Out-Of-Distribution Generalization on Graphs: A Survey

Haoyang Li, Xin Wang, *Member, IEEE,* Ziwei Zhang, *Member, IEEE,* Wenwu Zhu, *Fellow, IEEE*

Abstract—Graph machine learning has been extensively studied in both academia and industry. Although booming with a vast number of emerging methods and techniques, most of the literature is built on the in-distribution hypothesis, i.e., testing and training graph data are identically distributed. However, this in-distribution hypothesis can hardly be satisfied in many real-world graph scenarios where the model performance substantially degrades when there exist distribution shifts between testing and training graph data. To solve this critical problem, out-of-distribution (OOD) generalization on graphs, which goes beyond the in-distribution hypothesis, has made great progress and attracted ever-increasing attention from the research community. In this paper, we comprehensively survey OOD generalization on graphs and present a detailed review of recent advances in this area. First, we provide a formal problem definition of OOD generalization on graphs. Second, we categorize existing methods into three classes from conceptually different perspectives, i.e., data, model, and learning strategy, based on their positions in the graph machine learning pipeline, followed by detailed discussions for each category. We also review the theories related to OOD generalization on graphs and introduce the commonly used graph datasets for thorough evaluations. Finally, we share our insights on future research directions.

Index Terms—Graph Machine Learning, Graph Neural Network, Out-Of-Distribution Generalization.

1 Introduction

Graph data is ubiquitous in our daily life. It has been widely used to model the complex relationships and dependencies between entities, ranging from microscopic particle interactions in physical systems and molecular structures in proteins to macroscopic traffic networks and global communication networks. Machine learning approaches on graphs, especially for graph neural networks (GNNs), have attracted wide attention and been extensively studied in the last decade. They have shown great successes in both academia and industry, illustrating their excellent capabilities in a wide range of realistic applications, e.g., social networks [1], recommendation systems [2], knowledge representation [3], traffic forecasting [4], etc.

Despite the notable success of graph machine learning approaches, the existing literature generally relies on the assumption that the testing and training graph data are drawn from the identical distribution, i.e., the in-distribution (I.D.) hypothesis. However, in the real world, such a hypothesis is difficult to be satisfied due to the uncontrollable underlying data generation mechanism [5]. In practice, there will inevitably be scenarios with distribution shifts between testing and training graphs [6]. These classic graph machine learning approaches lack the ability of *out-of-distribution* (OOD) generalization, which fail dramatically with significant performance drop under distribution shifts. Therefore, it is of paramount importance to develop approaches capable of out-of-distribution generalization on graphs, especially for high-stake graph applications, e.g., molecule prediction [7], financial analysis [8], criminal justice [9], autonomous driving [10], particle physics [11], as well as pandemic prediction [12], medical detection [13] and drug repurposing [14] for COVID-19.

Out-of-distribution (OOD) generalization algorithm [15–17] aims to achieve satisfactory generalization performance under unknown distribution shifts. It has been occupying an important position in the research community due to the increasing demand for handling in-the-wild unseen data. Combining the strength of graph machine learning and OOD generalization, i.e., **OOD generalization on graphs**, naturally serves as a promising research direction to facilitate graph machine learning model deployments in real-world scenarios. However, this problem is highly non-trivial due to the following challenges.

- **Uniqueness of graph data**: The non-Euclidean nature of graph-structured data space leads the unique graph model designs and makes obstacles for the direct adoption of OOD generalization algorithms that are mainly developed on Euclidean data (e.g., images and texts).
- **Diversity of graph task**: The problems on graphs are highly diverse, ranging from node-level, link-level to graph-level tasks, along with distinct settings, objectives, and constraints. It is necessary to integrate different levels of graph characterizations into the graph OOD generalization methods.
- **Complexity of graph distribution shift type**: The distribution shifts on graphs can exist on feature-level (e.g., node features) and topology-level (e.g., graph size or other structural properties). Such complex types of graph distribution shifts (as shown in Fig. 1) render more difficulties for OOD generalization.

With both opportunities and challenges, it is the right time to review and carry out the studies of graph OOD generalization methods. In this paper, we provide a systematic and comprehen-

- *Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu are with the Department of Computer Science and Technology in Tsinghua University, Beijing, China. Xin Wang and Wenwu Zhu are also with BNRist, Tsinghua University. Haoyang Li and Xin Wang contribute equally. Corresponding authors: Xin Wang and Wenwu Zhu. E-mail: lihy18@mails.tsinghua.edu.cn, {xin_wang, zwzhang, wwzhu}@tsinghua.edu.cn.*
- *This work was supported by the National Key Research and Development Program of China No.2023YFF1205001, National Natural Science Foundation of China No. 62222209, Beijing National Research Center for Information Science and Technology under Grant No. BNR2023TD03006, and Beijing Key Lab of Networked Multimedia.*

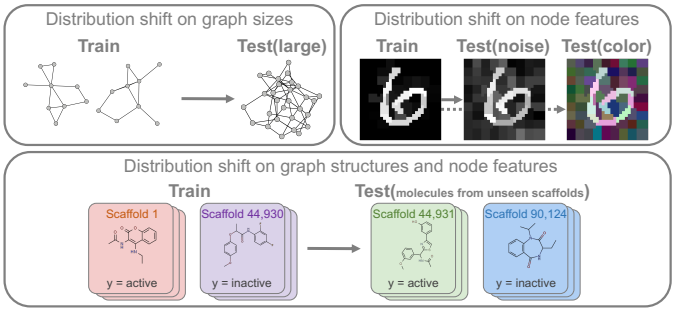

Fig. 1: Complex types of distribution shifts on graphs. The distribution shifts can exist on graph sizes, node features, and graph structural properties [6]. The OOD generalized graph approaches are expected to perform well on the unseen testing data even under distribution shifts rather than overfitting the training data.

sive review[1] for OOD generalization on graphs for the first time, to the best of our knowledge. Specifically, to cover the whole life cycle of OOD generalization on graphs, we start by providing a formal problem definition. We divide the existing methodologies into three conceptually different categories based on their positions in the graph machine learning pipeline, and elaborate typical approaches for each category. We also review the theories and datasets for evaluations to further promote the research on OOD generalization on graphs. Last but not least, we share our insights on potential research topics deserving future investigations.

Some related surveys review from the perspectives of graph data augmentation [18, 19], graph self-supervised learning [20, 21], graph adversarial learning [22, 23], etc. However, they are significantly different from ours. First, they do not focus on the graph OOD generalization that is the center topic of this survey. Then, a portion of their reviewed methods serves as an important piece of the puzzle for the whole problem of graph OOD generalization. To the best of our knowledge, there is no comprehensive review for current advancements of graph OOD generalization methods.

The rest of the paper is organized as follows. In Section 2, we formulate the problem of OOD generalization on graphs and present our categorization of existing literature. We comprehensively review three categories of methods in Sections 3–5, followed by our review of related theory (in Section 6) and evaluation datasets (in Section 7). Lastly, we point out future research opportunities in Section 8.

2 Problem Definition and Categorization

In this section, we first describe the formulation of OOD generalization on graphs. Then we provide the categorization of existing graph OOD generalization methods.

2.1 Problem Definition

Let $G = (V, E)$ denote a graph, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. $X \in \mathbb{R}^{|V| \times F}$ denotes node feature matrix where F is the dimensionality of node feature. A denotes the adjacency matrix reflecting the topological structure. Therefore, the graph G can be composed of the node feature and topological structure, i.e., $G = (X, A)$.

1. The summary of graph OOD generalization methods reviewed in this survey can be found at https://graph.ood-generalization.com.

Let $\mathbb{G}$ be the input graph space and $\mathbb{Y}$ be the label space. A graph predictor $f_\theta : \mathbb{G} \to \mathbb{Y}$ with parameter θ maps the input instance $G \in \mathbb{G}$ into the label $Y \in \mathbb{Y}$. A loss function ℓ measures the distance between prediction and ground-truth label. The graph OOD generalization problem is defined as:

Definition 1 (Graph OOD generalization)**.** *Given the training set of N instances (i.e., nodes, links, or graphs) $\mathcal{D} = \{(G_i, Y_i)\}_{i=1}^{N}$ that are drawn from training distribution $P_{train}(G, Y)$, where $G_i \in \mathbb{G}$ and $Y_i \in \mathbb{Y}$, the goal is to learn an optimal graph predictor f_θ^* that can achieve the best generalization on the data drawn from test distribution $P_{test}(G, Y)$, where $P_{test}(G, Y) \neq P_{train}(G, Y)$:*

$$f_\theta^* = \arg\min_{f_\theta} \mathbb{E}_{G,Y \sim P_{test}}[\ell(f_\theta(G), Y)]. \tag{1}$$

The distribution shifts between $P_{test}(G, Y)$ and $P_{train}(G, Y)$ can lead to the failure of graph predictor built on the in-distribution (I.D.) hypothesis, since directly minimizing the average loss on training instances $\mathbb{E}_{G,Y \sim P_{train}}[\ell(f_\theta(G), Y)]$ can not obtain an optimal predictor that generalizes to testing instances under distribution shifts. Note that the testing distribution is unknown during the training stage. Compared to traditional domain generalization problems [86], graph OOD generalization is inherently more complex, as it requires addressing potential multi-level distribution shifts, including those at the feature level (e.g., node features X) and topology level (e.g., graph size, structural patterns A). These shifts may occur independently or simultaneously, posing significant challenges to learning an optimal graph predictor f_θ^* that can generalize effectively within and even across diverse tasks, such as node-level, link-level, and graph-level predictions.

2.2 Categorization

To tackle the challenges brought by unknown distribution shifts and solve the graph OOD generalization problem, considerable efforts have been made in literature, which can be categorized into three classes:

- **Data**: This category of methods aims to manipulate the input graph data, i.e., graph augmentation. They are typically motivated by the view that OOD generalization failure is often induced by limited diversity or coverage in the training data. By systematically generating more training samples to increase the quantity and diversity of the training set while generally keeping the model backbone unchanged, graph augmentation techniques are effective in improving the OOD generalization performance.
- **Model**: This category of methods aims to propose new graph models for learning OOD generalized graph representations, including two types of representative methods: disentanglement-based graph models and causality-based graph models. They aim to improve OOD generalization directly into the design of graph neural networks with specific prior knowledge or causal assumptions. They are designed to separate causal from spurious factors through structural inductive biases, typically operating at the level of graph representations output by the graph model. Their contributions or claims are mainly in the new model architectural design for handling distribution shifts, although in principle these methods could potentially be combined with graph augmentations or customized training objectives.

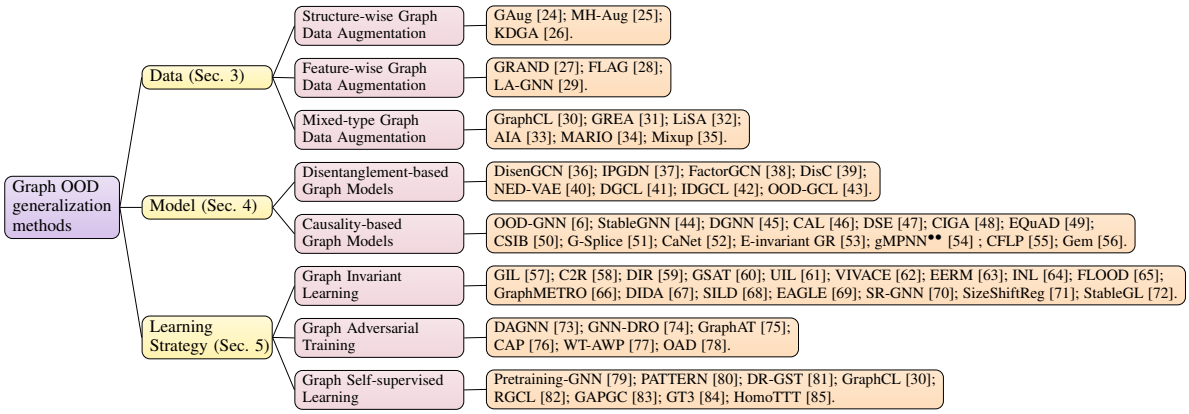

Fig. 2: Taxonomy of graph OOD generalization methods. We categorize existing methodologies into three conceptually different branches based on their positions in the graph machine learning pipeline, i.e., data, model and learning strategy.

TABLE 1: Conceptual relations and distinctions between the three categories of graph OOD generalization methods.

Aspect	Data	Model	Learning Strategy
Goal	Increase data diversity and quality	Encode prior knowledge or causal assumptions into model design	Enhance generalization via tailored training objectives and strategies
Targeted Component	Input graph structure or feature	Model architecture design or representations	Training procedures, loss functions or optimization schemes
Predominant Phase of Application	Primarily during data preparation stage	Typically during model design and representation learning	Mainly during optimization or training
GNN Backbone	Typically keep GNN backbone unchanged	Generally have specific GNN backbone design	Generally are compatible with different GNN backbones
Theoretical Foundation	Graph data augmentation	Representation disentanglement, causal modeling	Invariant learning, adversarial training, self-supervised learning
Typical Tools	Graph perturbation, mixup, graphon interpolation	Disentangled encoders, do-calculus, sample reweighting	Invariance loss, contrastive loss, adversarial training

- **Learning Strategy**: This category of methods focuses on exploiting the training schemes with tailored optimization objectives and constraints to enhance the OOD generalization capability, including graph invariant learning, graph adversarial training, and graph self-supervised learning. They typically retain the original data and generally do not rely on specific new model architectures but instead are often compatible with various GNN backbones to enhance OOD generalization through guiding the learning process.

These methods solve the graph OOD generalization problem from three conceptually different perspectives. We provide the taxonomy in Figure 2 and elaborate on these methods for each category in the following sections. Drawing inspiration from the existing surveys [15, 86–88] and also carefully considering the unique characteristics of graph OOD generalization methods, our categorization reflects the *primary mechanism of action* emphasized by each method, i.e., its core motivation or central design focus as described in the original work, and the component of the graph learning pipeline it primarily contributes, although some hybrid methods inevitably exist between categories. The key conceptual differences among these three categories are also summarized in Table 1 for better clarification, which highlights their goals, main point of modification, and underlying theoretical motivations. We summarize the characteristics of these methods in Table 2.

3 DATA

The OOD generalization ability of machine learning models, including graph models, heavily relies on the diversity and quality of training data [16]. In general, the more diverse and high-quality the training data, the better the generalization performance of graph models. With proper graph augmentation technique, this type of methods can obtain more graph instances with a simple way for training, whose goal can be formulated as:

$$\min_{f_\theta} \mathbb{E}_{X',Y'}[\ell(f_\theta(X'), Y')], \tag{2}$$

where (X', Y') belongs to training set $\mathcal{D}'$ augmented from $\mathcal{D}$. In general, the graph augmentation literature can be summarized into three types of strategies, including *structure-wise augmentations*, *feature-wise augmentations*, and *mixed-type augmentations*.

3.1 Structure-wise Graph Data Augmentation

Since the graph structure (i.e., topology) plays an important role in predicting the properties of graphs, some works focus on structure-wise augmentations for the input graphs to generate more diverse training topologies that potentially cover some unobserved testing topologies, leading to better OOD generalization ability. Here we mainly review the representative graph data augmentation approaches that *claim to or have practically been verified to* improve the OOD generalization in the paper, the same below. Please refer to the graph augmentation surveys [18, 19] for more details of other methods.

GAug (Graph Augmentation) [24] proposes to generate augmented graphs via a differentiable edge predictor for improving the generalization. It finds that the edge predictors can effectively encode class-homophilic structure to promote intra-class edges and demote inter-class edges in the given graph structure. Such edge manipulation can not only benefit the prediction accuracy but the generalization ability of the graph models. GAUG uses an edge prediction module to modify the given input graph for the downstream training and inference processes. It can also learn to

generate possible new edges for the input graph. The performance of node-level classification tasks can be improved without any modification at inference time. Based on both denoised structure and mimic variability, it boosts the generalization capability.

MH-Aug (Metropolis-Hastings Data Augmentation) [25] further proposes graph augmentation from a perspective of a Markov chain Monte Carlo sampling [89] to flexibly control the strength and diversity of augmentation. A sequence of augmented samples are drawn from the explicitly designed target distribution that controls the augmentation. For tackling the infeasibility of direct sampling from the complex distribution, it adopts the Metropolis-Hastings algorithm to obtain the augmented samples. Instead of random graph augmentations, this method is more controllable, including an efficient strategy to measure and control the augmentation strength reflecting the structural changes of ego-graphs (or samples in node classification). Finally, the OOD generalization power is increased by the diverse augmented training samples.

KDGA (Knowledge Distillation for Graph Augmentation) [26] identifies the negative augmentation problem of the graph augmentation methods above, namely these methods could cause overly severe distribution shifts between the augmented graphs for training and the graph for testing, leading to suboptimal generalization. KDGA is a graph structure augmentation method proposed based on the knowledge distillation technique to reduce the potential negative effects of distribution shifts. Specifically, it extracts the knowledge from the GNN teacher model trained on the augmented graph data and leverages such knowledge in a partially parameter-shared student model that is tested on the given input graph. The experiments on both homophily and heterophily graph datasets show the effectiveness in node-level tasks.

3.2 Feature-wise Graph Data Augmentation

Besides structure-wise augmentations introduced above that remove or add edges for the input graph, some techniques on manipulating node features are also developed recently, showing effectiveness in enhancing the OOD generalization.

GRAND (Graph Random Neural Network) [27] is one simple yet effective feature-wise augmentation method for improving the generalization. It first randomly drops on node features either partially or entirely and then propagates the perturbed node features over the input graph. Therefore, each node of the input graph can eliminate the excessive sensitivity to specific neighborhoods that could induce poor OOD generalization. Under the homophily assumption [90], it stochastically creates different augmented representations for each node. The consistency loss minimizes the distance of the representations learned from the augmented graphs.

FLAG (Free Large-scale Adversarial Augmentation on Graphs) [28] is another simple, scalable, and general graph data augmentation method for better generalization. It proposes to iteratively augment node features in input node feature space with gradient-based adversarial perturbations during training, while keeping graph structures unchanged. It leverages the free adversarial training method [91] to craft adversarial data augmentations. Due to its simple and scalable design, this method can conduct efficient training on some large-scale datasets and also can be easily incorporated into the training pipeline of common GNN backbones. Different from GRAND that is only designed for tasks on nodes, FLAG can be utilized into node/link/graph level tasks.

LA-GNN (Local Augmentation for GNN) [29] proposes a local augmentation for GNNs to learn the distribution of the node features of the neighbors conditioned on the center node's feature. Specifically, it first exploits a generative model to conduct the pre-training for learning the conditional distribution of the neighbors' node features of the center node's feature. Then, the learned distribution can be used to generate feature vectors associated with the center node as additional input for each training iteration. Since the pre-training of the generative model and downstream GNN training are decoupled, this data augmentation method is also model-agnostic, which can be applied to most GNN backbones in a plug-and-play manner. The feature vectors of new nodes can be directly generated via the generative model, so that it can enhance the generalization of the unseen testing nodes. The main difference between LA-GNN with some feature-wise graph augmentations above is that it pays more attention to the local information of the node neighbors rather than only focusing on global augmentation concerning the properties of the whole distribution of the graph.

3.3 Mixed-type Graph Data Augmentation

Moreover, for combining the advantages of structure-wise and feature-wise graph augmentation methods, some works do not conduct single type of augmentation on graph topology or node feature, but in the mixed-type paradigm, which are increasingly popular in the community for improving OOD generalization.

GraphCL (Graph Contrastive Learning) [30] first proposes four general data augmentations for graph-structured data, including node dropping, edge perturbation, attribute masking, and subgraph sampling. Specifically, node dropping is to randomly remove nodes as well as the links to neighbors. And the edge perturbation is to randomly add or remove a fraction of edges. Attribute masking is to mask off certain node attributes by setting the attributes to Gaussian noises. Subgraph sampling is to sample a subgraph using random walk, which includes a fraction of nodes from the input graph. After obtaining the augmented samples of the input, it makes the graph encoder maximize representation consistency under augmentations and has shown good OOD generalization ability in graph classification [92].

GREA (Graph Rationalization Enhanced by Environment-based Augmentations) [31] proposes a data augmentation strategy based on environment replacement to improve the rationale identification accuracy of the input graphs for OOD generalization. The graph rationale is defined as a part of each input graph, i.e., the representative subgraph, that best supports the prediction and can be OOD generalizable. The authors argue that existing augmentation methods (e.g., GraphCL) are mainly heuristic modification to the input graphs, which could not directly support the identification of graph rationales. They generate an augmented example by replacing the environment subgraph of the input graph with the environment subgraph of another graph and encourage the augmented examples to have the same label of the input graph. Considering the high complexity of explicit subgraph decoding and encoding, it turns to implicitly conduct rationale-environment separation and representation learning for the original and augmented graphs in latent space. Based on the accurately identified rationale of the input graph, they verify that the OOD generalization ability is improved.

LiSA (Label-invariant Subgraphs to Construct Augmented Environments) [32] is one inspiring and effective method to generate several augmented domains based on label-invariant subgraphs extracted from the source domain for OOD generalization. It is a promising graph data augmentation method designed specifically

for achieving graph OOD generalization. Since distribution shifts arise from domain disparities, LiSA ensures the graph predictor performs consistently across domains. To address the challenge of collecting sufficient domains, LiSA generates augmented domains by using variational subgraph generators to output diverse subgraphs while maintaining critical predictive information. An energy-based regularization promotes diversity by enlarging the distances between distributions of different augmented domains, while an information constraint ensures subgraphs retain label-relevant information. These augmented domains preserve consistent predictive relationships, enabling the graph predictor to generalize effectively on OOD testing graphs in unseen domains.

AIA (Adversarial Invariant Augmentation) [33] proposes a graph augmentation technique to alleviate the covariate shift problem that is one specific scenario in graph OOD generalization. The authors claim that existing graph augmentation strategies suffer from limited environments or unstable causal features, restricting their OOD generalization ability under covariate shift data. To tackle this problem, AIA first proposes two principles for graph augmentation, which are environmental diversity and causal invariance. The environmental diversity principle encourages the graph augmentation to extrapolate unseen environments (or domains). And the causal invariance principle reduces the distribution gap between the augmented graph data and unseen testing graph data. The method consists of two main modules, including adversarial augmenter to adversarially learn the masks on both graph topology and node features for enhancing environmental diversity, causal generator to output the masks that capture causal information. Based on the two principles and corresponding designs, AIA can get rid of vulnerability under covariate shift.

MARIO (Model-Agnostic Recipe for Improving OOD Generalization) [34] enforces representation consistency across diverse augmented views via graph augmentation, and incorporates conditional mutual information regularization to suppress redundant information while preserving task-relevant features. By jointly addressing augmentation-induced variability and representation redundancy, MARIO effectively mitigates overfitting to spurious correlations and achieves OOD generalization on both node- and graph-level classification tasks.

Besides, in parallel with the development of graph neural networks, **Mixup** and its variants [35, 93], as general data augmentation methods that generate new instances based on the interpolation of the given instances, have been theoretically and empirically shown to improve generalization ability in the fields of computer vision [94] and natural language processing [95]. The similar strategies are also applied in graphs [96–101]. For example, **GraphMix** [96] adopts manifold mixup [93] on node classification tasks by jointly training a fully-connected network (FCN) and a GNN. The loss of FCN is computed using manifold mixup while the loss of GNN is computed normally. A parameter sharing strategy is utilized between the FCN and GNN to help the transfer of critical node representations from the FCN to the GNN. **G-Mixup** [97] interpolates the node features and graph structure in the embedding space as data augmentation, i.e., interpolating the hidden representations of graphs. **NodeAug** [100] analogizes Mixup with a two-branch graph convolution module. It mixes the raw features of a pair of nodes, and feeds them into the two-branch GNN layer, followed by mixing their hidden representations of each layer. **ifMixup** (intrusion-free Mixup) [99] applies Mixup not for the latent representations but directly on the graph data. Due to the issue that graph data are irregular and the nodes of two graphs are not aligned, ifMixup assigns indices to the nodes arbitrarily and matches the nodes with the indices. $\mathcal{G}$**-Mixup** [101] tackles the key challenges when mixing up directly on the graph data, as graph data is irregular and not well-aligned, and graph topology between classes is divergent. Specifically, it first adopts graphs within the same class to estimate a graphon. After that, it does not manipulate graphs directly, but interpolates graphons of different classes in the Euclidean space to obtain the mixed graphons, where the synthetic graphs are produced via sampling based upon the mixed graphons. This method performs well in graph classification datasets with distribution shifts, reflecting its promising OOD generalization. **OOD-GMixup** [102] addresses hybrid structure distribution shifts through controllable data augmentation. It first extracts task-relevant graph rationales to eliminate spurious correlations. Then, it generates virtual samples via manifold mixup and calibrates them using Extreme Value Theory to reweight training, improving OOD generalization. Since these methods share similar ideas, we use the notation "Mixup" to denote these Mixup-based methods that are introduced above in Figure 2 and Table 2.

4 MODEL

Besides augmenting the input graph data to assist achieving good OOD generalization, there are branches of works that specially design new graph models, i.e., f_θ in Eq. (1). By introducing some prior knowledge to model design, the graph model is endowed with the ability to produce graph representation with the properties that could help to improve OOD generalization. Along this branch, there are two kinds of popular techniques: *disentanglement-based graph models* and *causality-based graph models*.

Distinction between Disentanglement-based and Causality-based Methods. While both aim to extract stable, task-relevant information while reducing the influence of spurious patterns, they are built upon fundamentally different theoretical principles and modeling strategies. Our categorization is based on their *core mechanisms for achieving OOD generalization* either through statistical or causal assumptions. **Disentanglement-based methods** originate from representation learning and aim to decompose latent representations into statistically independent components, each corresponding to a distinct latent factor. These methods emphasize modular and interpretable representations, often implemented via multi-channel encoders or routing-based mechanisms [36]. Notably, they do not require extra prior knowledge of the data-generating process or assumptions about causality. **Causality-based methods**, by contrast, are motivated by principles from causal inference. They assume that the observed graph data is generated from the underlying causal assumptions (e.g., structural causal model), and seek to learn representations that are stable across different interventions. Techniques in this category often include confounder balancing, backdoor/frontdoor adjustment, and counterfactual reasoning. Although both approaches aim to improve OOD generalization, they differ in the type of information they seek to capture: disentanglement-based methods focus on identifying statistically independent factors in the data, while causality-based methods aim to model the underlying causal mechanisms that govern the data.

4.1 Disentanglement-based Graph Models

In this section, we introduce the graph models based on disentanglement for OOD generalization.

TABLE 2: A summary of graph OOD generalization methods. "Task" denotes the task type that each method focuses on, including node/link/graph level tasks. "Shift Type" denotes the type of distribution shifts that each method can handle, including topology-level (i.e., graph size and graph structure) and feature-level (i.e., node features) distribution shifts. "Backbone agnostic" indicates whether the method can be used for other GNN backbones. "$|\mathcal{E}| > 1$" indicates whether the method relies on multiple environments during the training process.

Category	Subcategory	Method	Task			Shift Type			Backbone Agnostic	$\lvert\mathcal{E}\rvert > 1$
			Node	Link	Graph	Size	Structure	Feature		
Data	Structure-wise Graph Data Augmentation	GAug [24]	✓				✓		✓	
		MH-Aug [25]	✓				✓		✓	
		KDGA [26]	✓				✓		✓	
	Feature-wise Graph Data Augmentation	GRAND [27]	✓					✓	✓	
		FLAG [28]	✓	✓	✓			✓	✓	
		LA-GNN [29]	✓					✓	✓	
	Mixed-type Graph Data Augmentation	GraphCL [30]	✓		✓		✓	✓	✓	
		GREA [31]			✓		✓	✓	✓	
		LiSA [32]	✓		✓		✓	✓	✓	
		AIA [33]			✓		✓	✓	✓	
		MARIO [34]	✓		✓		✓	✓	✓	
		Mixup [35]	✓		✓		✓	✓	✓	
Model	Disentanglement-based Graph Models	DisenGCN [36]	✓				✓	✓		
		IPGDN [37]	✓				✓	✓		
		FactorGCN [38]			✓		✓	✓		
		DisC [39]			✓		✓	✓	✓	
		NED-VAE [40]			✓		✓	✓		
		DGCL [41]			✓		✓	✓	✓	
		IDGCL [42]			✓		✓	✓	✓	
		OOD-GCL [43]			✓		✓	✓	✓	
	Causality-based Graph Models	OOD-GNN [6]			✓	✓	✓	✓	✓	
		StableGNN [44]			✓		✓	✓	✓	
		DGNN [45]	✓				✓	✓	✓	
		CAL [46]			✓		✓	✓	✓	
		DSE [47]			✓		✓		✓	
		CIGA [48]			✓	✓	✓	✓	✓	
		EQuAD [49]			✓	✓	✓	✓	✓	
		CSIB [50]			✓	✓	✓	✓	✓	
		G-Splice [51]			✓	✓	✓	✓	✓	
		CaNet [52]	✓				✓	✓	✓	
		E-invariant GR [53]			✓	✓		✓	✓	
		gMPNN$^{\bullet\bullet}$ [54]		✓		✓			✓	
		CFLP [55]		✓			✓		✓	
		Gem [56]	✓		✓		✓		✓	
Learning Strategy	Graph Invariant Learning	GIL [57]			✓		✓	✓	✓	✓
		C2R [58]			✓		✓	✓	✓	✓
		DIR [59]			✓		✓	✓	✓	✓
		GSAT [60]			✓		✓	✓	✓	
		VIVACE [62]			✓		✓	✓	✓	
		UIL [61]			✓		✓	✓	✓	✓
		EERM [63]	✓				✓	✓	✓	✓
		INL [64]	✓				✓	✓	✓	✓
		FLOOD [65]	✓				✓	✓	✓	✓
		GraphMETRO [66]	✓		✓	✓	✓	✓	✓	
		DIDA [67]		✓			✓	✓	✓	✓
		SILD [68]		✓			✓	✓	✓	✓
		EAGLE [69]		✓			✓	✓	✓	✓
		SR-GNN [70]	✓				✓	✓	✓	✓
		SizeShiftReg [71]			✓	✓			✓	
		StableGL [72]	✓				✓	✓		✓
	Graph Adversarial Training	DAGNN [73]			✓		✓	✓		✓
		GNN-DRO [74]	✓				✓	✓	✓	
		GraphAT [75]	✓				✓	✓	✓	
		CAP [76]	✓				✓	✓	✓	
		WT-AWP [77]	✓		✓		✓	✓	✓	
		OAD [78]	✓				✓	✓	✓	
	Graph Self-supervised Learning	Pretraining-GNN [79]			✓		✓	✓	✓	
		PATTERN [80]			✓	✓			✓	
		DR-GST [81]	✓				✓	✓	✓	
		GraphCL [30]	✓		✓		✓	✓	✓	
		RGCL [82]			✓		✓	✓	✓	
		GAPGC [83]			✓		✓	✓	✓	
		GT3 [84]			✓		✓	✓	✓	
		HomoTTT [85]	✓				✓	✓	✓	

The formation of a real-world graph typically follows a complex and heterogeneous process driven by the interaction of many latent factors. Disentangled graph representation learning aims to learn representations that separate these distinct and informative factors behind the graph data and characterize these factors in different parts of the factorized vector representations [36]. Such representations have been shown to enhance OOD generalization [103, 104]. The existing methods fall into three groups, i.e., supervised disentanglement methods [36–39], unsupervised generative disentanglement methods [40], and self-supervised contrastive disentanglement methods [41, 42].

DisenGCN [36] is the first method to learn disentangled node representations, whose key ingredient is a disentangled multichannel convolutional layer DisenConv. Executing inside DisenConv, the proposed neighborhood routing mechanism is to identify the factor that may cause the link from a center node to one of its neighbors, and accordingly send the neighbor to the channel responsible for that factor. It infers the latent factors by iteratively analyzing the potential subspace clusters formed by the node and its neighbors, after projecting them into several subspaces. The authors prove that after a sufficient number of iterations, the proposed neighborhood routing mechanism can converge.

Therefore, each channel of DisenConv can extract features specific to only one disentangled latent factor from the neighbor nodes, and perform a convolution operation independently. By stacking multiple DisenConv layers, DisenGCN is able to extract information beyond the local neighborhood and produce disentangled representations. Since the latent factors of nodes are disentangled, it could lead to better OOD generalization performance.

IPGDN (Independence Promoted Graph Disentangled Network) [37] extends DisenGCN [36] by explicitly encouraging the latent factors to be as independent as possible in addition to the neighborhood routing mechanism for disentangling latent factors behind graphs. It minimizes the dependence among different representations with a kernel-based measure Hilbert-Schmidt Independence Criterion (HSIC) [105]. Specifically, to disentangle the target node, the convolution layer of IPGDN first constructs features from different aspects of its neighbors via disentangled representation learning, and then encourages the independence among latent representations through minimizing HSIC to obtain the final results. Note that the disentangled representation learning and independence regularization are jointly optimized in a unified framework, leading to more disentangled representations when compared with DisenGCN. And both DisenGCN [36] and IPGDN [37] are proposed for handling node-level tasks on graphs.

FactorGCN (Factorizable GCN) [38] is a disentangled GNN model for graph-level representation learning. It adopts a factorizing mechanism by decomposing input graphs into several interpretable factor graphs for graph-level disentangled representations. Each of the factor graphs is separately sent to a GCN, tailored to aggregate features in terms of only one disentangled latent factor, followed by an aggregating operation that concatenates together all derived features of disentangled latent factors. The final produced graph-level representations present block-wise interpretable features, and each of the factorized representations corresponds to a disentangled and interpretable relation space. These steps constitute one layer of FactorGCN, so that FactorGCN can produce a hierarchical disentanglement with various numbers of factor graphs at different levels by stacking a number of layers to disentangle the input data at different levels.

Compared with the methods disentangling latent factors, **DisC** (Disentangled Causal Substructure) [39] is a disentangled GNN model directly disentangling causal and noncausal information of the input graph. By explicitly disentangling the input graph into causal and bias subgraphs, this method can only utilize the causal substructures to make stable predictions when severe bias appears under distribution shifts. Specifically, it first filters edges into causal and bias (i.e., noncausal) subgraphs by a parameterized edge mask generator, whose parameters are shared across entire datasets. The edge masker is expected to indicate the importance for each edge and extract causal and bias subgraphs. Then, the causal and bias subgraphs are fed to two GNNs trained with causal-aware weighted cross-entropy loss and bias-aware generalized cross-entropy loss respectively, leading to disentangled representations. Next, it further permutes the latent representations extracted from different graphs to generate more training samples. Although containing both causal and bias information, the causal and bias subgraph of newly generated samples are decorrelated. Finally, the proposed model could focus on the true correlation between the disentangled causal subgraphs and labels for achieving OOD generalized prediction.

Besides the supervised methods above, there exist some unsupervised disentangled methods.

NED-VAE (Node-Edge Disentangled Variational Autoencoder) [40] is a deep unsupervised generative approach for disentanglement learning on graphs, which can automatically capture the independent latent factors in both edges and nodes from attributed graphs. The objective is designed for node-edge joint disentanglement by optimizing three sub-encoders (i.e., a node encoder, an edge encoder, and a node-edge co-encoder) that learn the three types of representations, and two sub-decoders (i.e., a node-decoder and an edge decoder) that co-generate both nodes and edges to model the complicated relationships between nodes and edges. The base NED-VAE can also be extended to realize the group-wise and variable-wise disentanglement to support more fine-grained disentanglement.

Since reconstruction in unsupervised generative methods could be computationally expensive and even introduce bias that has a negative effect on the learned representations, **DGCL** (Disentangled Graph Contrastive Learning) [41] first proposes to learn disentangled graph representations with self-supervision. Specifically, it first identifies the latent factors behind the input graph and derives its factorized representations by the tailored disentangled graph encoder whose key ingredient is a multi-channel message-passing layer. Each of the factorized representations describes a latent and disentangled aspect pertinent to a specific latent factor of the graph. Then it conducts factor-wise contrastive learning in each representation subspace characterized by each factor independently instead of in the whole representation space. This tailored design can encourage that each disentangled factor of the factorized representations is sufficiently discriminative only under one specific aspect of the whole graph, so as to help the graph encoder produce disentangled graph representations that independently reflect the expressive information of latent factors. Unlike generative models, contrastive learning is an instance-wise discriminative approach that makes similar instances closer and dissimilar instances far from each other in representation space [106, 107], so it can eliminate computationally expensive graph reconstruction and learn informative graph representations.

To further promote the disentanglement of the learned graph representations, **IDGCL** (Independence Promoted Disentangled Graph Contrastive Learning) [42] further extends DGCL by explicitly employing HSIC [105] to eliminate the dependence among disentangled representations that reflect different aspects of graphs pertinent to different latent factors. Since the disentangled graph representations are expected to capture mutually exclusive information in terms of the latent factors, IDGCL formulates the statistical independence among different latent representations effectively. The factor-wise contrastive representation learning and independence regularization are jointly optimized in a unified framework so that the disentangled graph encoder can produce better disentangled graph representations. Compared with the existing methods, IDGCL encodes a graph with multiple disentangled representations in a self-supervised manner, making it possible to explore the meaning of each channel, which benefits in more explainability and OOD generalization for producing graph representations.

OOD-GCL (OOD Generalized Disentangled Graph Contrastive Learning) [43] further introduces a theoretically-guaranteed disentangled graph contrastive learning model to address OOD generalization challenges. By employing a disentangled graph encoder and tailored invariant self-supervised learning, it can capture invariant latent factors, ensuring generalized graph representations under distribution shifts. Theoretical analyses con-

firm its ability to provably learn disentangled graph representations and achieve OOD generalization based on the learned disentangled graph representations.

4.2 Causality-based Graph Models

In this section, we introduce the graph models based on causality for OOD generalization.

Causal inference is one important technique to achieve OOD generalization. Graph machine learning models tend to exploit subtle statistical correlation existing in the training set even though it is a spurious correlation (unexpected "shortcut") for predictions to boost training accuracy. The performance of graph models that heavily rely on the spurious correlations can be substantially degraded since the spurious correlations could change in the wild OOD testing environments. In contrast, the causality-based graph models supported by causal inference theory can inherently capture causal relations between input graph data and labels that are stable under distribution shifts [108], leading to good OOD generalization. The existing methods can be divided according to their theoretical ground including confounder balancing [6, 44, 45], predefined structural causal model [46, 47, 53, 54], and counterfactual inference [55] and Granger causality [56].

4.2.1 Confounder Balancing based Methods

Some methods [6, 44, 45] introduce confounder balancing into graph models.

OOD-GNN [6], backed by confounder balancing theory [109] in causality, first tackles the OOD generalization problem by a non-linear decorrelation operation on graphs. Specifically, OOD-GNN proposes to eliminate the statistical dependence between causal and noncausal graph representations of the graph encoder by a nonlinear graph representation decorrelation method utilizing random Fourier features [110], which scales linearly with the sample size and can get rid of spurious correlations. The parameters of the graph encoder and sample weights for graph representation decorrelation are optimized iteratively to learn discriminant graph representations for predictions. The decorrelation operation actually has the same effect with confounder balancing that encourages the independence between treatment and confounder. The graph encoder trained on the weighted dataset can estimate the causal effect of the variables in graph representations to the labels more accurately, while getting rid of the spurious correlations. In this way, OOD-GNN achieves the satisfactory performance on several graph benchmarks with various types of distribution shifts (i.e., shifts on graph sizes, node features, and graph structures), indicating its strong OOD generalization ability in the wild environments.

StableGNN [44] proposes to exploit a differentiable graph pooling layer to extract subgraph-based decorrelated representations based on sample reweighting, which is similar in principle to OOD-GNN. First, the graph high-level variable learning component employs a graph pooling layer [111, 112] to map nearby low-level nodes to a set of clusters, where each cluster is expected to be one densely-connected subgraph unit of original graph. Then, it generates the cluster-level embeddings through aggregating the node embeddings in the same cluster, and aligns the cluster semantic space across graphs through an ordered concatenation operation. The cluster-level embeddings act as the high-level variables for graphs. Next, the sample weights are optimized to eliminate the statistical dependences between these high-level variables. Thus, the graph encoder can concentrate more on the true connection between discriminative substructures and labels, leading to good OOD generalization ability.

In addition to the graph-level decorrelation models above, **DGNN** (Debiased GNN) [45] is a node-level decorrelation model with a similar methodology with StableGNN [44] that removes the spurious correlations on nodes to achieve stable predictions under distribution shifts. Specifically, it proposes a framework for OOD generalized node representation learning by jointly optimizing a decorrelation regularizer and a weighted GNN model. The decorrelation regularizer is expected to learn a set of sample weights for eliminating the spurious correlation between causal and noncausal node information for OOD generalization. And the learned sample weights via the decorrelation regularizer are used to reweight the prediction loss of GNN model so that the prediction could be OOD generalized.

4.2.2 Structural Causal Model based Methods

Some methods [46–48, 53, 54] take the structural causal model (SCM) into account in their model designs. In general, the SCM describes the underlying causal mechanisms. It can improve OOD generalization when introducing appropriate causal mechanisms into model designs.

CAL (Causal Attention Learning) [46] takes a causal look at the GNN model and constructs a structural causal model via presenting the causality among five variables: graph data, causal feature, shortcut feature, graph representation, and prediction. Based on this SCM, they focus on the backdoor path between causal feature C and prediction, wherein the shortcut feature S plays a confounder role. This backdoor path could form spurious correlation, namely using the shortcut feature instead of using causal feature to make predictions, leading to poor OOD generalization under distribution shifts. Therefore, this method exploits the do-calculus on the causal feature to cutting off the backdoor path (i.e., backdoor adjustment [113]), and gets rid of the confounding effect. Finally, it can learn the true relationships between the causal feature and prediction, without being influenced by the unstable shortcut features, which enhances OOD generalization on graph classification tasks.

DSE (Deconfounded Subgraph Evaluation) [47] proposes to faithfully measure the causal effect of explanatory subgraphs on the prediction. The authors claim that distribution shift is hardly measurable, so that it is hard to block the backdoor path from causal subgraph to label by the backdoor adjustment given the predefined SCM. So, they utilize front-door adjustment and introduce a surrogate variable of the causal subgraphs. Instead of adopting the feature removal principle that is used in assessing the explanatory subgraph, it designs a generative model, termed conditional variational graph auto-encoder, to generate the possible surrogates that conform to the data distribution. Therefore, it can conduct unbiased estimation of the relation between causal subgraph and label. Since evaluating the explanatory causal subgraphs unbiasedly, it mitigates the out-of-distribution effect and achieves good OOD generalization.

CIGA (Causality Inspired Invariant Graph Learning) [48] further categorizes the latent interaction between causal part C and noncausal part S into fully informative invariant features (FIIF) and partially informative invariant features (PIIF), depending on whether the latent causal part C is fully informative about label Y, i.e.,$(S, E) \perp\!\!\!\perp Y|C$. For FIIF assumption, the noncausal part S is directly controlled by the causal part C. And for PIIF, the noncausal part S is indirectly controlled by the causal part C

through the label Y. The two SCMs exhibit different behaviors in the observed distribution shifts. If one of FIIF or PIIF is excluded, the performances of graph OOD generalization can degrade dramatically. Similarly, CIGA instantiates the causal part C as the critical subgraph that includes the information about the underlying causes of the label. So the OOD generalization can be achieved by identifying this critical subgraph that maximally preserves the intra-class information among different training environments, hence the predictions will be stable to distribution shifts. **EQuAD** (Encoding-Quantifying Decorrelation) [49] improves upon CIGA by identifying spurious and causal features through a quantification mechanism, which maps spurious features into a compact space for effective decorrelation. It also incorporates a sample-specific reweighting strategy to address data imbalance.

CSIB (Causal Subgraphs and Information Bottlenecks) [50] leverages SCMs to identify invariant subgraphs that causally influence labels across environments. It distinguishes FIIF and PIIF scenarios, and integrates an information bottleneck to suppress spurious features, enabling graph OOD generalization under complex distribution shifts.

G-Splice (Graph Splicing for Structural Linear Extrapolation) [51] integrates SCMs to address graph OOD generalization by explicitly modeling causal and environmental subgraphs. By identifying causal patterns from environment-dependent features, the proposed framework ensures that extrapolated graph structures maintain causal validity. It leverages SCM to generate diverse and causally consistent OOD samples through non-Euclidean space linear extrapolation, significantly enhancing the generalization capabilities of GNNs under complex distribution shifts.

CaNet (Causal Intervention for Network Data) [52] builds on causal intervention theory to address the confounding bias in node-level prediction tasks induced by latent environments. It introduces an environment estimator to infer pseudo-environment labels, dynamically guiding a mixture-of-experts GNN predictor. This collaborative learning framework ensures that stable, environment-insensitive relations are captured, improving generalization across diverse distribution shifts.

E-invariant GR [53] proposes a twin network directed acyclic graph [114] as their SCM to learn size-invariant graph representations (GR) that better extrapolate between test and train graph data. Different from the SCMs mentioned above, the proposed SCM depicts the more complex and fine-grained relations among several variables, including graphon, train/test environment, node feature, edge, and graph size. In this SCM, the training graph is characterized by a graphon, which defines both the label and structural and attribute characteristics of graphs. The training environment is indicated by one unobserved environment variable that represents specific graph properties in terms of environments so that it could change between the training and test set. Based on this SCM, the authors propose an approximately size-invariant graph representation that is able to extrapolate to OOD test data and prove that the learned graph representation can perform no worse on the OOD test data than on a test dataset having the same environment distribution as the training data. Furthermore, this method can achieve extrapolations based on only one training environment (e.g., all training graphs have the same size).

Since E-invariant GR [53] only studies the OOD generalization of GNNs for graph classification, **gMPNN$^{\bullet\bullet}$** [54] further extends it to study the OOD generalization of GNNs for link prediction in a similar setting, where test graph sizes are larger than training graphs. Specifically, the authors first proposed a SCM assuming the data generation process for the goal to learn link predictors that generalize under distribution shifts on graph sizes. And they prove nonasymptotic bounds to indicate that as the sizes of test graphs increase, the link predictors based on permutation-equivariant structural node embeddings will converge to a random guess. They show that the output structural pairwise embeddings can converge to embeddings of a continuous function that achieves OOD generalization in link prediction tasks.

4.2.3 Counterfactual Inference and Granger Causality based Methods

Besides, some graph OOD methods are inspired by counterfactual learning [113], which is at the highest level in the causation ladder [115] and answers what would happen in another possible world if something had or had not happened. And some methods are motivated by Granger causality [116], which describes a causal relationship between variables of some feature and label if we are better able to predict label using all available information than if the information apart from such feature had been used.

CFLP (Counter-Factual Link Prediction) [55] focuses on OOD link prediction tasks to learn the causal relationship between the global graph structure and link existence by training GNN-based link predictors to predict both factual and counterfactual links. It aims to deal with the counterfactual question: "would the link still exist if the graph structure became different from observation?" By answering this question, the counterfactual links will be used to train the graph encoder for producing OOD generalized representation. To generate counterfactual link samples, this method employs causal models that treat the information (i.e., learned representations) of node pairs as context, global graph structural properties as treatment, and link existence as outcome. After that, the proposed model can generate counterfactual training link samples and thus learn representations from both the factual (i.e., observed) and counterfactual (i.e., generated) links for improving OOD generalization.

Gem [56], built upon the Granger causality, inputs the original computation graph into the explainer and outputs a causal explanation graph, exhibiting better generalization abilities. This method considers there exists a causal relationship between this edge/node and its corresponding prediction if the prediction performance decreases as some node or edge is missing. Since graph data is inherently interdependent, where nodes and their edges are correlated variables, it further incorporates various graph rules, e.g., connectivity check, to encourage the obtained explanations to be valid and human-intelligible causal subgraphs. Finally, this method can provide interpretable causal explanations and OOD generalized predictions for GNNs.

5 Learning Strategy

Besides graph data augmentation and graph models, some works focus on exploiting training schemes with tailored optimization objectives and constraints to promote OOD generalization, including graph invariant learning, graph adversarial training, and graph self-supervised learning.

5.1 Graph Invariant Learning

First, we introduce the graph invariant learning methods for OOD generalization.

Invariant learning, which aims to exploit the invariant relationships between features and labels across different distributions

while disregarding the variant spurious correlations, can provably achieve satisfactory OOD generalization under distribution shifts [117–119]. When assessing causality is challenging or the strong assumptions are potentially violated in practice, it can approximate the task by searching features that are invariant under distribution shifts [118] for OOD generalization. Invariant learning assumes that the information of each instance for prediction includes two parts, i.e., invariant part whose relationship with the label is stable across different environments, and variant part whose relationship with the label can change across different environments. A good OOD generalization can be obtained when making predictions only on the invariant information. Along this line, there are mainly two types of graph invariant learning methods: invariance optimization [57, 59, 60, 63, 67] and explicit representation alignment [70–72].

5.1.1 Invariance Optimization

These methods are built upon the invariance principle to address the graph OOD generalization problem. The invariance principle assumes the invariance property inside the data, so that it can find such invariance in multiple environments to achieve OOD generalization. The assumption can be formulated as:

Assumption 1. *(Invariance Assumption). There exists a portion of information $\Phi(X)$ inside input instance X such that $\forall e, e' \in \text{supp}(\mathcal{E}), P^e(Y|\Phi(X)) = P^{e'}(Y|\Phi(X))$, where $\mathcal{E}$ denotes all possible environments and $\Phi(X)$ is often called as invariant rationales of input instance X.*

Following the recent invariant learning based OOD generalization studies [117–119], these invariance optimization methods treat the cause of distribution shifts between testing and training graph data as a potential unknown environmental variable e. The optimization objective can be formulated as:

$$\min_{f_\theta} \max_{e \in \text{supp}(\mathcal{E})} \mathcal{R}(f_\theta|e), \tag{3}$$

where $\mathcal{R}(f_\theta|e) = \mathbb{E}_{X,Y\sim P^e}[\ell(f_\theta(\Phi(X)), Y)]$ is the risk of the f_θ on the environment e that makes predictions based on the invariant information $\Phi(X)$. Therefore, as shown in the last column of Table 2, this type of methods relies on explicit multiple-environment split (indicated by $|\mathcal{E}| > 1$) that can be provided in advance or generated during the training process.

GIL (Graph Invariant Learning) [57] is proposed to capture the invariant relationships between predictive graph structural information (i.e., subgraphs or rationales) and labels under distribution shifts for graph-level OOD generalization. One of the main challenges for graph invariant learning is that the environment labels for graphs is generally unobserved or prohibitively expensive to collect, leading that it is difficult to learn invariance in multiple environments. Therefore, this method first studies invariant learning without explicit environment split. Specifically, GIL jointly optimizes three mutually promoting modules, including the invariant subgraph identification module, the environment inference module, and the invariant learning module. First, the invariant subgraph identification module is a GNN-based subgraph generator $\Phi(\cdot)$. Given the input graph G, it identifies the invariant subgraph $\Phi(G)$ and defines the rest of the graph, i.e., the complement of invariant subgraph, as the variant subgraph and denote it as $G\backslash\Phi(G)$. Then, the environment inference module cluster all identified variant subgraphs of the datasets to infer the latent environments. The intuition is that since the invariant subgraph captures invariant relationships between predictive graph structural information and labels, the variant subgraphs in turn capture variant correlations under different distributions, which are environment-discriminative features. Finally, the invariant learning module optimizes the proposed maximal invariant subgraph generator criterion given the identified invariant subgraphs and inferred environments to generate graph representations capable of OOD generalization under distribution shifts. Theories are provided to show that the OOD generalization problem on graphs is equivalent to finding a maximal invariant subgraph generator of GIL, and further prove that GIL satisfies permutation invariance.

C2R (Cooperative Classification and Rationalization) [58] further proposes a cooperative framework by integrating classification and rationalization modules. By clustering non-rationale subgraphs across the dataset, C2R infers global environments instead of local environment, and feeds them to enhance classification, which can improve OOD generalization under distribution shifts.

DIR (Discovering Invariant Rationale) [59] is proposed to handle graph-level OOD generalization tasks by discovering invariant subgraphs $\Phi(G)$ for GNN under interventional distributions. The basic setting of DIR is also different from the traditional setting where environments are observable and attainable, but follows a similar setting with GIL that does not assume explicit environment split in advance. In detail, it uses a GNN-based subgraph generator to split the input graph into invariant and variant subgraphs under distribution shifts, which are encoded by the encoder into representations respectively. Then, the proposed distribution intervener conducts interventions on the variant representations to create multiple interventional distributions as the multiple environments. Finally, the two classifiers that are respectively built upon the invariant and variant subgraphs make predictions for the input graph instance jointly, so that the invariant risk is minimized across different environments. With this strategy, DIR can capture the invariant rationales that are stable across different distributions while filtering out the spurious patterns for OOD generalization.

GSAT (Graph Stochastic Attention) [60] addresses graph-level OOD generalization problem utilizing the attention mechanism to build inherently interpretable GNNs for learning invariant subgraphs $\Phi(G)$ under distribution shifts. The learned invariant subgraphs of GSAT root in the notion of information bottleneck [120]. The attention is formulated as the information bottleneck by injecting stochasticity into the attention mechanism so as to constrain the information flow from the input graph to the prediction. The injected stochasticity over the invariant label-relevant subgraphs can be automatically reduced during the training stage, while that over the variant label-irrelevant subgraphs can be kept. Besides, GSAT also penalizes the amount of information from the input graph data. Finally, GSAT can output the interpretable and OOD generalizable subgraphs that provably do not contain patterns that are spuriously correlated with the task under some assumptions.

UIL (Unified Invariant Learning) [61] proposes a unified framework for graph OOD generalization by jointly enforcing structural and semantic invariance. It separates stable and environmental features via node- and edge-level masks, then estimates stable graphons to capture class-specific structural patterns. By minimizing graphon distances across environments and enforcing label-consistent predictions, it accurately identifies minimal stable features.

VIVACE (Variance Contrastive Estimation) [62] highlights the importance of variant subgraphs, which carry environment-related information often overlooked in prior work. It proposes leveraging

variant subgraphs to estimate spurious correlations and guide the identification of invariant subgraphs. A reweighting mechanism based on inverse propensity scores is further introduced to correct for spurious effects, leading to enhanced OOD generalization.

Besides the graph-level OOD generalized methods, **EERM** (Explore-to-Extrapolate Risk Minimization) [63] is designed to handle node-level tasks under distribution shifts, which can achieve a valid solution for the node-level OOD problem under mild conditions. First, to account for the non-IID nature of nodes on graphs, this method proposes to transform a graph into a set of ego-graphs for center nodes, so that it can formulate the node-level OOD generalization problem inspired by the graph-level problem. Then, it extends the invariance principle with the recursive computation on the induced BFS trees of ego-graphs to consider the structural information. Finally, the GNN backbone is optimized by minimizing the mean and variance of risks from multiple training environments that are generated by the environment generators, while the environment generators are trained by maximizing the variance loss via a policy gradient method.

INL (Invariant Node Representation Learning) [64] also builds upon the invariant learning principle to address distribution shifts in graph data with multiple latent environments. By defining invariant and variant patterns as ego-subgraphs, INL employs contrastive modularity-based graph clustering to infer node environments. It then optimizes a maximal invariant pattern criterion to produce node representations that generalize effectively to unseen distributions. Theoretical guarantees support its performance, and experiments on both synthetic and real-world datasets demonstrate substantial gains over state-of-the-art methods in node classification tasks under distribution shifts.

FLOOD (Flexible Invariant Learning for Out-Of-Distribution Generalization) [65] further introduces a dual approach: invariant representation learning and bootstrapped representation learning. By constructing training environments and also refining the shared encoder during the test phase, FLOOD achieves improved OOD generalization. This framework effectively addresses distribution shifts in both transductive and inductive settings.

GraphMETRO (Graph Mixture-of-Experts for OOD generalization) [66] tackles distribution shifts by leveraging a mixture-of-experts (MoE) framework. Each expert is trained to mitigate a distinct shift component, such as graph size, node degree, or feature noise, via stochastic graph transformations. A gating model dynamically identifies the relevant shift components per input and guides the aggregation of expert outputs into an invariant representation. This architecture enables flexible modeling of heterogeneous distribution shifts.

DIDA (Disentangled Intervention-based Dynamic Graph Attention Network) [67] is the first method to handle graph OOD generalization under more complex spatial-temporal distribution shifts. The existing methods usually focus on only spatial distribution shifts existing on node features or graph structures while can not be directly utilized in more complex scenarios where the distribution shifts can simultaneously exist in spatial and temporal information. Specifically, it first designs a disentangled spatial-temporal attention network to discover the invariant and variant patterns behind the dynamic graphs, which enables each node to attend to all its historic neighbors through a disentangled attention message-passing mechanism. Then, it introduces a spatial-temporal intervention mechanism to create multiple intervened distributions via sampling and reassembling the variant patterns across neighborhoods and time, leading that the spurious correlations between the variant patterns and labels can be eliminated. Note that the variant patterns are highly entangled across nodes and it is computationally expensive if directly generating and mixing up subsets of structures and features to do intervention. So, this method approximates the intervention process with summarized patterns obtained by the disentangled spatio-temporal attention network instead of the original structures and features. Lastly, the invariance regularization is used to minimize prediction variance in multiple-intervened distributions for learning invariant patterns.

Furthermore, **SILD** (Spectral Invariant Learning for Dynamic Graphs) [68] extends DIDA to spectral domain with discovering the invariant and variant spectral patterns for handling distribution shifts on dynamic graphs. **EAGLE** (Environment-Aware Dynamic Graph Learning) [69] addresses OOD generalization on dynamic graphs by modeling spatio-temporal latent environments through environment-aware convolution and disentanglement. It can model invariant patterns while mitigating spurious correlations, achieving good generalization performance under distribution shifts.

5.1.2 Explicit Representation Alignment

The key idea of this line of works is to explicitly align the graph representations among multiple environments (or domains) to learn environment-invariant graph representations for OOD generalization. The graph representation alignment strives to minimize the difference (or encourage the similarity) across multiple environments via the introduced regularization strategy, which can be formulated as:

$$\min_{f_\theta} \mathbb{E}_{X,Y}[\ell(f_\theta(X), Y)] + \ell_{reg}(\mathcal{E}), \tag{4}$$

where $\ell_{reg}(\mathcal{E})$ denotes the loss of the adopted regularizer. And the multiple environments $\mathcal{E}$ for calculating the regularizer are also usually unavailable in advance for most graph scenarios and are generated during the training process.

SR-GNN (Shift-Robust GNN) [70] proposes to address node-level OOD generalization in GNNs by explicitly minimizing the distributional differences between biased training data and a graph's true inference distribution of graphs. It encourages a biased sample of labeled nodes to more closely conform to the distributional characteristics present in an independent and identically distributed sample of the graph. The two kinds of bias occurring in both deeper GNNs and more recent linearized (shallow) versions of these models can be handled. Specifically, SR-GNN first addresses the distribution shift via a regularization over the hidden layers of the network for standard GNN models (e.g., GCN [121]) that iteratively update information upon the graph structure. The regularizations for measuring discrepancy among different distributions can be maximum mean discrepancy (MMD) [122] or central moment discrepancy (CMD) [123]. Then, for the linearized models (e.g., SimpleGCN [124]) that decouple GNNs into non-linear feature encoding and linear message passing, SR-GNN adopts an instance reweighting strategy for encouraging the training examples to be representative over the graph data, since the graph can introduce bias over the features after all learnable layers. It learns a group of optimal instance weights via kernel mean matching (KMM) [125].

SizeShiftReg [71] aims to train GNNs with good size generalization performance from smaller to larger graphs, which adopts a similar idea with SR-GNN [70]. It does not rely on handcrafting GNNs based on specific knowledge or assumptions, but studies a general regularization for any GNNs to be OOD generalizable to

the graph size distribution shifts. The introduced graph coarsening strategy is to simulate the distribution shifts in the size of the training graphs. And the proposed regularization is expected to encourage the GNNs to be OOD generalized. For a given training graph, they minimize the discrepancy measured by CMD [123] between the distributions of the node representations learned by the GNNs from the original training graphs and the coarsened graphs. Under such a training paradigm, the learned GNNs can achieve OOD generalization among different coarsened versions of the graph as well as graphs with unknown size.

StableGL [72] focuses on stable graph learning (GL) to capture environment-invariant node properties and explicitly balance the multiple environments for generalizing well under distribution shifts. Given one input graph as the training environment, they aim to train a GNN that has a high average prediction performance but a low variance of performance on multiple agnostic testing environments. In more detail, the proposed method first performs biased selection on the input training graph to construct multiple training environments. From a local perspective, since one node in graph is partially represented by the other neighbor nodes, this method proposes to capture stable node properties via reweighting the neighborhood aggregation process. From a global perspective, the authors find that the prediction errors in different environments progressively diverge in biased training, eventually leading to unstable performance across environments. Therefore, the proposed method explicitly aligns the training process by reducing the training gap among different training environments, enforcing the learned GNN to generalize well across unseen testing environments. Different from SR-GNN [70] and SizeShiftReg [71] that adopt some discrepancy measurement like MMD or CMD, the regularization in this method is directly to minimize the variance of training losses in several environments.

5.2 Graph Adversarial Training

In this section, we discuss the graph adversarial learning methods for OOD generalization. Adversarial training has been demonstrated to improve model robustness against adversarial attacks and OOD generalization ability. Here we mainly focus on the graph adversarial training methods that improve the generalization ability, while the works protecting GNNs from attacks can be found in the previous survey [22].

DAGNN (Domain Adversarial GNN) [73] is a method motivated by DANN [126] that is one OOD generalization algorithm to learn domain (or environment) invariant graph representations by advocating domain-adversarial learning between the domain classifier and the encoder. In particular, the first objective is to minimize the classification loss in terms of the encoder on the source domain data, and the second objective aims to facilitate the differentiation between the source and target domains. Such graph adversarial training strategy can maximally utilize the domain information to train classifiers for OOD generalized predictions classification. Note that this method is proposed for text classification where the graphs are converted from the documents, thus the domain (or environment) splits are available in the dataset.

GNN-DRO [74] adopts distributionally robust optimization [127] that is one type of classical algorithm to handle distribution shifts for node-level tasks. The GNN is trained by minimizing the worst expected loss over the considered Wasserstein ball, following the assumption that the data distribution resides in a Wasserstein ball centered at empirical data distribution.

In addition to directly extending existing OOD approaches for general machine learning to graph data above, there are some other works taking more account of the properties of graph itself.

GraphAT (Graph Adversarial Training) [75] aims to improve the model's generalization via exploring the adversarial training on graphs. When generating adversarial perturbations on a target sample, GraphAT maximizes the divergence between the prediction of the target sample and its connected samples, meaning that the adversarial perturbations should affect the graph smoothness as much as possible. After that, GraphAT minimizes the graph adversarial regularizer to update model parameters, reducing the divergence between the prediction of the perturbed target sample and its connected samples. And a linear approximation method for calculating the adversarial perturbations efficiently is derived based on back-propagation. By resisting the worst-case perturbations, it can enhance model robustness and generalization.

CAP (Co-Adversarial Perturbation) [76] is proposed from the perspective of loss landscapes during training process. The authors observe GNNs are prone to falling into sharp local minima in loss landscapes in terms of model weight and feature. Therefore, they propose co-adversarial perturbation (CAP) optimization to flatten the weight and feature loss landscapes alternately, which can avoid falling into locally sharp minima and improve generalization ability. Typically, they formulate the co-adversarial training objective to minimize the maximum training loss within a couple regions of model weights and node features. For further tackling the efficiency problem of co-adversarial training, they decouple the training objective and devise the alternating adversarial perturbations: one step to conduct the adversarial weight perturbation and training GNNs, as well as another step to calculate the adversarial feature perturbation for each node to update GNNs.

WT-AWP (Weighted Truncated Adversarial Weight Perturbation) [77] follows the line that flatting local minima to improve generalization for OOD graph data. Since directly applying existing adversarial weight perturbation techniques to train GNNs is not effective in practice induced by the vanishing-gradient issue, WT-AWP uses the loss of adversarial weight perturbation as an additional regularizer with the loss function (e.g., standard cross-entropy) for training GNN. It also removes perturbation in the last layer of the GNN for a more fine-grained control of the training dynamics. Besides the designs for training strategy, a generalization bound for OOD graph classification is also derived.

OAD (Online Adversarial Distillation) [78] is an online adversarial knowledge distillation technique for GNNs. Different from the above methods that introduce adversarial training into the training process of GNNs, this method brings adversarial training to solve the problem caused by the knowledge distillation. Motivated by the knowledge distillation technique can improve the OOD generalization, OAD trains a group of student GNNs in an online fashion with both global and local knowledge. By transferring informative knowledge of teacher network, the OOD generalization performance of student network can be enhanced. To learn the complex structure of the local knowledge, adversarial cyclic learning is proposed to achieve more accurate embedding alignment among student models. It is not only more efficient than vanilla knowledge distillation technique with fewer parameters, but also more effective to handle graph distribution shift.

5.3 Graph Self-supervised Learning

Finally, we introduce the graph self-supervised learning methods for OOD generalization.

Self-supervision as an emerging technique has been employed to train neural networks for more generalizable predictions on the image field [128–130]. It is also shown that self-supervised learning can benefit GNNs in gaining more generalization ability [131], whose motivations are as follows. First, the self-supervised learning tasks encourage the GNN models to capture salient critical information of the input graph while avoiding the learned representations trivially overfitting "shortcuts" information as supervised learning, leading to better OOD generalization. Then, Xu *et al.* [132] also attribute such success to that self-supervised learning could map semantically similar data to similar representations and therefore some OOD testing data might fall inside the training distribution after the mapping.

Here we mainly review the typical graph self-supervised methods that claim to improve the graph OOD generalization. For more details of other graph self-supervised methods, the readers could refer to the surveys [20, 21].

Pretraining-GNN [79] explores several graph pre-training techniques on both node-level and graph-level to improve OOD generalization of GNNs. They encourage GNNs to capture domain-specific knowledge about nodes and edges, in addition to graph-level knowledge such that the learned representations can be more OOD generalized. For node-level pre-training of GNNs, they propose two self-supervised methods, i.e., context prediction and attribute masking. For graph-level pre-training of GNNs, they also provide two options including making predictions about domain-specific attributes of entire graphs (e.g., supervised labels), or making predictions about graph structure namely modeling the structural similarity of two graphs. Overall, such pre-training strategy for GNNs is to first perform node-level self-supervised pre-training and then graph-level multi-task supervised pre-training.

PATTERN [80] is proposed to study the ability of GNNs to generalize from small to large graphs, by proposing a self-supervised pretext task that aims at learning useful d-pattern representations. Although GNNs can naturally be applied to graphs with different sizes, it is largely unknown about the mechanism of such size OOD generalization of GNNs. Therefore, the authors first formalize a representation of local structures called d-patterns for characterizing generalization to new graph sizes. The d-patterns generalize the notion of node degrees to a d-step neighborhood of the center node, which models the values of the node and its d-step neighbors, as seen by GNNs. It is proved that even only a small discrepancy in the d-patterns distribution between the testing and training distributions may result in weight assignments that do not generalize well, indicating the existence of bad global minima with poor generalization. Then, the self-supervised pretext task is proposed aiming at learning useful d-patterns representations from both small and large graphs improving the OOD generalization on graph size with noticeable gains.

DR-GST (Distribution Recovered Graph Self-Training) [81] is a graph self-training framework that can recover the original labeled dataset without distribution shifts. Specifically, it first shows that the equality of loss function in self-training framework under the distribution shifts and the population distribution if each pseudo-labeled node is weighted by a proper coefficient. Due to the intractability of the coefficient, it replaces the coefficient with the information gain after discovering the same changing trend between them. The information gain is respectively estimated via both dropout variational inference and dropedge variational inference. Then, it can recover the shifted distribution with the proposed information gain weighted loss function, which forces the GNN to focus on nodes with high information gain. Overall, DR-GST tackles the distribution shift problem from the perspective of information gain, and proposes a loss correction strategy to improve qualities of pseudo labels. Therefore, more unlabeled nodes can be assigned with pseudo labels whose distribution is the same as that of labeled nodes so as to benefit the OOD generalization ability.

Besides, graph contrastive learning can also be adopted to promote OOD generalization.

GraphCL (Graph Contrastive Learning) [30] is one of the representative self-supervised learning methods for GNNs and has shown its generalization ability in practice. The authors argue that self-supervision with handcrafted pretext tasks relies on heuristics to design, and thus could limit the generality of the learned graph representations. Therefore, they develop the contrastive learning method GraphCL, whose key idea is to make graph representations agree with each other under the proposed four types of transformations for the input graph. The generalizability ability of GraphCL is verified on molecular property prediction in chemistry and protein function prediction in biology.

RGCL (Rationale-aware Graph Contrastive Learning) [82] is proposed to automatically discover rationales as graph augmentations in contrastive learning for further improving the generalization performance in unseen domains with distribution shifts. The authors claim that despite promising performance of some representative methods like GraphCL, etc., the intrinsic random nature makes them suffer from potential semantic information loss, thus hardly capturing the salient information and undermining the generality ability. RGCL is proposed to tackle this problem, which consists of two modules, i.e., rationale generator and contrastive learner. The rationale generator decides fractions to reveal and conceal in the graph, and yields the rationale encapsulating its instance-discriminative information. The contrastive learner makes use of rationale-aware views to perform instance-discrimination of graphs. Thus, it can prevent losing discriminative semantics in augmented views as random augmentation and in turn preserve more rationale information with generalization ability.

GAPGC (Graph Adversarial Pseudo Group Contrast) [83] is a test-time training method designed for GNNs with a contrastive loss variant as the self-supervised objective during testing. Recently the effectiveness of test-time training has been validated to improve the performance on OOD test data, where some self-supervised auxiliary tasks are proposed. The authors argue that the simple augmentations in self-supervised training (e.g., randomly dropping nodes or edges) could harm the label-related critical information in graph representations. Therefore, GAPGC generates relatively reliable pseudo-labels, avoiding the severe shifts caused by the incorrect positive samples. The proposed adversarial learnable augmenter and group pseudo-positive samples can promote the relevance between the self-supervised task and the main task, so as to enhance the performance of the main task. The theoretical evidence is also derived to show that GAPGC can capture minimal sufficient information for the main task from information theory perspective, which benefits the predictions on the OOD testing data.

GT3 (Graph Test-Time Training with Constraint) [84] is another test-time training method on graphs, which proposes a hierarchical self-supervised learning framework. Specifically, it first introduces the global contrastive learning strategy to encourage node representations to capture the global information of the whole graph. The global contrastive learning is based on

TABLE 3: Commonly used synthetic and real-world graph datasets for OOD generalization. "Task" denotes each dataset can be used in graph-level, node-level task or link-level task. "Type" indicates what kind of graph data that each dataset includes. "Cause of Shifts" indicates the reason for inducing distribution shifts between training and testing data. "Metric" is the evaluation metric adopted by each dataset. And "References" denotes the work developing each dataset.

Dataset	Task	Type	Cause of Shifts	Metric	References
Spurious-Motif	Graph	Synthetic Graph	Correlations	Accuracy	[59]
MNIST-75sp	Graph	Superpixel Graph	Feature Noises	Accuracy	[133]
CMNIST-75sp	Graph	Superpixel Graph	Feature Colors	Accuracy	[92, 134]
$D\&D_{200}$	Graph	Molecular Graph	Graph Size	Accuracy	[133]
Graph-SST2	Graph	Text Sentiment	Node Degree	Accuracy	[135]
OGBG-Molhiv	Graph	Molecular Graph	Scaffold	ROC-AUC	[7]
OGBG-Molpcba	Graph	Molecular Graph	Scaffold	Average Precision	[7]
OGBG-PPA	Graph	Protein Network	Species	Accuracy	[7]
DrugOOD	Graph	Molecular Graph	Assay/Scaffold/Size	Accuracy/AUC	[136]
CBA-Shapes	Node	Synthetic Graph	Feature Colors	Accuracy	[134]
Facebook-100	Node	Social Network	Structure	Accuracy	[63]
WebKB	Node	Webpage Network	Structure	Accuracy	[134]
Twitch-Explicit	Node	Social Network	Structure	ROC-AUC	[137]
Elliptic	Node	Bitcoin Transactions	Time	F1 Score	[138]
OGBN-Arxiv	Node	Citation Network	Time	Accuracy	[7]
OGBN-Proteins	Node	Protein Network	Species	ROC-AUC	[7]
OGBN-Products	Node	Co-purchasing	Popularity	Accuracy	[7]
COLLAB	Link	Collaboration Network	Field	ROC-AUC	[67]
Yelp	Link	Social Network	Food Category	ROC-AUC	[67]
ACT	Link	Social Network	Attribute	ROC-AUC	[69]
OGBL-PPA	Link	Protein Network	Biological Throughput	Hits@100	[7]
OGBL-DDI	Link	Drug Interaction Network	Protein-target	Hits@20	[7]

maximizing the mutual information between the local node representation and the global graph representation. Then, it presents the local contrastive learning for distinguishing different nodes from different augmented views of a graph, so that the node representation can capture more local information. Besides, an additional constraint is proposed to encourage that the representations of testing samples are close to the representations of the training samples. The model's OOD generalization capacity for the graph classification task can be enhanced based on this test time training strategy with self-supervised learning.

HomoTTT (Homophily-guided Fully Test-Time Training) [85] is a model-agnostic framework for node classification under OOD settings. It performs fully test-time training using a parameter-free, homophily-based self-supervised contrastive learning objective with adaptive graph augmentation. To avoid performance degradation, it further introduces a homophily-based model selection to selectively apply the adapted model per node.

6 THEORY

In this section, we review some literature focusing on theoretical analyses of the generalization of GNNs.

First, there are some theories mainly developed to derive the generalization bound of GNNs based on different statistical learning theories. **Scarselli *et al.*** [139] provide a generalization bound for GNNs based on VC-dimension [140]. The authors find that the upper bounds on the VC-dimension for GNNs are comparable to the upper bounds for the recurrent neural networks, meaning that the generalization capability of GNNs increases with the number of connected nodes. **Verma & Zhang** [141] take a further step towards deriving a theoretical analysis of GCN [121] based on algorithmic stability [142] and provide generalization bounds for one-layer GCN. They conclude that one-layer GCN with stable graph convolution filters can satisfy the strong notion of uniform stability and therefore are generalizable.

Garg *et al.* [143] study the generalization properties of GNNs on graph classification based on Rademacher complexity. The generalization analysis explicitly considers the local permutation invariance of the GNN aggregation function. The derived Rademacher bounds are tighter than the VC bounds from [139] for GNNs. **Lv** [144] adopts similar theoretical basis with the work [143], providing the Rademacher complexity bound for GCNs with one single hidden layer. The primary difference is that this work accounts for the specific node-level task of GCNs, which only involves a fixed adjacency matrix.

Liao *et al.* [145] establish a PAC-Bayesian generalization bound of GNNs on graph classification. It further improves upon the Rademacher complexity based bound proposed in the work [143], deriving a tighter dependency on the maximum node degree and the maximum hidden dimension. Also, **Ma *et al.*** [146] present a PAC-Bayesian analysis for generalization performances of GNNs on subgroups of nodes under non-IID node-level tasks, which is the key difference compared with the work [145].

Du *et al.* [147] establish Graph Neural Tangent Kernel (GNTK) to characterize the generalization bound of GNNs on graph classification. GNTK is induced by infinitely wide GNNs, whose prediction depends only on pairwise kernel values between graphs, and can be calculated efficiently with an analytic formula. It enjoys the expressive power of GNNs, while inheriting the benefits of graph kernels, e.g., easy to train, provable theoretical guarantees, etc. Based on GNTK, **Xu *et al.*** [132] derive theoretical evidence of generalization capabilities in one-layer GNNs and study the effect of the alignment of network architecture and target algorithmic tasks on OOD generalization. Along with this line, **Zhang *et al.*** [148] prove that using proper tensor initialization and accelerated gradient descent, their algorithm can learn a GNN with one hidden layer having the zero generalization error for regression problems or sufficiently close to the ground-truth model, assuming such a ground-truth model exists.

Considering most methods mentioned above are developed

based on that graph data can be generated and labeled in any arbitrary way which is hard to be satisfied in practice, some works establish generalization bounds that depend on the graph data as follows. **Baranwal *et al.*** [149] study OOD generalization of GNNs under a specific data generating mechanism namely contextual stochastic block model and analyze the relation between linear separability and OOD generalization on graphs. The generalization guarantee for one-layer GCNs on binary node classification is derived. Furthermore, **Maskey *et al.*** [150] consider a generative model graphons for the graphs which is not only theoretically powerful and general, but allows tighter generalization bounds.

In addition to deriving the generalization bound, there are also some theoretical frameworks on causality, invariant learning, and information bottleneck to analyze the OOD generalization capabilities.

Causal inference offers a strong theoretical foundation for improving OOD generalization by focusing on stable causal relationships between input features and labels. Unlike spurious correlations that are sensitive to distribution shifts, causal features remain invariant across environments, providing a reliable basis for OOD generalized predictions. Theoretical frameworks such as structural causal models (SCMs) [46, 48, 51] and causal intervention [52] facilitate the identification and utilization of these causal features, enabling models to capture the true determinants of labels. Additionally, counterfactual reasoning [55] could enhance this perspective by considering hypothetical scenarios, thereby allowing models to better generalize to unseen data. These approaches collectively underline the importance of causality in addressing distribution shifts and establishing a principled basis for generalization, ensuring that predictions are reliable across diverse environments.

Invariant learning provides a principled theoretical framework for OOD generalization by focusing on identifying and leveraging features that maintain stable relationships with labels across different environments [117–119]. This approach assumes that the input data can be decomposed into invariant components, which are consistent predictors of the target, and variant components, which are spurious and environment-specific. The key idea is to optimize for predictive performance while ensuring invariance across training distributions, thereby aligning model predictions with the stable causal mechanisms underlying the data. From a theoretical standpoint, invariant learning relies on the invariance principle, which assumes that the conditional distribution of the label given the invariant features should remain constant across environments. This principle is often adopted through optimization objectives that minimize risks across multiple training environments or regularization techniques that explicitly enforce alignment in representation spaces [57, 59, 64]. By focusing on invariant patterns and discarding variant ones, invariant learning not only enhances OOD generalization but also offers theoretical guarantees under certain assumptions, such as the existence of sufficient environmental diversity [69] or latent invariance within the data [67, 68]. By ensuring that predictions are grounded in invariant features, this framework establishes a foundation for graph OOD generalization.

Information Bottleneck (IB) theory is used in some works for generalized graph learning. The key idea is to maximize the mutual information between task-relevant subgraphs and labels while constraining information from task-irrelevant graph components. For example, GSAT [60] jointly trains the predictor and subgraph extractor, leveraging a stochastic attention mechanism for the information control. InfoIGL [151] introduces a redundancy filter combined with multi-level contrastive learning to extract invariant features of graphs, maximizing the mutual information among graphs of the same class and reducing task-irrelevant noise. Finally, the derived IB-based objective guarantees the removal of spurious correlations, improving OOD generalization.

7 Datasets for Evaluation

To promote further research of graph OOD generalization, we summarize the existing popular graph datasets for evaluation in Table 3. There are three groups of datasets, including datasets for graph-level, node-level, and link-level tasks. These datasets cover multiple sources of graphs (e.g., social network, citation network, molecular graph, etc) and their causes of distribution shifts are also complex and diverse (e.g., time, species, scaffold, etc.).

7.1 Datasets for Graph-level Tasks

First, we review some representative datasets for evaluating the model performances on graph classification tasks.

Spurious-Motif [59]: It is a synthetic dataset created by following the work [152], which is designed for distribution shifts on graph structure. Each graph consists of one motif and one base subgraph. The base subgraph includes Tree, Ladder, and Wheel (denoted by $V = 0, 1, 2$, respectively) and the motif includes Cycle, House, and Crane (denoted by $I = 0, 1, 2$). The ground-truth label Y only depends on the motif I, which is sampled uniformly. The spurious correlation between V and Y is injected by controlling the base subgraphs distribution as: $P(V) = b$ if $V = I$ and $P(V) = (1 - b)/2$ if $V = I$. Intuitively, b controls the strength of the spurious correlation. It can set b to different values in the testing and training set to simulate distribution shifts.

MNIST-75sp [133]: It is a semi-artificial dataset, where each graph is converted from an image in MNIST [153] using superpixels [154]. The nodes are superpixels, and the edges are calculated by the spatial distance between nodes. The node features are the super-pixel coordinates and intensity. The task is to classify each graph into the corresponding handwritten digit labeled from 0 to 9. To simulate distribution shifts on graph features, it generates testing graphs by colorizing images, i.e., adding two more channels and adding independent Gaussian noise to each channel.

CMNIST-75sp [92, 134]: It is also a semi-artificial dataset, consisting of graphs converted from the images in MNIST using superpixels. Different from MNIST-75sp that adds noise to simulate distribution shifts, CMNIST-75sp colorizes the digits with different colors according to the digit labels or dataset split, inspired by the work [117]. Note that there are two choices of CMNIST-75sp to simulate the covariate shifts or concept shifts respectively. For the former choice, the testing data are colorized with unseen colors compared with the colors for the training data. For the latter choice, the colors are correlated with the digit labels for the training data, while colors have different correlations with labels for testing data, respectively.

$\mathbf{D}\&\mathbf{D}_{200}$ [133]: It is a real-world graph classification dataset that consists of 1,178 protein network structures with 82 discrete node labels. The task is to classify each graph into enzyme or non-enzyme class. To create distribution shifts on graph sizes, the training and testing sets are split by graph sizes, i.e., the models are trained on small graphs but tested on larger graphs. Specifically, the training set includes graphs with 30 to 200 nodes while the testing set includes graphs with 201 to 5,748 nodes.

Graph-SST2 [135]: It is a real-world graph dataset originating from a natural language sentimental analysis dataset. Each graph is converted from a text sequence, where nodes represent words, edges indicate relations between words, and label is the sentence sentiment. Graphs are split into different sets according to average node degree to create distribution shifts. The node features are initialized by the pre-trained BERT word embedding [155]. Thanks to the graph semantics, this dataset is more human-understandable for visualizing or analyzing some intermediate results.

OGBG [7]: Open Graph Benchmark (OGB) is a benchmark consisting of realistic, large-scale, and diverse datasets for machine learning on graphs, where OGBG is a subset including several representative datasets for evaluation OOD generalization in graph-level tasks, e.g., OGBG-Molhiv, OGBG-Molpcba, OGBG-PPA, etc. Specifically, **OGBG-Molhiv** and **OGBG-Molpcba** are two graph property prediction datasets with distribution shifts. The task is to predict the target molecular properties. The dataset provides the default scaffold splitting procedure, i.e., splitting the graphs based on their two-dimensional structural frameworks. Note that this scaffold splitting strategy aims to separate structurally different molecules into different subsets, which provides a more realistic and challenging scenario for testing graph OOD generalization. And **OGBG-PPA** consists of undirected protein association neighborhoods extracted from the protein-protein association networks of 1,581 different species. The task is to predict what taxonomic group the given protein association neighborhood graph originates from. The dataset adopts species split, i.e., separating graphs from different species into different subsets.

DrugOOD [136]: It is a benchmark for AI-aided drug discovery, including some realistic molecular graph datasets. It provides an automated pipeline for curating OOD datasets based on a large-scale bioassay dataset ChEMBL [156]. It presents diverse dataset splitting indicators than OGB to generate specific domains that are aligned with the domain knowledge of biochemistry. Rather than only adopting scaffold as the indicator of dataset splitting, it can provide more choices for separating graphs into different subsets in terms of assay and size to create distribution shifts.

7.2 Datasets for Node-level Tasks

Then, we review some representative datasets for evaluating the model performances on node classification tasks.

CBA-Shapes [134]: It is a synthetic dataset created by following the BA-Shapes dataset from the work [152]. The input graph contains a base graph and a set of motifs, where the base graph is a Barabási-Albert (BA) graph on 300 nodes and the set of motifs includes 80 house-structured motifs. The task is to predict the structural role of each node, including the top, middle, or bottom node of a house-structured motif, or the node from the base graph, i.e., a 4-class classification task. Node features are assigned with colors to create distribution shifts, which also have two choices to simulate the covariate shifts or concept shifts. For the former choice, the testing nodes are colorized with unseen colors compared with the colors of the training nodes. For the latter choice, the colors are correlated with the labels of the training nodes, while colors have different correlations with labels of the testing nodes, respectively.

Facebook-100 [63]: It is a real-world node classification dataset which consists of 100 Facebook social network snapshots from the year 2005. Each network contains nodes as Facebook users from a specific American university. The distribution shifts can be introduced by splitting training and testing sets via selecting different universities that the users in a network are from, since these networks have significantly diverse sizes, densities and degree distributions. For example, the default dataset split in the work [63] is to adopt the corresponding networks from three of fourteen universities (e.g., John Hopkins, Cornell, etc.) as training set, and the network from another three universities (i.e., Penn, Brown and Texas) as the testing set. Of course, the other combinations can also be used to evaluate the node-level OOD generalization ability.

WebKB [134]: It is a real-world university webpage network dataset for node classification. The nodes denote webpages and edges are hyperlinks between two webpages. The node features are from the words appearing in the webpage. The task is to predict the classes of webpages including student, project, course, staff, or faculty. The distribution shifts are from splitting the dataset conforming to the domain university. The OOD generalized predictions can be achieved when only using the word contents and hyperlinks of webpages rather than using the university features.

Twitch-Explicit [137]: It is a real-world social network dataset, where nodes are Twitch users and edges are friendships between two users. Node features are games liked, location and streaming habits. Each network is collected from a specific region, including DE, ENGB, ES, FR, PTBR, RU and TW. The seven networks have significantly different structural properties, e.g., densities and maximum node degrees [63]. The distribution shifts between training and testing sets are from splitting the dataset according to the network region.

Elliptic [138]: It is a realistic Bitcoin transaction network dataset consisting of several snapshots, where nodes are transactions and edges are payment flows. The task is to distinguish between licit and illicit transactions in future data. By adopting older snapshots in terms of time as the training set while newer snapshots as the testing set, the distribution shifts can be observed due to some emerging events in the market.

OGBN [7]: It includes some node properties prediction datasets, e.g., OGBN-Arxiv, OGBN-Proteins, and OGBN-Products, which is another subset of the whole OGB [7]. Specifically, **OGBN-Arxiv** is a real-world citation dataset, where nodes are arXiv papers, and edges are citations between papers. Its 40-class prediction task is to predict the subject area of arXiv papers. The node distribution shifts are introduced by splitting papers from different time ranges into training and testing sets. And **OGBN-Proteins** a protein graph, where nodes represent proteins and edges indicate different types of biologically meaningful associations between proteins. The task is to predict the presence of protein functions. The distribution shifts are introduced by splitting protein nodes into different subsets according to the species that the proteins come from. **OGBN-Products** is an Amazon product co-purchasing network. Nodes represent products in Amazon, and edges indicate that the two products are purchased together. The task is to predict the product category. The distribution shifts are created by a more challenging and realistic dataset splitting according to the popularity of products, i.e., using the popular products for training but relatively unpopular products for testing.

7.3 Datasets for Link-level Tasks

Furthermore, we review some representative datasets for evaluating the model performances on link prediction tasks.

COLLAB [67]: It is a link prediction dataset derived from academic collaboration networks. Nodes represent authors, and

edges denote coauthorships on papers published between 1990 and 2006. The dataset is enriched with field-specific information, categorizing edges by the coauthored publication's field, such as "Data Mining", "Database", "Medical Informatics", "Theory", and "Visualization". It spans 16 yearly time slices, capturing the evolution of collaborations over time. The dataset's distribution shifts are introduced by splitting based on the fields of coauthored publications, where "Data Mining" serves as the unseen domain during training, creating a challenging test scenario for OOD generalized link predictions.

Yelp [67]: It is a real-world link prediction dataset originating from customer-business interaction records. Nodes correspond to customers and businesses, while edges represent review interactions over time. The dataset includes data from January 2019 to December 2020. Categories such as "Pizza", "American (New) Food", "Coffee & Tea", "Sushi Bars", and "Fast Food" are used to label interactions. Distribution shifts are introduced by withholding interactions involving "Pizza" as a testing domain, offering a real-world scenario to evaluate models under distribution shifts.

ACT [69]: It documents dynamic student activity within a MOOC (Massive Open Online Course) platform. Nodes represent students, and edges denote their actions, such as course participation or interaction with learning materials. Different categories of actions, including "Lecture Viewing", "Assignment Submissions" and "Forum Participation" are tracked to introduce varying interaction patterns. The distribution shifts are created by excluding specific categories of actions during the training phase, challenging models to generalize across unseen patterns of student behaviors during testing.

OGBL-PPA [7]: It is a real-world graph dataset constructed from protein-protein association networks. Nodes represent proteins from 58 different species, and edges capture biologically meaningful associations, including physical interactions, co-expression, homology, or genomic neighborhoods. Each node is associated with a 58-dimensional one-hot feature vector indicating its species origin. The dataset focuses on link prediction tasks, where the goal is to rank positive protein-protein associations higher than randomly sampled negative edges. The evaluation metric, Hits@100, assesses the proportion of positive edges ranked among the top 100 positions. The dataset introduces distribution shifts through a biological throughput-based splitting strategy: training edges are derived from cost-effective, high-throughput experimental methods or computational techniques, while validation and test edges consist of associations confirmed via low-throughput, resource-intensive laboratory experiments.

OGBL-DDI [7]: It is a real-world graph dataset originating from drug-drug interaction networks. Nodes represent FDA-approved or experimental drugs, and edges indicate interactions where the combined effect of two drugs significantly deviates from their independent actions. The task is to predict new drug-drug interactions by ranking known interactions higher than approximately 100,000 randomly sampled negative interactions. The evaluation metric, Hits@20, measures the proportion of true interactions ranked among the top 20 positions, providing a challenging benchmark for model performance. The dataset employs a protein-target split strategy, where training and validation sets include drugs targeting one set of proteins, while the test set consists of drugs targeting entirely different proteins. This splitting approach ensures that models are evaluated on their ability to generalize to drugs with distinct biological mechanisms, reflecting real-world OOD scenarios in drug discovery.

7.4 Other Benchmarks

In addition, there are also some works that collect these commonly used or more than one datasets above into a standard evaluation open-source benchmark and report the experimental results for some well-known general OOD algorithms and graph OOD methods under the proposed evaluation protocols. Since the details of most datasets have been discussed above, here we review these packages briefly. Specifically, **GDS** [92] collects eight datasets for graph-level tasks reflecting a diverse range of distribution shifts across graphs to compare the performance of popular OOD generalization algorithms and GNN backbones. **GOOD** [134] summarizes more than ten datasets for both graph-level and node-level tasks with diverse types of distribution shifts introduced by combining different domain selection strategies and distribution shift types. It also contains the experiments to show the significant performance gaps between in-distribution and OOD settings and the comparisons among different OOD methods for both general machine learning and the graph field.

8 Discussions

In this section, we summarize this survey and discuss several challenges as well as opportunities worthy of future explorations.

8.1 Summary

The diversity and quality of training graph data play an important role in OOD generalization of graph machine learning approaches. Several graph data augmentation methods, including structure-wise, feature-wise, and mixed-type methods are developed to achieve good performances with simple yet effective paradigms.

Another line of works focuses on exploiting new graph models to promote the OOD generalization capability. Compared to graph data augmentation, these models overall enjoy more solid theoretical ground and more graph-specific designs. The disentanglement-based graph models present good motivations while the causality-based graph models are backed by diverse causal inference theories. These tailored graph models also show promising OOD generalization performances in practice.

Recently, there is a rapid development for graph learning strategies, including graph invariant learning, graph adversarial training, and graph self-supervised learning. Compared with the graph models, these methods pay more attention to the learning process, so that they are more flexible to be compatible with different GNN backbones for enhancing OOD generalization.

To build the theoretical framework of graph generalization, a number of theoretical derivations on generalization bounds are proposed, which benefit the deeper understanding of graph OOD generalization methods. And to promote deeper research, diverse datasets under complex realistic distribution shifts covering node-level and graph-level tasks are adopted to verify the effectiveness of graph OOD generalization methods comprehensively and fairly.

8.2 Future Directions

There exist plenty of opportunities worthy of future explorations.

8.2.1 More Theoretical Guarantees

While some graph OOD generalization methods have demonstrated substantial empirical progress, a critical gap remains in connecting these methods to the theoretical foundations outlined in Section 6. Bridging this gap still requires rigorous theoretical

characterizations of learnable graph OOD generalization problems. Moreover, it is vital to extend the understanding of specific types of distribution shifts, such as covariate shifts, concept shifts, and label shifts, which often interact in complex ways in graph-structured data. Existing works have shown initial success in addressing specific shift types. Future research should explore OOD generalization theories that account for diverse shift types, backed by generalization bounds, causality, invariant learning, or information bottleneck.

8.2.2 GNN Architecture

Recent works [132, 133, 157–159] emphasize the critical role of architecture design in GNNs, such as readout operations, to enable generalization to OOD graph data. These studies provide foundational insights into the interaction between GNN architecture and distribution shifts. To systematically enhance GNNs for OOD generalization, methods for automatically tailoring a customized GNN architecture suitable for each graph instance benefit the predictions under distribution shifts [160], which represent a promising direction. And more research efforts need to be paid on automatically learning OOD generalized GNN architectures suitable for diverse environments.

8.2.3 Environment Split

The majority of general OOD generalization algorithms rely on access to multiple training environments [15]. However, acquiring accurate environment labels for real-world graph data is often prohibitively expensive, limiting the applicability of these methods. Future research could explore developing single-environment OOD generalization methods that leverage graph structure and feature heterogeneity to learn environment splits dynamically. Moreover, real-world graph data often evolves over time, requiring models to adapt to dynamic or continuous environments. Existing works on lifelong learning and continual graph learning [161, 162] provide a foundation for developing methods capable of efficiently updating graph models and learning strategies to generalize across temporal distribution shifts. Extending these methods to dynamically evolving graphs under unknown distribution shifts remains a promising and underexplored research direction.

8.2.4 Test-Time Training for Generalization

Graph test-time training can allow more flexibility in inference time to make use of the inference unlabeled data during the testing stage. It can improve the graph OOD generalization under unknown distribution shifts via solving a test-time task. In addition to the two works [83, 84] introduced in Section 5.3 that adopt contrastive test-time tasks, one more recent attempt **GTrans** [163] proposes to adapt and refine graph data at test-time. And **LEBED** [164] estimates generalization errors of well-trained GNNs on unlabeled test graphs under distribution shifts by leveraging a parameter-free re-training strategy and measuring node prediction and structure reconstruction discrepancies. It is a valuable direction to design more test-time training tasks or explore more test-time training strategies to improve OOD generalization on graphs.

8.2.5 Broader Scope of Applications

OOD graph data widely exist in our daily life. While classical machine learning approaches on graphs have been applied in diverse applications, deploying OOD generalized graph methods in real-world settings with distribution shifts remains an essential and underexplored challenge. Applications such as recommender systems, social networks, traffic prediction, materials science, and risk-sensitive domains like healthcare and finance demand not only predictive accuracy but also trustworthiness in decision-making [165–170]. The integration of domain knowledge is suggested as a potential avenue to improve graph OOD generalization.

REFERENCES

[1] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "Deepinf: Social influence prediction with deep learning," in *KDD*, 2018.
[2] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey," *ACM Computing Surveys*, 2022.
[3] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE TKDE*, 2017.
[4] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *IJCAI*, 2018.
[5] Y. Bengio *et al.*, "A meta-transfer objective for learning to disentangle causal mechanisms," *ICLR*, 2019.
[6] H. Li, X. Wang, Z. Zhang, and W. Zhu, "Ood-gnn: Out-of-distribution generalized graph neural network," *IEEE TKDE*, 2022.
[7] W. Hu *et al.*, "Open graph benchmark: Datasets for machine learning on graphs," *NeurIPS*, 2020.
[8] S. Yang *et al.*, "Financial risk analysis for smes with graph-based supply chain mining." in *IJCAI*, 2020.
[9] C. Agarwal, H. Lakkaraju, and M. Zitnik, "Towards a unified framework for fair and stable graph representation learning," in *UAI*, 2021.
[10] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *ECCV*, 2020.
[11] J. Shlomi, P. Battaglia, and J.-R. Vlimant, "Graph neural networks in particle physics," *Machine Learning: Science and Technology*, 2020.
[12] G. Panagopoulos *et al.*, "Transfer graph neural networks for pandemic forecasting," *AAAI*, 2021.
[13] M. J. Horry *et al.*, "Covid-19 detection through transfer learning using multimodal imaging data," *IEEE Access*, 2020.
[14] K. Hsieh *et al.*, "Drug repurposing for covid-19 using graph neural network and harmonizing multiple evidence," *Scientific reports*, 2021.
[15] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui, "Towards out-of-distribution generalization: A survey," *arXiv preprint arXiv:2108.13624*, 2021.
[16] J. Wang *et al.*, "Generalizing to unseen domains: A survey on domain generalization," *IJCAI*, 2021.
[17] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE TPAMI*, 2022.
[18] K. Ding, Z. Xu, H. Tong, and H. Liu, "Data augmentation for deep graph learning: A survey," *arXiv preprint arXiv:2202.08235*, 2022.
[19] T. Zhao, G. Liu, S. Günnemann, and M. Jiang, "Graph data augmentation for graph machine learning: A survey," *arXiv preprint arXiv:2202.08871*, 2022.
[20] Y. Liu *et al.*, "Graph self-supervised learning: A survey," *IEEE TKDE*, 2022.
[21] Y. Xie, Z. Xu, J. Zhang, Z. Wang, and S. Ji, "Self-supervised learning of graph neural networks: A unified review," *IEEE TPAMI*, 2022.
[22] L. Sun, Y. Dou, C. Yang, K. Zhang, J. Wang, S. Y. Philip, L. He, and B. Li, "Adversarial attack and defense on graph data: A survey," *IEEE TKDE*, 2022.
[23] L. Chen, J. Li, J. Peng, T. Xie, Z. Cao, K. Xu, X. He, and Z. Zheng, "A survey of adversarial learning on graphs," *arXiv preprint arXiv:2003.05730*, 2020.
[24] T. Zhao, Y. Liu, L. Neves, O. Woodford, M. Jiang, and N. Shah, "Data augmentation for graph neural networks," *AAAI*, 2021.
[25] H. Park *et al.*, "Metropolis-hastings data augmentation for graph neural networks," *NeurIPS*, vol. 34, 2021.
[26] L. Wu, H. Lin, Y. Huang, and S. Z. Li, "Knowledge distillation improves graph structure augmentation for graph neural networks," in *NeurIPS*, 2022.
[27] W. Feng *et al.*, "Graph random neural networks for semi-supervised learning on graphs," *NeurIPS*, 2020.
[28] K. Kong, G. Li, M. Ding, Z. Wu, C. Zhu, B. Ghanem, G. Taylor, and T. Goldstein, "Robust optimization as data augmentation for large-scale graphs," in *CVPR*, 2022.
[29] S. Liu *et al.*, "Local augmentation for graph neural networks," in *ICML*, 2022.
[30] Y. You *et al.*, "Graph contrastive learning with augmentations," *NeurIPS*, 2020.
[31] G. Liu *et al.*, "Graph rationalization with environment-based augmentations," in *KDD*, 2022.
[32] J. Yu, J. Liang, and R. He, "Mind the label shift of augmentation-based graph ood generalization," in *CVPR*, 2023.
[33] Y. Sui *et al.*, "Unleashing the power of graph data augmentation on covariate distribution shift," in *NeurIPS*, 2023.
[34] Y. Zhu, H. Shi, Z. Zhang, and S. Tang, "Mario: Model agnostic recipe for improving ood generalization of graph contrastive learning," in *WWW*, 2024.
[35] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *ICLR*, 2018.
[36] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, "Disentangled graph convolutional networks," in *ICML*, 2019.
[37] Y. Liu, X. Wang, S. Wu, and Z. Xiao, "Independence promoted graph disentangled networks," in *AAAI*, 2020.
[38] Y. Yang, Z. Feng, M. Song, and X. Wang, "Factorizable graph convolutional networks," *NeurIPS*, 2020.
[39] S. Fan, X. Wang, Y. Mo, C. Shi, and J. Tang, "Debiasing graph neural networks via learning disentangled causal substructure," in *NeurIPS*, 2022.

[40] X. Guo, L. Zhao, Z. Qin, L. Wu, A. Shehu, and Y. Ye, "Interpretable deep graph generation with node-edge co-disentanglement," in *KDD*, 2020.
[41] H. Li, X. Wang, Z. Zhang, Z. Yuan, H. Li, and W. Zhu, "Disentangled contrastive learning on graphs," *NeurIPS*, 2021.
[42] H. Li, Z. Zhang, X. Wang, and W. Zhu, "Disentangled graph contrastive learning with independence promotion," *IEEE TKDE*, 2022.
[43] H. Li, X. Wang, Z. Zhang, H. Chen, Z. Zhang, and W. Zhu, "Disentangled graph self-supervised learning for out-of-distribution generalization," in *ICML*, 2024.
[44] S. Fan, X. Wang, C. Shi, P. Cui, and B. Wang, "Generalizing graph neural networks on out-of-distribution graphs," *arXiv preprint arXiv:2111.10657*, 2021.
[45] S. Fan, X. Wang, C. Shi, K. Kuang, N. Liu, and B. Wang, "Debiased graph neural networks with agnostic label selection bias," *TNNLS*, 2022.
[46] Y. Sui, X. Wang, J. Wu, M. Lin, X. He, and T.-S. Chua, "Causal attention for interpretable and generalizable graph classification," *KDD*, 2022.
[47] Y. Wu *et al.*, "Deconfounding to explanation evaluation in graph neural networks," *arXiv preprint arXiv:2201.08802*, 2022.
[48] Y. Chen *et al.*, "Learning causally invariant representations for out-of-distribution generalization on graphs," in *NeurIPS*, 2022.
[49] T. Yao, Y. Chen, Z. Chen, K. Hu, Z. Shen, and K. Zhang, "Empowering graph invariance learning with deep spurious infomax," *ICML*, 2024.
[50] W. An *et al.*, "Causal subgraphs and information bottlenecks: Redefining ood robustness in graph neural networks," in *ECCV*, 2024.
[51] X. Li, S. Gui, Y. Luo, and S. Ji, "Graph structure extrapolation for out-of-distribution generalization," in *ICML*, 2024.
[52] Q. Wu, F. Nie, C. Yang, T. Bao, and J. Yan, "Graph out-of-distribution generalization via causal intervention," in *WWW*, 2024.
[53] B. Bevilacqua, Y. Zhou, and B. Ribeiro, "Size-invariant graph representations for graph classification extrapolations," in *ICML*, 2021.
[54] Y. Zhou, G. Kutyniok, and B. Ribeiro, "Ood link prediction generalization capabilities of message-passing gnns in larger test graphs," *NeurIPS*, 2022.
[55] T. Zhao, G. Liu, D. Wang, W. Yu, and M. Jiang, "Learning from counterfactual links for link prediction," in *ICML*. PMLR, 2022, pp. 26 911–26 926.
[56] W. Lin, H. Lan, and B. Li, "Generative causal explanations for graph neural networks," in *ICML*, 2021.
[57] H. Li, Z. Zhang, X. Wang, and W. Zhu, "Learning invariant graph representations under distribution shifts," in *NeurIPS*, 2022.
[58] L. Yue, Q. Liu, Y. Liu, W. Gao, F. Yao, and W. Li, "Cooperative classification and rationalization for graph generalization," in *WWW*, 2024.
[59] Y.-X. Wu, X. Wang, A. Zhang, X. He, and T. seng Chua, "Discovering invariant rationales for graph neural networks," in *ICLR*, 2022.
[60] S. Miao, M. Liu, and P. Li, "Interpretable and generalizable graph learning via stochastic attention mechanism," in *ICML*, 2022.
[61] Y. Sui, J. Sun, S. Wang, Z. Liu, Q. Cui, L. Li, and X. Wang, "A unified invariant learning framework for graph classification," *KDD*, 2025.
[62] H. Li, X. Wang, X. Zhu, W. Wen, and W. Zhu, "Disentangling invariant subgraph via variance contrastive estimation under distribution shifts," in *ICML*, 2025.
[63] Q. Wu, H. Zhang, J. Yan, and D. Wipf, "Handling distribution shifts on graphs: An invariance perspective," in *ICLR*, 2022.
[64] H. Li, Z. Zhang, X. Wang, and W. Zhu, "Invariant node representation learning under distribution shifts with multiple latent environments," *ACM TOIS*, 2023.
[65] Y. Liu *et al.*, "Flood: A flexible invariant learning framework for out-of-distribution generalization on graphs," in *KDD*, 2023.
[66] S. Wu, K. Cao, B. Ribeiro, J. Y. Zou, and J. Leskovec, "Graphmetro: Mitigating complex graph distribution shifts via mixture of aligned experts," *NeurIPS*, 2024.
[67] Z. Zhang, X. Wang, Z. Zhang, H. Li, Z. Qin, and W. Zhu, "Dynamic graph neural networks under spatio-temporal distribution shift," in *NeurIPS*, 2022.
[68] Z. Zhang *et al.*, "Spectral invariant learning for dynamic graphs under distribution shifts," *NeurIPS*, 2023.
[69] H. Yuan, Q. Sun, X. Fu, Z. Zhang, C. Ji, H. Peng, and J. Li, "Environment-aware dynamic graph learning for out-of-distribution generalization," *NeurIPS*, 2023.
[70] Q. Zhu, N. Ponomareva, J. Han, and B. Perozzi, "Shift-robust gnns: Overcoming the limitations of localized graph training data," *NeurIPS*, 2021.
[71] D. Buffelli, P. Liò, and F. Vandin, "Sizeshiftreg: a regularization method for improving size-generalization in graph neural networks," *NeurIPS*, 2022.
[72] S. Zhang *et al.*, "Stable prediction on graphs with agnostic distribution shift," *arXiv preprint arXiv:2110.03865*, 2021.
[73] M. Wu, S. Pan, X. Zhu, C. Zhou, and L. Pan, "Domain-adversarial graph neural networks for text classification," in *ICDM*, 2019.
[74] A. Sadeghi, M. Ma, B. Li, and G. B. Giannakis, "Distributionally robust semi-supervised learning over graphs," *arXiv preprint arXiv:2110.10582*, 2021.
[75] F. Feng, X. He, J. Tang, and T.-S. Chua, "Graph adversarial training: Dynamically regularizing based on graph structure," *IEEE TKDE*, 2019.
[76] H. Xue, K. Zhou, T. Chen, K. Guo, X. Hu, Y. Chang, and X. Wang, "Cap: Co-adversarial perturbation on weights and features for improving generalization of graph neural networks," *arXiv preprint arXiv:2110.14855*, 2021.
[77] Y. Wu, A. Bojchevski, and H. Huang, "Adversarial weight perturbation improves generalization in graph neural networks," *AAAI*, 2023.
[78] C. Wang, Z. Wang, D. Chen, S. Zhou, Y. Feng, and C. Chen, "Online adversarial distillation for graph neural networks," *arXiv preprint arXiv:2112.13966*, 2021.
[79] W. Hu *et al.*, "Strategies for pre-training graph neural networks," *ICLR*, 2020.
[80] G. Yehudai, E. Fetaya, E. Meirom, G. Chechik, and H. Maron, "From local structures to size generalization in graph neural networks," in *ICML*, 2021.
[81] H. Liu, B. Hu, X. Wang, C. Shi, Z. Zhang, and J. Zhou, "Confidence may cheat: Self-training on graph neural networks under distribution shift," *WWW*, 2022.
[82] S. Li, X. Wang, A. Zhang, Y. Wu, X. He, and T.-S. Chua, "Let invariant rationale discovery inspire graph contrastive learning," in *ICML*, 2022.
[83] G. Chen, J. Zhang, X. Xiao, and Y. Li, "Graphtta: Test time adaptation on graph neural networks," *arXiv preprint arXiv:2208.09126*, 2022.
[84] Y. Wang, C. Li, W. Jin, R. Li, J. Zhao, J. Tang, and X. Xie, "Test-time training for graph neural networks," *arXiv preprint arXiv:2210.08813*, 2022.
[85] J. Zhang, Y. Wang, X. Yang, and E. Zhu, "A fully test-time training framework for semi-supervised node classification on out-of-distribution graphs," *TKDD*, 2024.
[86] J. Wang *et al.*, "Generalizing to unseen domains: A survey on domain generalization," *IEEE TKDE*, 2022.
[87] J. Li, Z. Yu, Z. Du, L. Zhu, and H. T. Shen, "A comprehensive survey on source-free domain adaptation," *IEEE TPAMI*, 2024.
[88] X. Wu *et al.*, "Out-of-distribution generalization in time series: A survey," *arXiv preprint arXiv:2503.13868*, 2025.
[89] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," 1970.
[90] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, 2001.
[91] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *NeurIPS*, vol. 32, 2019.
[92] M. Ding *et al.*, "A closer look at distribution shifts and out-of-distribution generalization on graphs," *NeurIPS Workshop*, 2021.
[93] V. Verma *et al.*, "Manifold mixup: Better representations by interpolating hidden states," in *ICML*, 2019.
[94] L. Zhang, Z. Deng, K. Kawaguchi, A. Ghorbani, and J. Zou, "How does mixup help with robustness and generalization?" in *ICLR*, 2021.
[95] H. Guo, "Nonlinear mixup: Out-of-manifold data augmentation for text classification," in *AAAI*, 2020.
[96] V. Verma *et al.*, "Graphmix: Improved training of gnns for semi-supervised learning," in *AAAI*, 2021.
[97] Y. Wang, W. Wang, Y. Liang, Y. Cai, and B. Hooi, "Mixup for node and graph classification," in *WWW*, 2021.
[98] L. Wu, J. Xia, Z. Gao, H. Lin, C. Tan, and S. Z. Li, "Graphmixup: Improving class-imbalanced node classification by reinforcement mixup and self-supervised context prediction," in *ECML-PKDD*, 2022.
[99] H. Guo and Y. Mao, "Intrusion-free graph mixup," *arXiv preprint arXiv:2110.09344*, 2021.
[100] Y. Wang, W. Wang, Y. Liang, Y. Cai, J. Liu, and B. Hooi, "Nodeaug: Semi-supervised node classification with data augmentation," in *KDD*, 2020.
[101] X. Han, Z. Jiang, N. Liu, and X. Hu, "G-mixup: Graph data augmentation for graph classification," *ICML*, 2022.
[102] B. Lu, Z. Zhao, X. Gan, S. Liang, L. Fu, X. Wang, and C. Zhou, "Graph out-of-distribution generalization with controllable data augmentation," *TKDE*, 2024.
[103] M. L. Montero, C. J. Ludwig, R. P. Costa, G. Malhotra, and J. Bowers, "The role of disentanglement in generalisation," in *ICLR*, 2020.
[104] A. Dittadi *et al.*, "On the transfer of disentangled representations in realistic settings," *arXiv preprint arXiv:2010.14407*, 2020.
[105] A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola, "A kernel statistical test of independence," *NeurIPS*, 2007.
[106] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, 2020.
[107] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive representation learning: A framework and review," *IEEE Access*, 2020.
[108] B. Schölkopf *et al.*, "Toward causal representation learning," *Proceedings of the IEEE*, 2021.
[109] K. Kuang *et al.*, "Stable prediction across unknown environments," in *KDD*, 2018.
[110] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *NeurIPS*, 2007.
[111] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.
[112] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," *NeurIPS*, 2018.
[113] M. Glymour, J. Pearl, and N. P. Jewell, *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.
[114] A. Balke and J. Pearl, "Probabilistic evaluation of counterfactual queries," *AAAI*, 1994.
[115] J. Pearl and D. Mackenzie, *The book of why: the new science of cause and effect*. Basic books, 2018.
[116] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica: journal of the Econometric Society*, 1969.
[117] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.
[118] S. Chang *et al.*, "Invariant rationalization," in *ICML*, 2020.
[119] K. Ahuja *et al.*, "Invariance principle meets information bottleneck for out-of-distribution generalization," *NeurIPS*, 2021.
[120] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *IEEE information theory workshop*, 2015.
[121] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[122] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, 2015.
[123] W. Zellinger *et al.*, "Robust unsupervised domain adaptation for neural networks via moment alignment," *Information Sciences*, 2019.
[124] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *ICML*, 2019.
[125] A. Gretton *et al.*, "Covariate shift by kernel mean matching," *Dataset shift in machine learning*, 2009.
[126] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *JMLR*, 2016.
[127] H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review," *arXiv preprint arXiv:1908.05659*, 2019.

[128] Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker, "Domain generalization via model-agnostic learning of semantic features," *NeurIPS*, vol. 32, 2019.
[129] D. Mahajan, S. Tople, and A. Sharma, "Domain generalization using causal matching," in *ICML*, 2021.
[130] M. Zhang *et al.*, "Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations," *arXiv preprint arXiv:2203.01517*, 2022.
[131] Y. You, T. Chen, Z. Wang, and Y. Shen, "When does self-supervision help graph convolutional networks?" in *ICML*, 2020.
[132] K. Xu *et al.*, "How neural networks extrapolate: From feedforward to graph neural networks," *ICLR*, 2021.
[133] B. Knyazev, G. W. Taylor, and M. Amer, "Understanding attention and generalization in graph neural networks," *NeurIPS*, vol. 32, 2019.
[134] S. Gui, X. Li, L. Wang, and S. Ji, "GOOD: A graph out-of-distribution benchmark," in *NeurIPS Datasets and Benchmarks Track*, 2022.
[135] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," *IEEE TPAMI*, 2022.
[136] Y. Ji *et al.*, "Drugood: Out-of-distribution (ood) dataset curator and benchmark for ai-aided drug discovery," *arXiv preprint arXiv:2201.09637*, 2022.
[137] B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-scale attributed node embedding," *Journal of Complex Networks*, 2021.
[138] A. Pareja *et al.*, "Evolvegcn: Evolving graph convolutional networks for dynamic graphs," in *AAAI*, 2020.
[139] F. Scarselli, A. C. Tsoi, and M. Hagenbuchner, "The vapnik–chervonenkis dimension of graph and recursive neural networks," *Neural Networks*, 2018.
[140] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," in *Measures of complexity*, 2015.
[141] S. Verma and Z.-L. Zhang, "Stability and generalization of graph convolutional neural networks," in *KDD*, 2019.
[142] O. Bousquet and A. Elisseeff, "Stability and generalization," *JMLR*, 2002.
[143] V. Garg, S. Jegelka, and T. Jaakkola, "Generalization and representational limits of graph neural networks," in *ICML*, 2020.
[144] S. Lv, "Generalization bounds for graph convolutional neural networks via rademacher complexity," *arXiv preprint arXiv:2102.10234*, 2021.
[145] R. Liao, R. Urtasun, and R. Zemel, "A pac-bayesian approach to generalization bounds for graph neural networks," *ICLR*, 2021.
[146] J. Ma, J. Deng, and Q. Mei, "Subgroup generalization and fairness of graph neural networks," *NeurIPS*, vol. 34, 2021.
[147] S. S. Du *et al.*, "Graph neural tangent kernel: Fusing graph neural networks with graph kernels," *NeurIPS*, 2019.
[148] S. Zhang, M. Wang, S. Liu, P.-Y. Chen, and J. Xiong, "Fast learning of graph neural networks with guaranteed generalizability: one-hidden-layer case," in *ICML*, 2020.
[149] A. Baranwal *et al.*, "Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization," *ICML*, 2021.
[150] S. Maskey, R. Levie, Y. Lee, and G. Kutyniok, "Generalization analysis of message passing neural networks on large random graphs," in *NeurIPS*, 2022.
[151] W. Mao *et al.*, "Invariant graph learning meets information bottleneck for out-of-distribution generalization," *FCS*, 2025.
[152] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," *NeurIPS*, 2019.
[153] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.
[154] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE TPAMI*, 2012.
[155] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019.
[156] D. Mendez *et al.*, "Chembl: towards direct deposition of bioassay data," *Nucleic acids research*, 2019.
[157] P. Veličković, R. Ying, M. Padovano, R. Hadsell, and C. Blundell, "Neural execution of graph algorithms," *ICLR*, 2020.
[158] J. Ko, T. Kwon, K. Shin, and J. Lee, "Learning to pool in graph neural networks for extrapolation," *arXiv preprint arXiv:2106.06210*, 2021.
[159] Y. Wang, Y. Ma, W. Jin, C. Li, C. C. Aggarwal, and J. Tang, "Customized graph neural networks," *arXiv preprint arXiv:2005.12386*, 2020.
[160] Y. Qin, X. Wang, Z. Zhang, P. Xie, and W. Zhu, "Graph neural architecture search under distribution shifts," in *ICML*, 2022.
[161] L. Galke, B. Franke, T. Zielke, and A. Scherp, "Lifelong learning of graph neural networks for open-world node classification," in *IJCNN*, 2021.
[162] Z. Zhang *et al.*, "Disentangled continual graph neural architecture search with invariant modular supernet," in *ICML*, 2024.
[163] W. Jin, T. Zhao, J. Ding, Y. Liu, J. Tang, and N. Shah, "Empowering graph representation learning with test-time graph transformation," *ICLR*, 2023.
[164] X. Zheng, D. Song, Q. Wen, B. Du, and S. Pan, "Online gnn evaluation under test-time graph distribution shifts," *ICLR*, 2024.
[165] K. Han, B. Lakshminarayanan, and J. Liu, "Reliable graph neural networks for drug discovery under distributional shift," *NeurIPS Workshop*, 2021.
[166] N. Yang, K. Zeng, Q. Wu, X. Jia, and J. Yan, "Learning substructure invariance for out-of-distribution molecular representations," in *NeurIPS*, 2022.
[167] K. Sinha, S. Sodhani, J. Pineau, and W. L. Hamilton, "Evaluating logical generalization in graph neural networks," *ICML Workshop*, 2020.
[168] R. Li *et al.*, "How does knowledge graph embedding extrapolate to unseen data: a semantic evidence view," *AAAI*, 2022.
[169] K. Li *et al.*, "A critical examination of robustness and generalizability of machine learning prediction of materials properties," *npj Computational Materials*, 2023.
[170] S. H. Gheshlaghi, M. Aryal *et al.*, "Explainability-based graph augmentation for out-of-distribution robustness in digital pathology," *KBS*, 2025.

Haoyang Li is currently a Postdoctoral Associate at Weill Cornell Medicine of Cornell University. He received his Ph.D. from the Department of Computer Science and Technology of Tsinghua University in 2023. He received his B.E. from the Department of Computer Science and Technology of Tsinghua University in 2018. His research interests are mainly in machine learning on graphs and out-of-distribution generalization. He has published high-quality papers in prestigious journals and conferences, e.g., IEEE TKDE, ACM TOIS, ICML, NeurIPS, ICLR, KDD, WWW, etc.

Xin Wang is currently an Associate Professor at the Department of Computer Science and Technology, Tsinghua University. He got both his Ph.D. and B.E degrees in Computer Science and Technology from Zhejiang University, China. He also holds a Ph.D. degree in Computing Science from Simon Fraser University, Canada. His research interests include multimedia intelligence, machine learning and its applications in multimedia big data analysis. He has published over 150 high-quality research papers in top journals and conferences including IEEE TPAMI, IEEE TKDE, ACM TOIS, ICML, NeurIPS, ACM KDD, ACM Web Conference, ACM SIGIR and ACM Multimedia etc., winning three best paper awards. He is the recipient of 2020 ACM China Rising Star Award, 2022 IEEE TCMC Rising Star Award and 2023 DAMO Academy Young Fellow.

Ziwei Zhang received his Ph.D. from the Department of Computer Science and Technology, Tsinghua University, in 2021. He is currently a postdoc researcher in the Department of Computer Science and Technology at Tsinghua University. His research interests focus on machine learning on graphs, including graph neural network (GNN), network embedding (a.k.a. network representation learning), and automated graph machine learning. He has published over 40 papers in prestigious conferences and journals, including KDD, NeurIPS, ICML, AAAI, IJCAI, and TKDE.

Wenwu Zhu is currently a Professor in the Department of Computer Science and Technology at Tsinghua University. Prior to his current post, he was a Senior Researcher and Research Manager at Microsoft Research Asia. He was the Chief Scientist and Director at Intel Research China from 2004 to 2008. He worked at Bell Labs, New Jersey as Member of Technical Staff during 1996-1999. He received his Ph.D. degree from New York University in 1996. His research interests are in the area of multimedia intelligence, data-driven multimedia networking and cross-media big data computing. He has published over 400 referred papers and is the inventor or co-inventor of over 100 patents. He received eight Best Paper Awards, including ACM Multimedia 2012 and IEEE TCSVT in 2001 and 2019. He served as EiC for IEEE TMM (2017-2019) and IEEE TCSVT (2024-2025). He served in the steering committee for IEEE TMM (2015-2016) and IEEE TMC (2007-2010), respectively. He serves as General Co-Chair for ACM Multimedia 2018 and ACM CIKM 2019, respectively. He is an AAAS Fellow, ACM Fellow, IEEE Fellow, SPIE Fellow, and a member of The Academy of Europe (Academia Europaea).