GNN) has become a building block for deep graph learning [33]. It leverages the message-passing module to aggregate the adjacent information to the central node, which shows expressive power in embedding rational data. Various GNN variants have shown superior performance on social network analysis [6], recommender system [56], and biochemistry [63]. While GNNs have achieved notable success on many tasks, they rely on the i.i.d assumption that the training and testing samples are drawn independently from the same distribution [11]. This triggers concerns about the applications of GNN-based models in real-world scenarios where there is a distribution shift between the training and testing data.

Out-of-distribution (OOD) Generalization. Given the training samples from several source domains, out-of-distribution generalization aims at generalizing deep models to unseen test environments [3]. Recent studies focus on learning an invariant predictive relationship across these training environments, such as invariant representation/predictor learning [3, 37, 38], and invariant causal prediction [8, 24, 42]. They either learn a predictor that performs equally well (also known as equi-predictive [37]) in different environments or seek a stable parent variable of the label in the structural causal model (SCM) [28] for the prediction. When the environment partition is missing, many works resort to group robust optimization [26, 43, 46], environment inference [16, 64], and data augmentation [17, 52]. Although domain generalization on Euclidean data has drawn much attention, the focus on its counterpart to the graph-structured data is comparatively lacking [13]. Our work follows the data augmentation strategy. Differently, we consider generating the augmentations of graph-structured data, which is usually challenging due to their discrete and irregular structures.

OOD Generalization on Graphs. Some pioneering works [4,5] on graph OOD generalization study whether the GNN trained on small graphs can generalize to larger graph size [15]. Recently, researchers have extended OOD generalization methods, such as invariant learning [38] to handle the distribution shift on graphs [13,39]. The main challenge is environmental scarcity, which makes the learned invariant relationship insufficient for graph OOD generalization. To this end, recent works [7, 55, 57] employ different graph edition policies to generate augmented environments. Since the graph label is sensitive to the graph structure, graph editing is prone to change the label of the augmented graph and makes it difficult for graph OOD generalization

3. Graph Augmentation for Graph OOD Generalization

3.1. Problem formulation

Let $D_{tr} = \{(G_i, Y_i)|1 \leq i \leq N\}$ be the training graphs which are sampled from the distribution $p(G, Y) =$

$\sum_{e\in\mathcal{E}_{tr}} p(G,Y|e)p(e)$. Here, $G \in \mathcal{G}$ and $Y \in \mathcal{Y}$ are graphs and their labels. $e \in \mathcal{E}$ represents the environment. The goal of graph OOD generalization is to learn a graph neural network (GNN) $f : \mathcal{G} \to \mathcal{Y}$ on D_{tr}, which can generalize to unseen testing environments. This is formulated as a bi-level optimization problem, which minimizes the worst-case risk across the training environments [3]:

$$\min_{f} \mathcal{R}^e(f), s.t. e = \arg\max_{e\in\mathcal{E}_{tr}} \mathcal{R}^e(f). \tag{1}$$

Here, $\mathcal{R}^e(f)$ is the risk of f in environment e. A GNN f which minimizes Eqn. 1 is called invariant GNN, and is supposed to generalize to OOD graphs at testing time. Recent works show that the performance drop on OOD graphs is attributed to learning from spurious subgraphs, which is unstable across different environments [13, 57]. Thus, they aim to learn an invariant predictive relationship between the causal subgraph G_{inv} and the graph label Y. A GNN leveraging such a predictive relationship is stable across different environments and is supposed to generalize. However, the training environments for graphs [13, 39] are usually scarce, making it difficult to learn a generalizable GNN.

3.2. Augmentation-based Graph OOD Generalization

To address the environmental scarcity issue, some works employ different graph editing policies to change the graph structures for augmentation. For example, EERM [55] introduces a graph extrapolation strategy, which adds new edges to the training graphs with reinforcement learning. DIR [57] exchanges part of graph structures within a training batch, which is known as the graph intervention strategy. We elaborate more details on these methods in the appendix. While different augmentation strategies have been proposed, they are likely to cause the label shift in augmentations since graph labels are sensitive to structure editing. This introduces inconsistent predictive relationships among the augmented environments, and brings negative effects on graph OOD generalization. To formalize this problem, we build a unified structural equation model (SEM) for augmentation-based graph OOD generalization:

$$\begin{aligned} Y^e_{Aug} &\leftarrow I(W_{inv}\cdot G^e_{inv}) \oplus I(W_{aug}\cdot G^e_{aug}) \oplus N^e \\ G^e_{Aug} &\leftarrow S_{aug}(G^e, G^e_{aug}) \\ N^e &\sim \text{Bernoulli}(q), q < 0.5 \\ N^e &\perp (G^e, G^e_{aug}) \end{aligned} \tag{2}$$

Here, G^e_{Aug} and Y^e_{Aug} are the augmented graph and its label. S_{aug} is the augmentation function, which generates G^e_{Aug} given G^e and the augmented structure G^e_{aug}. The formulation of S_{aug} depends on the augmentation strategy. For EERM, S_{aug} represents appending G^e_{aug} to G^e. And it denotes exchanging G^e_{aug} between batched G^e for DIR. $I(\cdot)$ is the labeling function. W_{inv} is the parameterized invariant prediction relationship within original graphs and W_{aug} is the perturbed prediction relationship introduced by augmentations. W_{aug} changes the original graph label with a flipping probability p_{aug}, making $I(W_{inv}\cdot G^e_{inv}) \neq I(W_{aug}\cdot G^e_{aug})$. N_e is the independent noise within the training graphs. $\oplus$ is the XOR operation to summarize the impacts of augmentations and noise on the graph label. With the SEM model in Eqn. 2, we could compute the risk R of any classifier W following prior work [1]:

$$R = \mathrm{E}_e \mathrm{E}_{Y^e_{Aug}, G^e_{Aug}}[Y^e_{Aug} \oplus I(W\cdot G^e_{Aug})] \tag{3}$$

It is straightforward to verify that the label-invariant augmentation ($p_{aug} = 0$) can guide the GNN to leverage the invariant predictive relationship by risk minimization. However, when $p_{aug} \neq 0$, the invariant predictive relationship could be sub-optimal, making the GNN classifier to leverage the perturbed predictive relationship.

Theorem 3.1. *Denote the risk of $W = W_{inv}$ and $W = W_{aug}$ as R_{inv} and R_{aug} respectively. We have $R_{inv} \geq R_{aug}$ when $p_{aug} \in [\frac{0.5-q}{1-q}, 1]$ and $R_{inv} < R_{aug}$ when $p_{aug} \in [0, \frac{0.5-q}{1-q})$.*

The proof is in the appendix. When the label shift occurs in augmentation, a GNN classifier may fail to generalize by leveraging a perturbed predictive relationship. In this case, the OOD generalization will be unsatisfactory. Thus, it is important to maintain label-invariance in graph augmentation for OOD generalization.

4. Label-Invariant Subgraph Augmentation

In this work, we seek label-invariant augmentations to enhance the graph OOD generalization performance.

Definition 4.1 (Label-invariance Augmentation)**.** Denote $g \in G$ as the augmentation function and $f : G \to Y$ as the labeling function. g is a label-invariant augmentation function i.f.f. $f(G) = f(g(G))$.

Designing a label-invariant graph edition policy is usually difficult since it usually needs to model the class-conditioned distribution for graphs $p(G|y)$. This usually requires learning a power conditional generative model to simulate $p(G|y)$, which introduces extensive computational burden [31]. For graph-structured data, it is rather difficult to learn such a generative model due to the discrete and vast graph space [29]. To alleviate this issue, we propose the label-invariant subgraph (LiSA) augmentation method as shown in Figure 1. Instead of resorting to graph edition, LiSA efficiently generates label-invariant augmentations by exploiting label-invariant subgraphs of the training graphs.

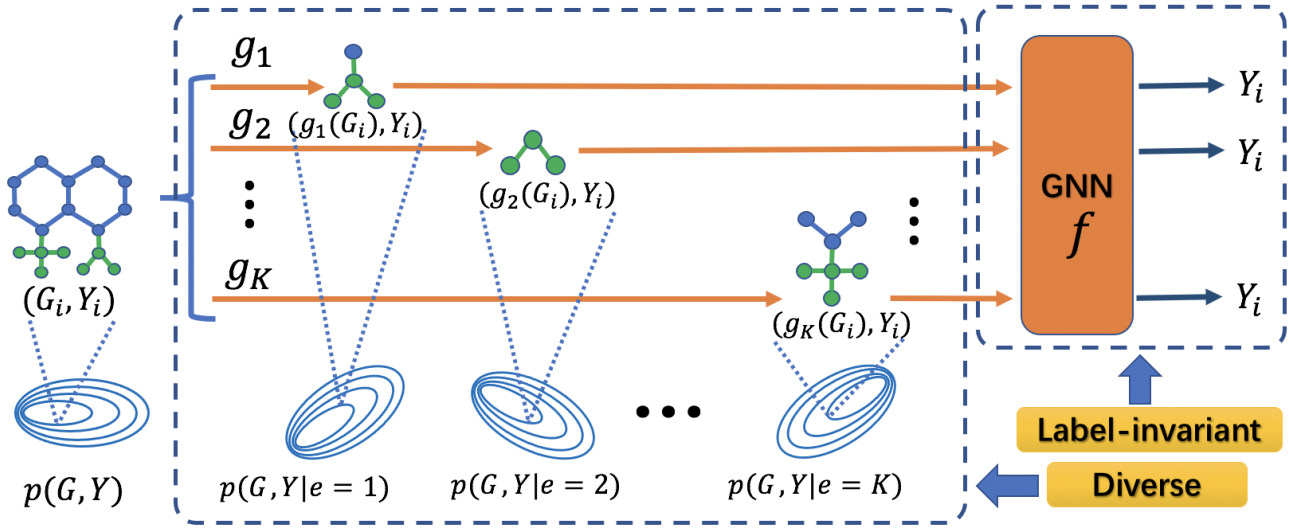

Figure 1. The whole framework of LiSA. LiSA obtains augmented environments by discovering label-invariant subgraphs with a set of variational subgraph generators $\{g_i\}_{i=1}^K$. Moreover, LiSA employs a tractable energy-based regularization to promote diversity among augmentations. With these environments, LiSA learns an invariant GNN for OOD generalization.

4.1. Discover Label-invariant Subgraph with Variational Subgraph Generator

Discovering the label-invariant subgraph of the input graph is non-trivial since the subgraph space is exponentially large [62]. We devise the variational subgraph generator to efficiently construct the label-invariant subgraph with a collection of locally predictive patterns. This factorization reduces the subgraph generation into selecting important nodes and edges to avoid directly searching in the large subgraph space.

Given an input graph $G = \{A, X\}$, the variational subgraph generator first outputs a node sampling mask to distill the structure information of G. Specifically, it employs a l-layer GNN and a Multi-Layer Perceptron (MLP) to output the sampling probability p_v of node v:

$$H = \text{GNN}(A, X),\ p_v = \text{Sigmoid}(\text{MLP}(h_v)). \quad (4)$$

Here, H is the node embedding matrix and h_v is the embedding of node v. The output of MLP is mapped into [0,1] via the Sigmoid function. A large sampling probability guides the node sampling mask $m_v = 1$ with a high probability, which indicates that the corresponding node v is important to the graph label. Since the node sampling process is non-differentiable, we further employ the concrete relaxation [19, 27] for m_v:

$$\hat{m}_v = \text{Sigmoid}(\frac{1}{t}\log\frac{p_{v,g_i}}{1-p_{v,g_i}} + \log\frac{u}{1-u}), \quad (5)$$

where t is the temperature parameter and $u \sim \text{Uniform}(0, 1)$. With the node sampling masks, we further obtain the edge sampling mask by averaging the adjacent nodes. For example, given two adjacent nodes v and n, the mask m_e for edge e_{vn} is computed as $m_e = 0.5(m_v + m_n)$. Finally, we mask the input graph G with the node and edge mask to generate the subgraph G_{sub}. By introducing the node sampling process, we decompose the subgraph generation process into the node sampling process, which greatly reduces computational expenses. We employ the information constraint [23,60] to restrict that the subgraph only contains a portion of original structural information.

$$\begin{aligned}\mathcal{L}_{info} &= \text{I}(G, G_{sub}) \\ &= \mathbb{E}_{G,G_{sub}}\log\frac{p(G_{sub}|G)}{q(G_{sub})} - \text{KL}[p(G_{sub})|q(G_{sub})] \\ &\le \mathbb{E}_{G\sim p(G)}\text{KL}[p(G_{sub}|G)|G)|q(G_{sub})].\end{aligned} \quad (6)$$

Here, KL is the KL-divergence. The inequality is due to the fact that KL-divergence is non-negative. The posterior distribution $p(G_{sub}|G, e = i)$ is factorized into $\prod_{v\in G}\text{Bernoulli}(p_v)$ due to the node sampling process. The specification of the prior $p(G_{sub}|e = i)$ in Eqn. 6 is the non-informative distribution $\prod_{v\in G}\text{Bernoulli}(0.5)$ following [2], which encodes equal probability of sampling or dropping nodes in prior knowledge.

We proceed to encode the label-invariance into the obtained subgraph with a jointly trained GNN classifier f. In each iteration, it is first updated with the labeled training graphs. Then, it serves as a proxy to measure the gap between the graph label and the subgraph label.

$$\mathcal{L}_{cls} = \text{CE}(f(g(G)), Y), \quad (7)$$

where CE is the cross-entropy loss. During the subgraph generation process, the variational subgraph generator recognizes the label-invariant subgraph by jointly minimizing

the following loss:

$$\mathcal{L} = \mathcal{L}_{cls}(f, g) + \alpha \mathcal{L}_{info}(g). \tag{8}$$

4.2. Graph OOD Generalization with LiSA

Existing works show that the performance drop in graph OOD generalization results from preferring superficial knowledge, such as spurious subgraphs, for the prediction [13, 57]. With locally easy-to-learn information, the GNN classifier can achieve a low training risk without a global understanding of the whole graph structure, making it difficult for OOD generalization. Our work alleviates this issue by first decomposing each training graph into multiple label-invariant subgraphs using a set of variational subgraph generators. Intuitively, different subgraph generators generate diverse label-invariant subgraphs to construct the augmented training environments. Suppose we use n variational subgraph generators; we can obtain $n+1$ training environments together with the original training graphs. We treat the subgraphs produced by the same variational subgraph generator as an augmented environment. And the probability density function of $n+1$ total environments is $p(G, Y) = \frac{1}{n+1} \sum_{i=1}^{n+1} p(G, Y|e_i)$ and e is the environment variable. Then, we aim to train an invariant GNN which performs equally well on these environments. In this manner, the GNN avoids from only preferring the spurious subgraph for the prediction and give stable prediction on different locally crucial patterns.

Denote the GNN classifier as f. We employ the variance regularization [38] to learn an invariant GNN:

$$\min_{f} \mathcal{L}_{cls}(f) + \mathrm{Var}_e(\mathcal{L}_{cls}(f)). \tag{9}$$

The first term is the classification loss of GNN on all the training environments and the second term is the variance of classification losses in different environments. To jointly optimize the variational subgraph generators and the GNN classifier, we minimize the loss terms in Eqn. 8 and Eqn. 9 with a bi-level optimization framework:

$$\begin{aligned} &\min_{f} \mathcal{L}_{cls}(f, g_i^*) + \mathrm{Var}_e(\mathcal{L}_{cls}(f, g_i^*)), i = 1 \sim n \\ &s.t. g_i^* = \arg\min_{g_i} \mathcal{L}_{cls}(f, g_i) + \alpha \mathcal{L}_{info}(g_i). \end{aligned} \tag{10}$$

In practice, we first obtain a sub-optimal g^* by optimizing g for T steps in the inner loop. Then, we use the updated g^* as a proxy in the outer loop to optimize f. We provide pseudo-code for optimizing Eqn. 10 in Appendix.

4.3. Enforcing Diversity Among Augmented environments

Directly optimizing Eqn. 10 may lead to a sub-optimal solution where different variational subgraph generators generate similar subgraphs. Thus, we aim to enlarge the distances between the distributions of different augmented environments to promote diversity.

Energy-based Regularization. We propose a novel energy-based diversity regularization to enlarge the distance between the underlying distributions of augmented environments. We employ the energy-based model (EBM) $p(G|e) \propto \exp -E_\theta(G|e)$ to specify the graph distribution. Here $E_\theta : \mathcal{G} \to \mathcal{R}$ is the energy score and θ is the model parameter. The energy score assigns the density of data points in each environment. Thus, we can compute the distance between the distributions of two environments based on the energy scores of pair-wised samples:

$$d(e_j, e_k) = \frac{1}{2N} \sum_{i=1}^{N} [E_{\theta_j}(G_i|e_j), E_{\theta_k}(G_i|e_k)]^2. \tag{11}$$

Directly computing the distance in Eqn. 11 requires estimating the model parameters θ_j and θ_k of EBMs in two environments, which is computationally inefficient. To this end, we compute the energy scores with the predictive logits of the GNN classifier f. Recall that the GNN classifier outputs the prediction by applying the Softmax function to the predictive logits.

$$p(Y|G, e_j) = \frac{\exp f(g_j(G))[Y]}{\sum_{Y \in \mathcal{Y}} \exp f(g_j(G))[Y]}, \tag{12}$$

where $f()[Y]$ denotes the Y-th output of f. Following prior work [21], we obtain the joint distribution $p(Y, G|e_j) = \frac{\exp f(g_j(G))[Y]}{Z}$. Here Z is the partition function. Then, we marginalize Y to obtain $p(G|e_j) = \frac{\sum_{Y \in \mathcal{Y}} \exp f(g_j(G))[Y]}{Z}$. Combining Eqn. 12, the energy score is expressed using the predictive logits:

$$E_{\theta_j}(G|e_j) = -\log \sum_{Y \in \mathcal{Y}} \exp f(g_j(G))[Y]. \tag{13}$$

Combining Eqn. 13 and Eqn. 11, we can compute pair-wise distances among environments with the energy score:

$$\mathcal{L}_e = \frac{2}{N(N+1)} \sum_{j=1}^{N} \sum_{k=j+1}^{N+1} d(e_j, e_k). \tag{14}$$

Thus, the total loss of LiSA takes the following form:

$$\begin{aligned} &\min_{f} \mathcal{L}_{cls}(f, \{g_i^*\}_{i=1}^n) + \mathrm{Var}_e(\mathcal{L}_{cls}(f, g_i^*)), i = 1 \sim n \\ &s.t. g_i^* = \arg\min_{g_i} \mathcal{L}_{cls}(f, g_i) + \alpha \mathcal{L}_{info}(g_i) + \beta \mathcal{L}_e(g_i). \end{aligned} \tag{15}$$

4.4. Extension to Node-level Tasks

We proceed to introduce the extension of LiSA on node classification tasks. Different from the graph classification

Table 1. Performances of different methods on OOD graph classification tasks. We report the mean and standard deviation of Accuracy. In the Spurious-Motif dataset, b is the indicator of spurious correlation.

Methods	SpuriousMotif				MUTAG	MNIST-75sp	DD
	0.33	0.5	0.7	0.9			
ERM	0.509 ± 0.007	0.505 ± 0.004	0.490 ± 0.006	0.448 ± 0.004	0.903 ± 0.009	0.862 ± 0.015	0.718 ± 0.027
IRM	0.502 ± 0.003	0.501 ± 0.005	0.486 ± 0.007	0.443 ± 0.017	0.910 ± 0.015	0.875 ± 0.006	0.732 ± 0.017
V-Rex	0.526 ± 0.010	0.518 ± 0.010	0.484 ± 0.010	0.452 ± 0.017	0.900 ± 0.020	0.868 ± 0.006	0.730 ± 0.031
Attention	0.514 ± 0.038	0.484 ± 0.045	0.452 ± 0.049	0.430 ± 0.016	0.917 ± 0.012	0.878 ± 0.003	0.529 ± 0.053
TopKPool	0.439 ± 0.028	0.432 ± 0.038	0.482 ± 0.035	0.366 ± 0.006	0.913 ± 0.007	$\mathbf{0.879 \pm 0.003}$	0.663 ± 0.031
GIB	0.524 ± 0.024	0.492 ± 0.019	0.430 ± 0.062	0.355 ± 0.003	0.887 ± 0.053	0.865 ± 0.002	0.543 ± 0.178
DIR	0.468 ± 0.025	0.459 ± 0.030	0.427 ± 0.021	0.386 ± 0.011	0.895 ± 0.049	0.812 ± 0.031	0.741 ± 0.074
LiSA	$\mathbf{0.530 \pm 0.004}$	$\mathbf{0.529 \pm 0.003}$	$\mathbf{0.501 \pm 0.005}$	$\mathbf{0.474 \pm 0.009}$	$\mathbf{0.937 \pm 0.014}$	0.876 ± 0.008	$\mathbf{0.746 \pm 0.069}$

task, the nodes are associated with their neighborhoods in the node classification task. Hence, we take a local view of the nodes and relate them with 1-hop ego-graphs [55, 66]. For example, N_i is associated with $G_i = (A_i, X_i)$, where A_i is the adjacent matrix of the 1-hop subgraph centered at N_i and X_i is the neighborhood node feature matrix. Then, we generate multiple label-invariant subgraphs of G_i by optimizing the subgraph generators with Eqn. 8. The whole framework of LiSA is optimized with Eqn. 15.

5. Experiments

In this section, we extensively evaluate LiSA on both node-level and graph-level OOD generalization tasks with different types of distribution shifts. We run experiments on the server with Tesla V100 GPU and Intel(R) Xeon(R) Gold 6348 CPU, and use the PyG for implementation. The network architecture, sensitivity study of hyper-parameters, and detailed information on datasets are in the appendix.

5.1. Graph-level OOD Generalization

We first evaluate LiSA on out-of-distribution (OOD) graph classification tasks with various distribution shifts such as the graph size, noise feature, and spurious motif.

Datasets. We employ **Spurious-Motif** [59], **MUTAG** [44], **D&D** [36], and **MNIST-75sp** [36] datasets for OOD graph classification. The Spurious-Motif dataset consists of synthetic graphs with spurious motifs. Each graph is generated by attaching one base (Tree, Ladder, Wheel, denoted as $S = 0, 1, 2$) to a motif (Cycle, House, Crane, denoted as $C = 0, 1, 2$). The graph label Y is consistent with the class of motif. For the training graphs, the base is chosen with probability $P(S) = b \times \mathbb{1}(S = C) + \frac{1-b}{2} \times \mathbb{1}(S \neq C)$ to create a spurious correlation. b is changed to impose different biases on the training graphs. For testing graphs, the motifs and bases are randomly connected. The training and testing data in D&D and MUTAG datasets vary in the graph size. Specifically, we choose the graphs in the D&D dataset with less than 200 nodes for training, those with 200-300 nodes for validation, and graphs larger than 300 nodes for testing. For MUTAG, we select graphs with less than 15 nodes for training, those with 15-20 nodes for validation, and graphs larger than 20 nodes for testing. For MNIST-75sp, each image is converted as super-pixel graphs. The features of testing data contain random noises. We report accuracy (Acc) for these datasets.

Baselines. We compare our method with empirical risk minimization (ERM), invariant learning methods, including V-Rex [38] and IRM [3]; interpretable methods, such as Attention-based Pooling [36], TopK-Pooling [20], and GIB [61]; and augmentation-based invariant learning method DIR [57]. We employ GIN [58] as the backbone for model-agnostic baselines. Since there is only one domain for training, we randomly group graphs to mimic different domains to instantiate V-Rex and IRM. We report the mean and standard deviation of testing performances in 10 runs for different methods.

Performance. We report graph OOD classification performance in Table 1. LiSA outperforms most baseline methods on both synthetic and real-world datasets, with up to 5% absolute performance gain. When compared with ERM, IRM and V-Rex only achieve comparable performance on most datasets. This shows that it is usually insufficient to directly implement general OOD generalization methods to handle the distribution shifts on graphs. For interpretable methods, they somehow achieve performance gain compared with ERM as they discover a prediction-relevant subgraph for predictions. However, they underperform other methods when a large distribution shift occurs in the dataset, such as $b = 0.9$ for Spurious Motif and DD. Moreover, we find that DIR, an augmentation-based invariant learning method for graphs, can sometimes underperform ERM or interpretable methods due to the label shift problem in augmentation. Thus, it is important to maintain label invariance during augmentation.

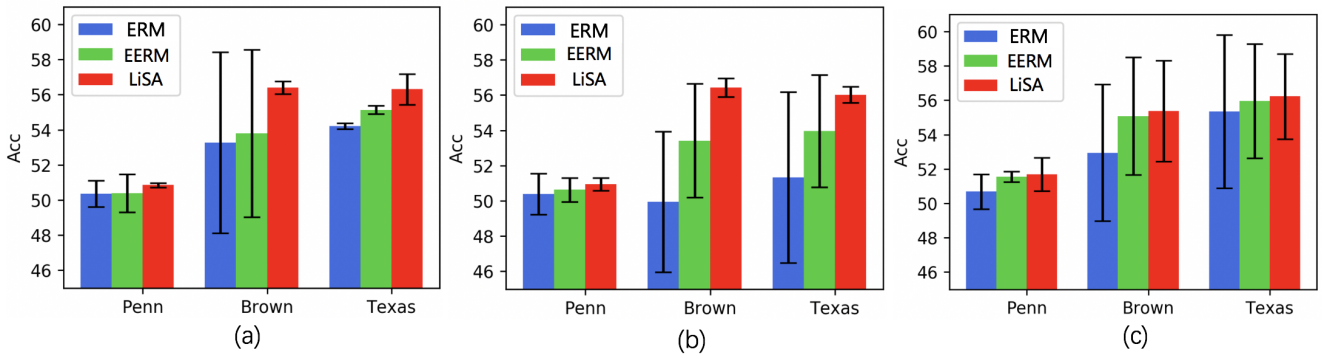

Figure 2. We report performances of different methods with 3 source domain combinations on the Facebook-100 dataset: (a). John Hopkins + Caltech + Amherst; (b).Bingham + Duke + Princeton; and (c). WashU + Brandeis+ Carnegie. We report the mean and standard deviation of Accuracy across different runs.

Table 2. Mean and standard deviation of Accuracy (Acc) on OGB-Arxiv dataset.

Test Domain	14-16		16-18		18-20	
Backbone	APPNP	SGGCN	APPNP	SGGCN	APPNP	SGGCN
ERM	46.30 ± 0.35	40.52 ± 1.24	43.75 ± 0.40	38.23 ± 2.15	39.78 ± 0.41	34.62 ± 2.14
EERM	46.42 ± 0.46	42.37 ± 2.37	44.53 ± 0.54	39.91 ± 2.07	**43.24 ± 0.79**	37.73 ± 1.42
LiSA	**47.50 ± 0.52**	**47.14 ± 0.34**	**45.10 ± 0.50**	**45.49 ± 0.37**	41.216 ± 0.249	**38.89 ± 0.71**

5.2. Node-level OOD Generalization

We proceed to apply LiSA to the OOD node classification, where the distribution shifts are spatial and temporal.

Datasets & Metrics. For the spatial shift, we adopt **Twitch-Explicit** [45] and **Facebook-100** [51] datasets for evaluation. These datasets contain different social networks which are related to different locations such as campuses and districts. For example, Twitch-Explicit contains seven social networks, including DE, ENGB, ES, FR, PTBR, RU, and TW. Following the protocol in prior work [55], we employ DE for training, ENGB for validation, and the rest five networks for testing. For the Facebook-100 dataset, we choose different combinations of three graphs for training, two for validation, and the rest three graphs for testing. We report ROC-AUC and Accuracy (Acc) for Twitch-Explicit and Facebook-100.

For the temporal shift, we use a citation network **OGB-Arxiv** [25], and a dynamic financial dataset **ELLIPTIC** [41]. For OGB-Arxiv, we employ the papers published before 2011 for training, from 2011∼2014 for validation, and those within 2014∼2016/2016∼2018/2018∼2020 for testing. For ELLIPTIC, we split the whole dataset into different snapshots, and use 5/5/33 for training, validation, and testing. The testing environments are further chronologically clustered into 9 folders for the convenience of comparing the performances of different methods. We report Test F1 Score and Accuracy (Acc) for ELLIPTIC and OGB-Arxiv.

Baselines. We compare the performance of the proposed LiSA with **ERM** and the state-of-the-art node generalization method, Explore-to-Extrapolate Risk Minimization (**EERM**) [55]. EERM generates augmentations by adding new edges while LiSA generates diverse label-invariant subgraphs. For a fair comparison, we generate 3 augmented domains for both EERM and LiSA. We further plug different methods into various GNN backbones, such as GCN [33], GraphSAGE [22], APPNP [35], SGGCN [54] and GCNII [12], to extensively evaluate their performance. We evaluate the model with the highest validation accuracy and report the mean and standard deviation of 10-run performance for each method.

Performance. We report the results on Twitch-Explicit in Table 3. The proposed LiSA exceeds the baselines in most testing environments. Since there is only one source domain for training, ERM is difficult to generalize. EERM outperforms ERM in 4 of 5 testing environments. In Figure 2, we compare different methods with the GCN backbone on the Facebook-100 dataset. We can see that LiSA achieves performance gains in different training environments. Moreover, the performance variance of LiSA is low, showing a more stable generalization performance compared with EERM and ERM.

For the temporal shift on nodes, we first plug different methods into APPNP and SGGCN backbones and evaluate their performances on the OGB-Arxiv dataset. As shown in Table 2, LiSA outperforms the baselines in five cases out of six with stable results in different runs. Then, we report the results on the Elliptic dataset in Figure 3. LiSA outperforms the baseline methods in most testing folds with different backbones and achieves up to 10% absolute performance gain. Moreover, we observe using different backbones can lead to different generalization performances. Nevertheless, LiSA still achieves better generalization performances when using the same backbone as the baselines.

Table 3. Test ROC-AUC on Twitch-Explicit dataset. Each method is trained in a single environment. For a fair comparison, both EERM and LiSA generate 3 augmented environments.

GCN	ES	FR	PTBR	RU	TW
ERM	52.50 ± 4.09	54.92 ± 2.60	48.78 ± 7.45	50.49 ± 1.82	48.95 ± 2.31
EERM	54.17 ± 5.04	54.10 ± 1.76	49.49 ± 7.96	51.34 ± 1.672	49.83 ± 3.15
LiSA-Rex	57.75 ± 3.75	53.77 ± 0.84	55.40 ± 9.04	52.47 ± 0.39	**54.66 ± 0.53**
LiSA	**57.97 ± 2.96**	**55.87 ± 2.66**	**59.96 ± 2.12**	**52.73 ± 0.67**	52.60 ± 2.64
LiSA w/o $\mathcal{L}_e$	57.28 ± 3.49	54.80 ± 1.37	57.73 ± 7.23	52.55 ± 0.93	52.67 ± 2.21
LiSA w/o $\mathcal{L}_{info}$	55.81 ± 2.21	54.94 ± 2.49	57.49 ± 2.17	51.76 ± 0.91	50.71 ± 2.47

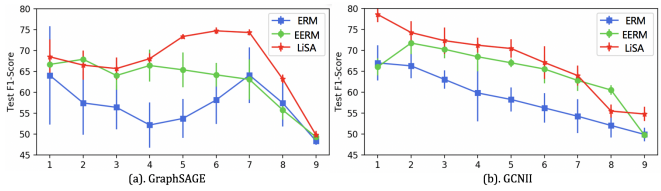

Figure 3. Test F1-Score on 9 testing folds in the ELLIPTIC dataset. LiSA achieves better OOD generalization performance with different GNN backbones.

5.3. Discussions

Influence of Backbone on OOD Generalization. The OOD generalization performance on graphs is sensitive to GNN backbones. As shown in Table 2, using APPNP as the backbone usually results in better generalization performance for the node-level temporal shift. Moreover, the generalization performances in Figure 3 behave differently when choosing different backbones. Thus, apart from label-invariant augmentation, adopting an appropriate GNN backbone is also essential for graph OOD generalization.

Ablation Study. We remove $\mathcal{L}_{info}$ and $\mathcal{L}_e$ from Eqn. 15 to study their effects on label-invariant augmentation. As shown in Table 3, removing these terms leads to performance drops in generalization, which validates that these two terms are important to generate label-invariant subgraphs for augmentation. Notice that prior work [55] maximizes the variance of classifier in the augmented environments to promote diversity. We replace $\mathcal{L}_e$ in Eqn. 15 with the variance-based regularization, leading to LiSA-Rex. As shown in Table 3, LiSA-Rex also achieves competitive performance. LiSA-Rex outperforms EERM when they both employ variance-based regularization for diversity. Thus, it is important to maintain label-invariant augmentation for graph OOD generalization. Moreover, using the proposed energy-based regularization leads to better OOD generalization performance than variance-based regularization since LiSA outperforms LiSA-Rex.

On the Diversity of Augmentations. Moreover, we study the augmentation performances of different methods. We compare the distance between the original training environment and 3 augmented environments, denoted as $d_1 \sim d_3$, and the average pair-wise distance across 3 augmented environments, denoted as d_{intra}. We employ the score-based distance in OOD detection [40] as the distance metric. As shown in Table 4, the augmented environments generated by EERM are similar since they have a small average pair-wise distance, which shows insufficient diversity. LiSA produces augmented environments with a large average pair-wise distance. Moreover, the augmented domains are also far from the source domain. Hence, LiSA can indeed generate more diverse augmentations to facilitate graph OOD generalization.

Table 4. On the diversity of different augmentation methods. $d_1 \sim d_3$ are distances between the original training environment and the augmented environments. d_{intra} is the average pair-wise distance across augmented environments.

Distance	d_1	d_2	d_3	d_{intra}
EERM	0.76	0.73	0.75	0.04
LiSA	0.67	0.70	0.64	0.52

6. Conclusion

In this work, we have studied augmentation-based OOD generalization for graphs. We show that the label shift during augmentation makes the learned GNN hard to generalize, and thus it is crucial to maintain label-invariant augmentation. We propose LiSA to efficiently generate label-invariant augmentations by exploiting multiple label-invariant subgraphs of the training graphs. LiSA contains a set of variational subgraph generators to discover label-invariant subgraphs efficiently. Moreover, a novel energy-based regularization is proposed to promote diversity among augmentations. With these augmentations, LiSA can learn an invariant GNN that is expected to generalize. LiSA is model-agnostic and can be plugged into various GNN backbones. Extensive experiments on node-level and graph-level benchmarks show the superior performance of LiSA on various graph OOD generalization tasks.

Acknowledgement

This work is partially funded by the National Natural Science Foundation of China (Grant No. U21B2045), the National Natural Science Foundation of China (Grant No. U20A20223), the Youth Innovation Promotion Association CAS (Grant No. Y201929), and Beijing Nova Program under (Grant Z211100002121108).

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, 3

[2] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016. 4

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

[4] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. *arXiv preprint arXiv:2102.06966*, 2021. 2

[5] 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. 2

[6] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 549–556, 2020. 2

[7] Davide Buffelli, Pietro Liò, and Fabio Vandin. Sizeshiftreg: a regularization method for improving size-generalization in graph neural networks. *arXiv preprint arXiv:2207.07888*, 2022. 2

[8] Peter Bühlmann. Invariance, causality and robustness. *Statistical Science*, 35(3):404–426, 2020. 2

[9] Heng Chang, Yu Rong, Tingyang Xu, Yatao Bian, Shiji Zhou, Xin Wang, Junzhou Huang, and Wenwu Zhu. Not all low-pass filters are robust in graph convolutional networks. *Advances in Neural Information Processing Systems*, 34:25058–25071, 2021. 1

[10] Heng Chang, Yu Rong, Tingyang Xu, Wenbing Huang, Honglei Zhang, Peng Cui, Xin Wang, Wenwu Zhu, and Junzhou Huang. Adversarial attack framework on graph embedding models with limited knowledge. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2022. 1

[11] Heng Chang, Yu Rong, Tingyang Xu, Wenbing Huang, Honglei Zhang, Peng Cui, Wenwu Zhu, and Junzhou Huang. A restricted black-box adversarial framework towards attacking graph embedding models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3389–3396, 2020. 2

[12] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pages 1725–1735. PMLR, 2020. 7

[13] Yongqiang Chen, Yonggang Zhang, Han Yang, Kaili Ma, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Invariance principle meets out-of-distribution generalization on graphs. *arXiv preprint arXiv:2202.05441*, 2022. 2, 3, 5

[14] Yongqiang Chen, Kaiwen Zhou, Yatao Bian, Binghui Xie, Bingzhe Wu, Yonggang Zhang, MA KAILI, Han Yang, Peilin Zhao, Bo Han, and James Cheng. Pareto invariant risk minimization: Towards mitigating the optimization dilemma in out-of-distribution generalization. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[15] Ching-Yao Chuang and Stefanie Jegelka. Tree mover's distance: Bridging graph metrics and stability of graph neural networks. *arXiv preprint arXiv:2210.01906*, 2022. 2

[16] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021. 1, 2

[17] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 2

[18] Kai Fischer, Martin Simon, Florian Olsner, Stefan Milz, Horst-Michael Gross, and Patrick Mader. Stickypillars: Robust and efficient feature matching on point clouds using graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 313–323, 2021. 1

[19] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. *arXiv preprint arXiv:1705.07832*, 2017. 4

[20] Hongyang Gao and Shuiwang Ji. Graph u-nets. In *international conference on machine learning*, pages 2083–2092. PMLR, 2019. 6

[21] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019. 5

[22] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017. 7

[23] Ran He, Bao-Gang Hu, and Xiao-Tong Yuan. Robust discriminant analysis based on nonparametric maximum entropy. In *Advances in Machine Learning: First Asian Conference on Machine Learning, ACML 2009, Nanjing, China, November 2-4, 2009. Proceedings 1*, pages 120–134. Springer, 2009. 4

[24] Christina Heinze-Deml, Jonas Peters, and Nicolai Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 2018. 2

[25] 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. 7

[26] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018. 2

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

[28] Yibo Jiang and Victor Veitch. Invariant and transportable representations for anti-causal domain shifts. *arXiv preprint arXiv:2207.01603*, 2022. 2

[29] 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, 3

[30] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures. In *International Conference on Machine Learning*, pages 4849–4859. PMLR, 2020. 1

[31] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 66–74, 2020. 3

[32] Kazi Zainab Khanam, Gautam Srivastava, and Vijay Mago. The homophily principle in social network analysis: A survey. *Multimedia Tools and Applications*, pages 1–44, 2022. 1

[33] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *The International Conference on Representation Learning*, 2017. 1, 2, 7

[34] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Algorithmic fairness. In *Aea papers and proceedings*, volume 108, pages 22–27, 2018. 1

[35] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018. 7

[36] Boris Knyazev, Graham W. Taylor, and Mohamed R. Amer. Understanding attention and generalization in graph neural networks. In *NeurIPS*, pages 4204–4214, 2019. 6

[37] Masanori Koyama and Shoichiro Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. 2020. 1, 2

[38] 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. 1, 2, 5, 6

[39] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *arXiv preprint arXiv:2202.07987*, 2022. 1, 2, 3

[40] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020. 8

[41] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5363–5370, 2020. 7

[42] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016. 2

[43] Qi Qian, Shenghuo Zhu, Jiasheng Tang, Rong Jin, Baigui Sun, and Hao Li. Robust optimization over multiple domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4739–4746, 2019. 2

[44] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. 6

[45] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021. 7

[46] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 2

[47] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021. 1

[48] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. 1

[49] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017. 1

[50] Shyam A Tailor, René de Jong, Tiago Azevedo, Matthew Mattina, and Partha Maji. Towards efficient point cloud graph neural networks through architectural simplification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2095–2104, 2021. 1

[51] Amanda L Traud, Peter J Mucha, and Mason A Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012. 7

[52] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018. 1, 2

[53] Bingzhe Wu, Jintang Li, Junchi Yu, Yatao Bian, Hengtong Zhang, CHaochao Chen, Chengbin Hou, Guoji Fu, Liang Chen, Tingyang Xu, et al. A survey of trustworthy graph learning: Reliability, explainability, and privacy protection. *arXiv preprint arXiv:2205.10014*, 2022. 1

[54] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019. 7

[55] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. *arXiv preprint arXiv:2202.02466*, 2022. 1, 2, 3, 6, 7, 8

[56] Shiwen Wu, Fei Sun, Wentao Zhang, and Bin Cui. Graph neural networks in recommender systems: a survey. *arXiv preprint arXiv:2011.02260*, 2020. 1, 2

[57] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. *International Conference on Learning Representations*, 2022. 1, 2, 3, 5, 6

[58] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *Proceedings of the 7th International Conference on Learning Representations*, ICLR '19, pages 1–17, 2019. 6

[59] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in neural information processing systems*, 2019. 6

[60] Junchi Yu, Jie Cao, and Ran He. Improving subgraph recognition with variational graph information bottleneck. *IEEE Conferences on Computer Vision and Pattern Recognition*, 2022. 4

[61] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. Graph information bottleneck for subgraph recognition. *International Conference on Learning Representations*, 2021. 2, 6

[62] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. Recognizing predictive substructures with subgraph information bottleneck. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 2021. 4

[63] Junchi Yu, Tingyang Xu, Yu Rong, Junzhou Huang, and Ran He. Structure-aware conditional variational auto-encoder for constrained molecule optimization. *Pattern Recognition*, 126:108581, 2022. 2

[64] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Ré. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. *arXiv preprint arXiv:2203.01517*, 2022. 2

[65] Mengxi Zhou, Wei Xu, Wenping Zhang, and Qiqi Jiang. Leverage knowledge graph and gcn for fine-grained-level clickbait detection. *World Wide Web*, 25(3):1243–1258, 2022. 1

[66] Qi Zhu, Carl Yang, Yidan Xu, Haonan Wang, Chao Zhang, and Jiawei Han. Transfer learning of graph neural networks with ego-graph information maximization. *Advances in Neural Information Processing Systems*, 34, 2021. 6