Invariance Principle Meets Out-of-Distribution Generalization on Graphs

Yongqiang Chen[1] Yonggang Zhang[2] Yatao Bian[3] Han Yang[1] Kaili Ma[1] Binghui Xie[1] Tongliang Liu[4]
Bo Han[2] James Cheng[1]

Abstract

Despite recent success in using the invariance principle for out-of-distribution (OOD) generalization on Euclidean data (e.g., images), studies on graph data are still limited. Different from images, the complex nature of graphs poses unique challenges to adopting the invariance principle. In particular, distribution shifts on graphs can appear in a variety of forms such as attributes and structures, making it difficult to identify the invariance. Moreover, domain or environment partitions, which are often required by OOD methods on Euclidean data, could be highly expensive to obtain for graphs. To bridge this gap, we propose a new framework to capture the invariance of graphs for guaranteed OOD generalization under various distribution shifts. Specifically, we characterize potential distribution shifts on graphs with causal models, concluding that OOD generalization on graphs is achievable when models focus *only* on subgraphs containing the most information about the causes of labels. Accordingly, we propose an information-theoretic objective to extract the desired subgraphs that maximally preserve the invariant intra-class information. Learning with these subgraphs is immune to distribution shifts. Extensive experiments on both synthetic and real-world datasets, including a challenging setting in AI-aided drug discovery, validate the superior OOD generalization ability of our method.

1. Introduction

Graph neural networks (GNNs) have gained great success in tasks involving relational information (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018; Xu et al., 2018; 2019). However, it assumes that the training and test graphs are drawn from the same distribution, which is often violated in reality (Hu et al., 2020; Koh et al., 2021; Huang et al., 2021; Ji et al., 2022). The mismatch between training and test distributions, i.e., *distribution shifts*, introduced by some underlying environmental factors related to data collection or processing, could seriously degrade the performance of deployed models (Beery et al., 2018; DeGrave et al., 2021). Such *out-of-distribution* (OOD) generalization failures become the major roadblock for practical applications of graph neural networks (Ji et al., 2022).

Meanwhile, enabling OOD generalization on Euclidean data has received surging attention and several solutions were proposed (Arjovsky et al., 2019; Sagawa* et al., 2020; Bengio et al., 2020; Krueger et al., 2021; Creager et al., 2021; Koyama & Yamaguchi, 2020; Ahuja et al., 2021). In particular, the invariance principle from causality is at the heart of those works (Peters et al., 2016). The principle leverages the Independent Causal Mechanism (ICM) assumption (Pearl, 2009; Peters et al., 2017) and implies that, predictions that focus only on the causes of the label can stay invariant to a large class of distribution shifts (Peters et al., 2016).

Despite the success of the invariance principle on Euclidean data, the complex nature of graphs raises several new challenges that prohibit direct adoptions of the principle. First, distribution shifts on graphs can happen at both attribute-level and structure-level, and be observed in multiple forms (Wu et al., 2022a), where each shift can spuriously correlate with labels in different modes (Nagarajan et al., 2021; Ahuja et al., 2021). The entangled distribution shifts make it more difficult to identify the invariance. Second, OOD methods developed on Euclidean data often require additional environment (or domain) labels for distinguishing the sources of distribution shifts (Arjovsky et al., 2019). However, the environment labels could be highly expensive to obtain and often unavailable for graphs, as collecting the labels usually requires expert knowledge due to the abstraction of graphs (Hu et al., 2020). These challenges render the problem studied in this paper even more challenging:

How could one generalize the invariance principle to enable OOD generalization on graphs?

To solve the above problem, we propose a new framework, **G**raph **O**ut-**Of**-Distribution **G**eneralization (GOOD), to

[1] The Chinese University of Hong Kong [2] Hong Kong Baptist University [3] Tencent AI Lab [4] The University of Sydney. Correspondence to: Yongqiang Chen <yqchen@cse.cuhk.edu.hk>.

Published at the ICML 2022 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA. Copyright 2022 by the author(s).

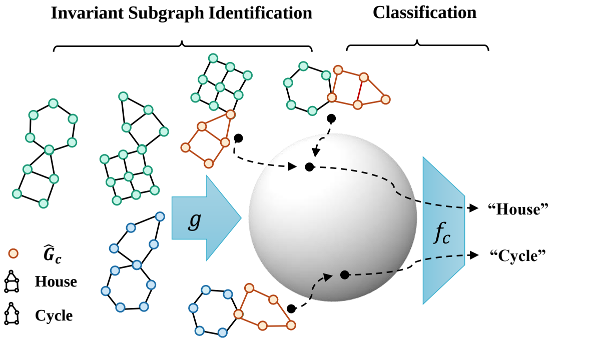


Figure 1. GOOD framework: GNNs need to classify graphs based on the specific motif ("House" or "Cycle"). The featurizer g will extract an (orange colored) subgraph $\hat{G}_c$ from each input for the classifier f_c to predict the label. The objective of g is implemented in a contrastive strategy s.t. the distribution of $\hat{G}_c$ at the latent sphere will be optimized to maximize intra-class information hence predictions will be invariant to distribution shifts.

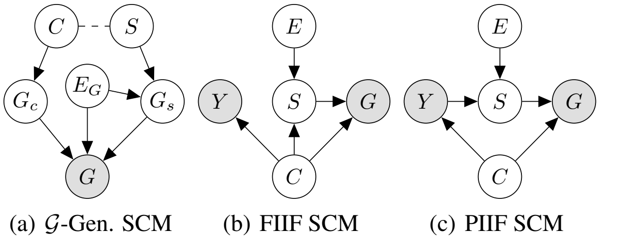
enable guaranteed OOD generalization on graphs under different distribution shifts. Specifically, we build three Structural Causal Models (SCMs) (Pearl, 2009) to characterize the distribution shifts that could happen on graphs (Sec. 2.1). Then, we generalize the invariance principle to graphs for OOD generalization: GNNs are invariant to distribution shifts if they focus only on a invariant and critical subgraph G_c that contains the most of the information in G about the underlying causes of the label. Thus, we can achieve OOD generalization on graphs with two processes: invariant subgraph identification and label prediction. Accordingly, shown as Fig. 1, we implement a prototypical algorithm that decomposes a GNN into: a) a featurizer g for identifying the invariant subgraph G_c from G; b) a classifier f_c for making predictions based on G_c. To identify the desired G_c, we derive an information-theoretic objective for g to extract subgraphs that maximally preserve the invariant intra-class information. We show that this approach can provably identify the underlying G_c (Sec. 3).

Experiments on 16 synthetic and realistic datasets with various distribution shifts, including a challenging setting from AI-aided drug discovery (Ji et al., 2022), show that GOOD can significantly outperform all of existing methods, demonstrating its promising OOD generalization ability (Sec. 4).

To our best knowledge, there is no existing work that could handle more comprehensive graph distribution shifts than GOOD while also with OOD generalization guarantees. Discussions on related works are deferred to Appendix C.2.

2. Graph OOD through the Lens of Causality

Problem Setup. In this work, we focus on OOD generalization in graph classification. Specifically, we are given a set of graph datasets $\mathcal{D} = \{\mathcal{D}^e\}_e$ collected from multiple



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

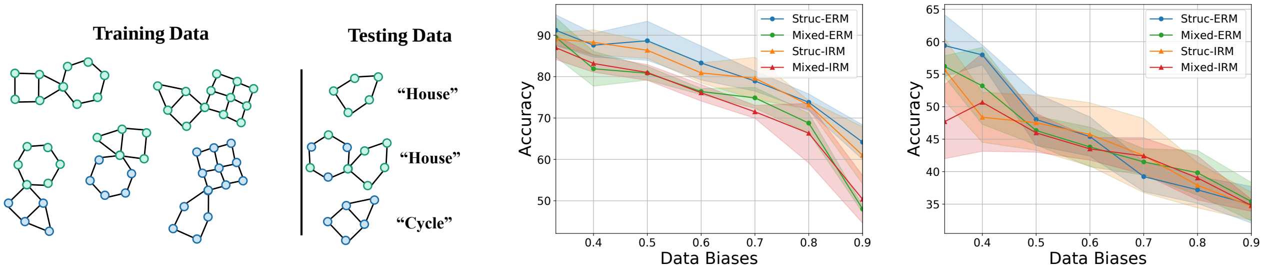
Figure 2. SCMs on graph distribution shifts.

environments $\mathcal{E}_{\text{all}}$. Samples $(G_i^e, Y_i^e) \in \mathcal{D}^e$ from the same environment are considered as drawn independently from an identical distribution $\mathbb{P}^e$. A GNN $\rho \circ h$ generically has an encoder $h : \mathcal{G} \to \mathbb{R}^h$ that learns a meaningful representation h_G for each graph G to help predict the label $\hat{Y}_G = \rho(h_G)$ with a downstream classifier $\rho : \mathbb{R}^h \to \mathcal{Y}$. The goal of OOD generalization on graphs is to train a GNN $\rho \circ h$ with data from training environments $\mathcal{D}_{\text{tr}} = \{\mathcal{D}^e\}_{e \in \mathcal{E}_{\text{tr}} \subseteq \mathcal{E}_{\text{all}}}$ that generalizes well to all (unseen) environments, i.e., to minimize $\max_{e \in \mathcal{E}_{\text{all}}} R^e$, where R^e is the empirical risk of $\rho \circ h$ under environment e (Vapnik, 1991; Arjovsky et al., 2019). We leave more background details in Appendix C.1.

2.1. Graph Generation Processes

It is known that OOD generalization is impossible without assumptions on the environments $\mathcal{E}_{\text{all}}$ (Pearl, 2009; Ahuja et al., 2021). In the next, we brief the assumptions on graph generation with SCMs and leave full details to Appendix A. We take a latent-variable model perspective on the graph generation process and assume that the graph is generated through a mapping $f_{\text{gen}} : \mathcal{Z} \to \mathcal{G}$, where $\mathcal{Z} \subseteq \mathbb{R}^n$ is the latent space and $\mathcal{G} = \cup_{N=1}^{\infty} \{0, 1\}^N \times \mathbb{R}^{N \times d}$ is the graph space. Let E denote environments. We partition the latent variable from $\mathcal{Z}$ into an invariant part $C \in \mathcal{C} = \mathbb{R}^{n_c}$ and a varying part $S \in \mathcal{S} = \mathbb{R}^{n_s}$, s.t., $n = n_c + n_s$, according to whether they are affected by E (Kügelgen et al., 2021). Similarly in images, C and S can represent content and style while E can refer to the locations where the images are taken (Zhang et al., 2021; Kügelgen et al., 2021). C and S further control the generation of the observed graphs and can have multiple types of interactions (Ahuja et al., 2021).

Graph generation model. The SCM for graph generation is given as Fig. 2(a). f_{gen} is decomposed into three processes to control the generation of G_c, G_s, and G, respectively. Among them, G_c inherits the invariant information of C that would not be affected by the interventions (or changes) of E (Pearl, 2009). For example, certain properties of a molecule can usually be described by a sub-molecule, or a functional group, which is invariant across different species, or assays (Bohacek et al., 1996; Sterling & Irwin, 2015; Ji et al., 2022). In contrast, the generation of G_s and G will be affected by environment $E_G \subseteq E$. Thus, graphs collected from different environments (or domains) can have different structure-level (e.g., graph sizes (Bevilacqua et al., 2021)) as



(a) Failure cases for existing methods.

(b) Structure and attribute shifts.

(c) Mixed with graph size shifts.

Figure 3. Failures modes of OOD generalization on graphs: (a) GNNs need to classify whether the graph contains a "house" or "cycle" motif, where colors represent node features. However, distribution shifts in the training data happen at both structure-level (from left to right: "house" mostly co-occur with a hexagon), attribute-level (from upper to lower: nodes are mostly colored green if the graph contains a "house", or blue otherwise), and graph sizes, making GNNs hard to capture the invariance. Thus, *ERM can fail* for taking the shortcuts and predicting graphs that have a hexagon or have nodes mostly colored green as "house". *IRM can fail* as the test data are not sufficiently supported by the training data. (b) GNNs optimized with neither ERM nor IRM can generalize to OOD graphs under structure-level shifts (Struc-) or mixed with feature shifts (Mixed-). (c) When more complex shifts presented, GNNs can fail more seriously.

well as feature-level statistics (e.g., homophily (McPherson et al., 2001; Chen et al., 2022; Wu et al., 2022a)).

Interactions at latent space. Following previous works (Arjovsky et al., 2019; Ahuja et al., 2021), we categorize the interactions between C and S into Fully Informative Invariant Features (FIIF, Fig. 2(b)) and Partially Informative Invariant Features (PIIF, Fig. 2(c)), depending on whether C is fully informative about Y, i.e., $(S, E) \perp\!\!\!\perp Y|C$. In the two SCMs, S is directly controlled by C in FIIF and indirectly controlled by C through Y in PIIF, which can exhibit different behaviors in the observed distribution shifts. In practice, performances of OOD algorithms can degrade dramatically if one of FIIF or PIIF is excluded (Aubin et al., 2021; Nagarajan et al., 2021). This issue can be more serious in graphs, since different distribution shifts can have different interaction modes at the latent space. Moreover, $C \to Y$ indicates the labelling process, which assigns labels Y for the corresponding G merely based on C. Consequently, $\mathcal{C}$ is better clustered than $\mathcal{S}$ when given Y (Burshtein et al., 1992; Chapelle et al., 2006; Schölkopf, 2019; Schölkopf et al., 2021), which also serves as the necessary separation assumption for a classification task (Muller et al., 2001; Chen et al., 2005; Mika et al., 1999), i.e., $H(C|Y) \leq H(S|Y)$.

2.2. Challenges of OOD Generalization on Graphs

Built upon the graph generation process, can existing methods produce a desired invariant GNN model? Using the BAMotif task (Luo et al., 2020) as Fig. 3, we find that, neither ERM nor IRM, or more expressive GNN architectures can help improve the OOD generalization ability of GNNs. The main reasons are: a) Distribution shifts on graphs are more complicated where different types of spurious correlations can be entangled via different graph properties; b) Environment labels are usually not available due to the abstraction of graphs. More results are given in Appendix D.

3. Invariance Principle for Graph OOD

Aiming to bridge the gap, we propose GOOD: **G**raph **O**ut-**O**f-**D**istribution Generalization, to generalize and instantiate the invariance principle on graphs. Full details and theoretical analysis are deferred to Appendix B.

Invariance for OOD generalization on graphs. According to the ICM assumption (Peters et al., 2017), the labeling process $C \to Y$ in Fig. 2 is not informed nor influenced by other processes, implying that the conditional distribution $P(Y|C)$ remains invariant to the interventions on the environment latent variable E (Pearl, 2009). Consequently, for a GNN with a permutation invariant encoder $h : \mathcal{G} \to \mathbb{R}^h$ and a downstream classifier $\rho : \mathbb{R}^h \to \mathcal{Y}$, if h can recover the information of C from G in the learned graph representations, then the learning of ρ resembles traditional ERM (Vapnik, 1991) and can stay invariant to distribution shifts.

Causal algorithmic alignment. To enable a GNN to learn to extract the information about C from G, we propose to *explicitly aligns with* the two causal mechanisms during the graph generation, i.e., $C \to G$ and $(G_s, E_G, G_c) \to G$, motivated by Xu et al. (2020). Specifically, we realize the alignment by decomposing a GNN into two sub-components: a) a featurizer GNN $g : \mathcal{G} \to \mathcal{G}_c$ aiming to identify the desired G_c; b) a classifier GNN $f_c : \mathcal{G}_c \to \mathcal{Y}$ that predicts the label Y based on the estimated G_c, where $\mathcal{G}_c$ refers to the space of subgraphs of G. Formally, the learning objectives of f_c and g can be formulated as:

$$\min_{f_c,\, g} \ R(f_c(\hat{G}_c)), \text{ s.t. } \hat{G}_c \perp\!\!\!\perp E,\ \hat{G}_c = g(G), \qquad (1)$$

where $R(f_c(\hat{G}_c))$ is the empirical risk of f_c that takes $\hat{G}_c$ as inputs to predict label Y for G, and $\hat{G}_c$ is the intermediate subgraph containing information about C and hence needs to be independent of E. Moreover, the extracted G_c can either share the same graph space with input G or has its own

Table 1. OOD generalization performance on structure and mixed shifts for synthetic graphs.

	SPMotif-Struc[†]			SPMotif-Mixed[†]			
	bias=0.33	bias=0.60	bias=0.90	bias=0.33	bias=0.60	bias=0.90	Avg
ERM	59.49 (3.50)	55.48 (4.84)	49.64 (4.63)	58.18 (4.30)	49.29 (8.17)	41.36 (3.29)	52.24
ASAP	64.87 (13.8)	64.85 (10.6)	**57.29 (14.5)**	66.88 (15.0)	59.78 (6.78)	50.45 (4.90)	60.69
DIR	58.73 (11.9)	48.72 (14.8)	41.90 (9.39)	67.28 (4.06)	51.66 (14.1)	38.58 (5.88)	51.14
IRM	57.15 (3.98)	61.74 (1.32)	45.68 (4.88)	58.20 (1.97)	49.29 (3.67)	40.73 (1.93)	52.13
V-Rex	54.64 (3.05)	53.60 (3.74)	48.86 (9.69)	57.82 (5.93)	48.25 (2.79)	43.27 (1.32)	51.07
EIIL	56.48 (2.56)	60.07 (4.47)	55.79 (6.54)	53.91 (3.15)	48.41 (5.53)	41.75 (4.97)	52.73
IB-IRM	58.30 (6.37)	54.37 (7.35)	45.14 (4.07)	57.70 (2.11)	50.83 (1.51)	40.27 (3.68)	51.10
CNC	70.44 (2.55)	**66.79 (9.42)**	50.25 (10.7)	65.75 (4.35)	59.27 (5.29)	41.58 (1.90)	59.01
GOODv1	**71.07 (3.60)**	63.23 (9.61)	51.78 (7.29)	**74.35 (1.85)**	**64.54 (8.19)**	49.01 (9.92)	**62.33**
GOODv2	**77.33 (9.13)**	**69.29 (3.06)**	**63.41 (7.38)**	**72.42 (4.80)**	**70.83 (7.54)**	**54.25 (5.38)**	**67.92**

[†]Higher accuracy and lower variance indicate better OOD generalization ability.

space with latent node and edge features, depending on the specific graph generation process. In practice, interpretable GNN architectures (Yuan et al., 2020) are compatible with GOOD, hence can serve as practical choices for GOOD. Details are given in Appendix F.

Optimization objective. To ensure the independence constraint $\hat{G}_c \perp\!\!\!\perp E$ under the *absence* of E, we translate other properties of G_c from SCMs in Sec. 2.1 into differentiable and equivalent conditions. In a simplistic setting where all the invariant subgraphs G_c have the same size s_c, i.e., $|G_c| = s_c$. We derive the first objective (GOODv1):

$$\max_{f_c,g} I(\hat{G}_c;Y), \text{s.t.} \hat{G}_c \in \underset{\hat{G}_c=g(G),|\hat{G}_c|\le s_c}{\arg\max} I(\hat{G}_c;\tilde{G}_c|Y), \tag{2}$$

where $\hat{G}_c = g(G), \tilde{G}_c = g(\tilde{G})$ and $\tilde{G} \sim \mathbb{P}(G|Y)$, i.e., $\tilde{G}$ and G have the same label. In Theorem B.2, we discuss why Eq. 2 is equivalent to Eq. 1. Although being effective, Eq. 2 requires a strong assumption about the size of G_c. However, the size of G_c is usually unknown or changes for different Cs. In this circumstance, maximizing Eq. 2 without additional constraints would lead to the presence of subgraphs of G_s in $\hat{G}_c$. For instance, $\hat{G}_c = G$ is a trivial solution to Eq. 2 when $s_c = \infty$.

To circumvent this limitation, we further resort to the properties of G_s and obtain a new objective GOODv2 as follows

$$\begin{aligned} &\max_{f_c,g} \ I(\hat{G}_c;Y) + I(\hat{G}_s;Y), \\ &\text{s.t. } \hat{G}_c \in \underset{\hat{G}_c=g(G),\tilde{G}_c=g(\tilde{G})}{\arg\max} I(\hat{G}_c;\tilde{G}_c|Y), \\ &I(\hat{G}_s;Y) \le I(\hat{G}_c;Y), \ \hat{G}_s = G - g(G), \end{aligned} \tag{3}$$

where $\hat{G}_c, \tilde{G}_c$ and $\tilde{G}$ are the same as Eq. 2. We deffer the theoretical analysis and implementation details to Appendix B.

4. Empirical Studies

We conduct extensive experiments with 16 datasets to verify the effectiveness of GOOD. We give analysis on the synthetic datasets and more details to Appendix G.

Datasets. We use the SPMotif datasets from DIR (Wu et al., 2022c) where artificial structural shifts and graph size shifts are nested (SPMotif-Struc). Besides, we construct a harder version mixed with attribute shifts (SPMotif-Mixed).

Baselines and our methods. Besides ERM, we also compare with SOTA interpretable GNNs, GIB (Yu et al., 2021), ASAP Pooling (Ranjan et al., 2020), and DIR (Wu et al., 2022c), to validate the effectiveness of the optimization objective in GOOD. To validate the effectiveness of the decomposition in GOOD, we compare GOOD with SOTA OOD objectives including IRM (Arjovsky et al., 2019), v-Rex (Krueger et al., 2021) and IB-IRM (Ahuja et al., 2021), EIIL (Creager et al., 2021) and CNC (Zhang et al., 2022). More comparison details are deferred to Appendix G.3.

Evaluation. We report the mean and standard deviation of classification accuracy for all datasets from multiple times.

OOD performance on structure and mixed shifts. In Table 1, we report the test accuracy of each method, where we omit GIB due to its poor convergence. Different biases indicate different strengths of the distribution shifts. Although the training accuracy of most methods converge to more than 99%, the test accuracy decreases dramatically as the bias increases and as more distribution shifts are mixed, which concurs with our discussions in Sec. 2.2 and Appendix D. Due to the simplicity of the task as well as the relatively high support overlap between training and test distributions, interpretable GNNs and OOD objectives can improve certain OOD performance, while they can have *high variance* since they donot have OOD generalization guarantees. In contrast, GOODv1 and GOODv2 outperform all of the baselines by a significant margin up to 10% with *lower variance*, which demonstrates the effectiveness and excellent OOD generalization ability of GOOD. More analysis and results on real-world datasets are given in Appendix G.

Conclusion.

We studied the OOD generalization on graphs via graph classification, and propose a new solution GOOD through the lens of causality. By modeling potential distribution shifts on graphs with SCMs, we generalized and instantiated the invariance principle to graphs, which was shown to have promising theoretical and empirical OOD generalization ability under a variety of distribution shifts.

Acknowledgements

We thank the reviewers for their valuable comments. This work was supported by GRF 14208318 from the RGC of HKSAR and CUHK direct grant 4055146.

References

Ahmad, I. and Lin, P.-E. A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.). *IEEE Transactions on Information Theory*, 22(3):372–375, 1976.

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

Alemi, A. A., Fischer, I., and and, J. V. D. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

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

Aubin, B., Słowik, A., Arjovsky, M., Bottou, L., and Lopez-Paz, D. Linear unit-tests for invariance discovery. *arXiv preprint arXiv:2102.10867*, 2021.

Battaglia, P. W., Pascanu, R., Lai, M., Rezende, D. J., and Kavukcuoglu, K. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, pp. 4502–4510, 2016.

Beery, S., Horn, G. V., and Perona, P. Recognition in terra incognita. In *Computer Vision European Conference, Part XVI*, volume 11220, pp. 472–489, 2018.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International Conference on Machine Learning*, volume 80, pp. 531–540, 10–15 Jul 2018.

Bellot, A. and van der Schaar, M. Generalization and invariances in the presence of unobserved confounding. *arXiv preprint*, arXiv:2007.10653, 2020.

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

Bevilacqua, B., Zhou, Y., and Ribeiro, B. Size-invariant graph representations for graph classification extrapolations. In *International Conference on Machine Learning*, volume 139, pp. 837–851, 18–24 Jul 2021.

Bohacek, R. S., McMartin, C., and Guida, W. C. The art and practice of structure-based drug design: A molecular modeling perspective. *Medicinal Research Reviews*, 16 (1):3–50, 1996.

Burshtein, D., Pietra, V. D., Kanevsky, D., and Nadas, A. Minimum impurity partitions. *The Annals of Statistics*, 20(3):1637–1646, 1992.

Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. S. Invariant rationalization. In *International Conference on Machine Learning*, volume 119, pp. 1448–1458, 2020.

Chapelle, O., Schölkopf, B., and Zien, A. *Semi-Supervised Learning*. The MIT Press, 2006.

Chen, P.-H., Lin, C.-J., and Schölkopf, B. A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry*, 21(2):111–136, 2005.

Chen, Y., Yang, H., Zhang, Y., Ma, K., Liu, T., Han, B., and Cheng, J. Understanding and improving graph injection attack by promoting unnoticeability. In *International Conference on Learning Representations*, 2022.

Chen, Z., Chen, L., Villar, S., and Bruna, J. Can graph neural networks count substructures? In *Advances in Neural Information Processing Systems*, 2020.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pp. 539–546, 2005.

Chuang, C., Torralba, A., and Jegelka, S. Estimating generalization under distribution shifts via domain-invariant representations. In *International Conference on Machine Learning*, volume 119, pp. 1984–1994. PMLR, 2020.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

Creager, E., Jacobsen, J., and Zemel, R. S. Environment inference for invariant learning. In *International Conference on Machine Learning*, volume 139, pp. 2189–2200, 2021.

DeGrave, A. J., Janizek, J. D., and Lee, S. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., and Xu, K. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Annual Meeting of the Association for Computational Linguistics*, pp. 49–54, 2014.

Dou, Q., de Castro, D. C., Kamnitsas, K., and Glocker, B. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pp. 6447–6458, 2019.

Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking graph neural networks. *arXiv preprint*, arXiv:2003.00982, 2020.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. S. Domain-adversarial training of neural networks. *Journal of Mache Learning Research*, 17:59:1–59:35, 2016.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M. E., Schmitz, M., and Zettlemoyer, L. Allennlp: A deep semantic natural language processing platform. *arXiv preprint*, arXiv:1803.07640, 2018.

Geirhos, R., Jacobsen, J., Michaelis, C., Zemel, R. S., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2 (11):665–673, 2020.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

Han, K., Lakshminarayanan, B., and Liu, J. Z. Reliable graph neural networks for drug discovery under distributional shift. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020.

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y. H., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, volume 37, pp. 448–456, 2015.

Ji, Y., Zhang, L., Wu, J., Wu, B., Huang, L.-K., Xu, T., Rong, Y., Li, L., Ren, J., Xue, D., Lai, H., Xu, S., Feng, J., Liu, W., Luo, P., Zhou, S., Huang, J., Zhao, P., and Bian, Y. DrugOOD: Out-of-Distribution (OOD) Dataset Curator and Benchmark for AI-aided Drug Discovery – A Focus on Affinity Prediction Problems with Noise Annotations. *arXiv preprint*, arXiv:2201.09637, 2022.

Kandasamy, K., Krishnamurthy, A., Poczos, B., Wasserman, L., and robins, j. m. Nonparametric von mises estimators for entropies, divergences and mutual informations. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv preprint*, arXiv:1611.07308, 2016.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Knyazev, B., Taylor, G. W., and Amer, M. R. Understanding attention and generalization in graph neural networks. In *Advances in Neural Information Processing Systems*, pp. 4204–4214, 2019.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning,*, volume 139, pp. 5637–5664, 2021.

Koyama, M. and Yamaguchi, S. Out-of-distribution generalization with maximal invariant predictor. *arXiv preprint*, arXiv:2008.01883, 2020.

Krueger, D., Caballero, E., Jacobsen, J., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. C. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, volume 139, pp. 5815–5826, 2021.

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

Li, Q., Han, Z., and Wu, X. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI Conference on Artificial Intelligence*, pp. 3538–3545, 2018a.

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., and Tao, D. Deep domain generalization via conditional invariant adversarial networks. In *European Conference on Computer Vision*, volume 11219, pp. 647–663, 2018b.

Lin, W., Lan, H., and Li, B. Generative causal explanations for graph neural networks. In *International Conference on Machine Learning,*, volume 139, pp. 6666–6679, 2021.

Lin, Y., Zhu, S., and Cui, P. ZIN: when and how to learn invariance by environment inference? *arXiv preprint arXiv:2203.05818*, 2022.

Lovász, L. and Szegedy, B. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, 96(6):933–957, 2006.

Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., and Zhang, X. Parameterized explainer for graph neural network. In *Advances in Neural Information Processing Systems*, 2020.

Luo, Y., Yan, K., and Ji, S. Graphdf: A discrete flow model for molecular graph generation. In *International Conference on Machine Learning*, volume 139, pp. 7192–7203, 2021.

Ma, K., Yang, H., Yang, H., Jin, T., Chen, P., Chen, Y., Kamhoua, B. F., and Cheng, J. Improving graph representation learning by contrastive regularization. *arXiv preprint*, arXiv:2101.11525, 2021.

Mahajan, D., Tople, S., and Sharma, A. Domain generalization using causal matching. In *International Conference on Machine Learning*, volume 139, pp. 7313–7324, 2021.

McPherson, M., Smith-Lovin, L., and Cook, J. M. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

Mendez, D., Gaulton, A., Bento, A. P., Chambers, J., Veij, M. D., Felix, E., Magariños, M. P., Mosquera, J. F., Mutowo-Meullenet, P., Nowotka, M., Gordillo-Marañón, M., Hunter, F. M. I., Junco, L., Mugumbate, G., Rodríguez-López, M., Atkinson, F., Bosc, N., Radoux, C. J., Segura-Cabrera, A., Hersey, A., and Leach, A. R. Chembl: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(Database-Issue):D930–D940, 2019.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K.-R. Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pp. 41–48, 1999.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI Conference on Artificial Intelligence*, pp. 4602–4609, 2019.

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

Morris, C., Lipman, Y., Maron, H., Rieck, B., Kriege, N. M., Grohe, M., Fey, M., and Borgwardt, K. M. Weisfeiler and leman go machine learning: The story so far. *arXiv preprint*, arXiv:2112.09992, 2021.

Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, 12(2):181–201, 2001.

Murphy, R. L., Srinivasan, B., Rao, V. A., and Ribeiro, B. Relational pooling for graph representations. In *International Conference on Machine Learning*, volume 97, pp. 4663–4673, 2019.

Nagarajan, V., Andreassen, A., and Neyshabur, B. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations*, 2021.

Namkoong, H. and Duchi, J. C. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems*, pp. 2208–2216, 2016.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

Pearl, J. *Causality*. Cambridge University Press, 2 edition, 2009.

Pearl, J. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62 (3):54–60, feb 2019. ISSN 0001-0782.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Peters, J., Janzing, D., and Schlkopf, B. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, 2017. ISBN 0262037319.

Ranjan, E., Sanyal, S., and Talukdar, P. P. ASAP: adaptive structure aware pooling for learning hierarchical graph representations. In *AAAI Conference on Artificial Intelligence*, pp. 5470–5477, 2020.

Rojas-Carulla, M., Schölkopf, B., Turner, R., and Peters, J. Invariant models for causal transfer learning. *Journal of Machine Learning Research*, 19(36):1–34, 2018.

Rosenfeld, E., Ravikumar, P. K., and Risteski, A. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2021.

Sagawa*, S., Koh*, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020.

Salakhutdinov, R. and Hinton, G. E. Learning a nonlinear embedding by preserving class neighbourhood structure. In *International Conference on Artificial Intelligence and Statistics*, volume 2, pp. 412–419, 2007.

Sanchez-Gonzalez, A., Heess, N., Springenberg, J. T., Merel, J., Riedmiller, M. A., Hadsell, R., and Battaglia, P. W. Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning*, volume 80, pp. 4467–4476, 2018.

Santoro, A., Hill, F., Barrett, D. G. T., Morcos, A. S., and Lillicrap, T. P. Measuring abstract reasoning in neural networks. In *International Conference on Machine Learning*, volume 80, pp. 4477–4486, 2018.

Saxton, D., Grefenstette, E., Hill, F., and Kohli, P. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*, 2019.

Schölkopf, B. Causality for machine learning. *arXiv preprint*, arXiv:1911.10500, 2019.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, 2021.

Snijders, T. A. and Nowicki, K. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. In *Journal of Classification*, volume 14, pp. 75–100, 1997.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Sterling, T. and Irwin, J. J. Zinc 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 2015.

Sun, B. and Saenko, K. Deep CORAL: correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, volume 9915, pp. 443–450, 2016.

Tang, H., Huang, Z., Gu, J., Lu, B., and Su, H. Towards scale-invariant graph-related problem solving by iterative homogeneous gnns. In *Advances in Neural Information Processing Systems*, 2020.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. In *Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377, 1999.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint*, arXiv:1807.03748, 2018.

Vapnik, V. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems*, pp. 831–838, 1991.

Velickovic, P., Ying, R., Padovano, M., Hadsell, R., and Blundell, C. Neural execution of graph algorithms. In *International Conference on Learning Representations*, 2020.

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

International Conference on Learning Representations, 2018.

Wang, J., Lan, C., Liu, C., Ouyang, Y., and Qin, T. Generalizing to unseen domains: A survey on domain generalization. In *International Joint Conference on Artificial Intelligence*, pp. 4627–4635, 2021.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, volume 119, pp. 9929–9939, 2020.

Wu, B., Li, J., Hou, C., Fu, G., Bian, Y., Chen, L., and Huang, J. Recent advances in reliable deep graph learning: Adversarial attack, inherent noise, and distribution shift. *arXiv preprint arXiv:2202.07114*, 2022a.

Wu, Q., Zhang, H., Yan, J., and Wipf, D. Handling distribution shifts on graphs: An invariance perspective. In *International Conference on Learning Representations*, 2022b.

Wu, Y., Wang, X., Zhang, A., He, X., and Chua, T.-S. Discovering invariant rationales for graph neural networks. In *International Conference on Learning Representations*, 2022c.

Xhonneux, L.-P. A. C., Deac, A., Veličković, P., and Tang, J. How to transfer algorithmic reasoning knowledge to learn new algorithms? In *Advances in Neural Information Processing Systems*, 2021.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pp. 5449–5458, 2018.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Xu, K., Li, J., Zhang, M., Du, S. S., Kawarabayashi, K., and Jegelka, S. What can neural networks reason about? In *International Conference on Learning Representations*, 2020.

Xu, K., Zhang, M., Jegelka, S., and Kawaguchi, K. Optimization of graph neural networks: Implicit acceleration by skip connections and more depth. In *International Conference on Machine Learning*, volume 139, pp. 11592–11602, 2021a.

Xu, K., Zhang, M., Li, J., Du, S. S., Kawarabayashi, K., and Jegelka, S. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations*, 2021b.

Yang, H., Ma, K., and Cheng, J. Rethinking graph regularization for graph neural networks. In *AAAI Conference on Artificial Intelligence*, pp. 4573–4581, 2021.

Yehudai, G., Fetaya, E., Meirom, E., Chechik, G., and Maron, H. From local structures to size generalization in graph neural networks. In *International Conference on Machine Learning*, volume 139, pp. 11975–11986, 2021.

Yeung, R. *Information Theory and Network Coding*. 01 2008. ISBN 978-0-387-79233-0.

Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W. L., and Leskovec, J. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, pp. 4805–4815, 2018.

Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems*, pp. 9240–9251, 2019.

You, J., Ying, R., Ren, X., Hamilton, W. L., and Leskovec, J. Graphrnn: Generating realistic graphs with deep autoregressive models. In *International Conference on Machine Learning*, volume 80, pp. 5694–5703, 2018.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5812–5823, 2020.

You, Y., Chen, T., Shen, Y., and Wang, Z. Graph contrastive learning automated. In *International Conference on Machine Learning*, volume 139, pp. 12121–12132, 18–24 Jul 2021.

Yu, J., Xu, T., Rong, Y., Bian, Y., Huang, J., and He, R. Graph information bottleneck for subgraph recognition. In *International Conference on Learning Representations*, 2021.

Yuan, H., Yu, H., Gui, S., and Ji, S. Explainability in graph neural networks: A taxonomic survey. *arXiv preprint*, arXiv:2012.15445, 2020.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Zhang, M., Sohoni, N. S., Zhang, H. R., Finn, C., and Ré, C. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. *arXiv preprint*, arXiv:2203.01517, 2022.

Zhang, Y., Gong, M., Liu, T., Niu, G., Tian, X., Han, B., Schölkopf, B., and Zhang, K. Adversarial robustness through the lens of causality. *arXiv preprint*, arXiv:2106.06196, 2021.

Zimmermann, R. S., Sharma, Y., Schneider, S., Bethge, M., and Brendel, W. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, volume 139, pp. 12979–12990, 2021.

Appendix of GOOD

Contents

A Full Structural Causal Models on Graph Generation **10**

B Full GOOD Framework **12**

B.1 Invariant Graph Neural Networks 12

B.2 Invariance for OOD Generalization on Graphs 12

B.3 Invariant Graph Learning Framework 12

B.4 Theoretical Analysis and Practical Discussions 14

C More Discussions on Related Works and Future Directions **14**

C.1 More backgrounds 14

C.2 Related work 15

C.3 More discussions on connections of GOOD with existing work 17

D More Details about Failure Case Studies in Sec. 2.2 **17**

D.1 More empirical details about failure case study in Sec. 2.2 17

D.2 More discussions about failure case study in Sec. 2.2 18

D.3 More empirical results about failure case study in Sec. 2.2 19

D.4 Theoretical discussions for failure case study in Sec. 2.2 21

E Theory and Discussions **23**

E.1 Challenges of OOD generalization on graphs. 23

E.2 Proof for theorem B.2 (i) 23

E.3 Proof for theorem B.2 (ii) 25

F Details of Prototypical GOOD Implementation **28**

G Full Experimental Results and Settings **29**

G.1 Full experimental results and analysis 29

G.2 Details about the datasets 31

G.3 Training and Optimization in Experiments 32

G.4 Software and hardware 33

A. Full Structural Causal Models on Graph Generation

Supplementary to the graph generation process in Sec. 2.1, we provide full SCMs on the graph generation process in this section as shown in Fig. 4. Formal descriptions are given as Assumptions A.1, A.2, A.3, A.4.

To begin with, we take a latent-variable model perspective on the graph generation process and assume that the graph is generated through a mapping $f_{\text{gen}} : \mathcal{Z} \to \mathcal{G}$, where $\mathcal{Z} \subseteq \mathbb{R}^n$ is the latent space and $\mathcal{G} = \cup_{N=1}^{\infty}\{0,1\}^N \times \mathbb{R}^{N\times d}$ is the graph space. Let E denote environments. Following previous works (Kügelgen et al., 2021; Ahuja et al., 2021), we partition the latent variable from $\mathcal{Z}$ into an invariant part $C \in \mathcal{C} = \mathbb{R}^{n_c}$ and a varying part $S \in \mathcal{S} = \mathbb{R}^{n_s}$, s.t., $n = n_c + n_s$, according to whether they are affected by E. Similarly in images, C and S can represent content and style while E can refer to the locations where the images are taken (Beery et al., 2018; Zhang et al., 2021; Kügelgen et al., 2021). While in graphs, C can be the latent variable that controls the generation of functional groups in a molecule, which can not be affected by the changes of environments, such as species (Scaffold), experimental environment for examining the chemical property (Assay) (Ji et al., 2022). On the contrary, the other latent variable S inherits environment-specific information thus can further affect the finally generated graphs. Besides, C and S can have multiple types of interactions at the latent space with environments E and labels Y, which will generate different types of spurious correlations (Ahuja et al., 2021).

Assumption A.1 (Graph Generation SCM)**.**

$$\begin{aligned}
&(Z_A^c, Z_X^c) := f_{\text{gen}}^{(A,X)^c}(C),\ G_c := f_{\text{gen}}^{G_c}(Z_A^c, Z_X^c),\\
&(Z_A^s, Z_X^s) := f_{\text{gen}}^{(A,X)^s}(S),\ G_s := f_{\text{gen}}^{G_s}(Z_A^s, Z_X^s, E_G),\\
&G := f_{\text{gen}}^{G}(G_c, G_s, E_G).
\end{aligned}$$

Specifically, the graph generation process is shown as Fig. 4(a). The generative mapping f_{gen} is decomposed into $f_{\text{gen}}^{(A,X)^c}$,$f_{\text{gen}}^{G_c}$, $f_{\text{gen}}^{(A,X)^s}$,$f_{\text{gen}}^{G_s}$ and f_{gen}^{G} to control the generation of (Z_A^c, Z_X^c), G_c, (Z_A^s, Z_X^s), G_s, and G, respectively. Given the variable partitions C and S at the latent space $\mathcal{Z}$, they control the generation of the adjacency matrix and features for the invariant subgraph G_c and spurious subgraph G_s through two pairs of latent variables (Z_A^c, Z_X^c) and (Z_A^s, Z_X^s), respectively. Z_A^c and Z_A^s will control the structure-level properties in the generated graphs, such as degrees, sizes, and subgraph densities. While Z_X^c and Z_X^s mainly control the attribute-level properties in the generated graphs, such as homophily. Then, G_c and G_s are entangled into the observed graph G through f_{gen}^{G}. It can be a simply JOIN of a G_c with one or multiple G_s, or more complex generation processes controlled by the latent variables (Snijders & Nowicki, 1997; Lovász & Szegedy, 2006; You et al., 2018; Luo et al., 2021; Bevilacqua et al., 2021). Note that since our focus is to describe the potential distribution shifts with SCMs, in Assumption A.1, we aim to build a SCM that is compatible to many graph generation processes (Snijders & Nowicki, 1997; Lovász & Szegedy, 2006; You et al., 2018; Luo et al., 2021), and leave specific graph generation processes and their implications to OOD generalization to future work.

Moreover, a subset of environment latent variable $E_G \subseteq E$ will affect the generation of G and G_s. Thus, graphs collected from different environments can have different structure-level properties such as degrees, graph sizes, and subgraph densities, as well as feature-level properties such as homophily (Knyazev et al., 2019; Yehudai et al., 2021; Bevilacqua et al., 2021; Chen et al., 2022). Meanwhile, all of them can spuriously correlated with the labels depending on how the underlying latent variables are interacted with each others. The interaction types can be further divided into two axiom types FIIF and PIIF, as well as the mixed one MIIF. Previous OOD methods such as GIB (Yu et al., 2021) and DIR (Wu et al., 2022c) mainly focus on FIIF case, while others such as IRM (Arjovsky et al., 2019) mainly focuses on the PIIF case. Evidences show that failing to model either of them when developing the OOD objectives can have serious performance degenerations in practice (Aubin et al., 2021; Nagarajan et al., 2021). That is why we aim to model both of them in our solution.

Assumption A.2 (FIIF SCM)**.**

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

Assumption A.3 (PIIF SCM)**.**

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

Assumption A.4 (MIIF SCM)**.**

$$Y := f_{\text{inv}}(C),\ S_1 := f_{\text{spu}}(C, E),\ S_2 := f_{\text{spu}}(Y, E),\ G := f_{\text{gen}}(C, S_1, S_2).$$

As for the interactions between C and S at the latent space, we categorize the interaction modes into Fully Informative Invariant Features (FIIF, Fig. 4(b)), and Partially Informative Invariant Features (PIIF, Fig. 4(c)), depending on whether the latent invariant part C is fully informative about label Y, i.e., $(S, E) \perp\!\!\!\perp Y|C$. It is also possible that FIIF and PIIF are entangled into a Mixed Informative Invariant Features (MIIF,Fig. 4(d)). We follow Arjovsky et al. (2019); Ahuja et al. (2021) to formulate the SCMs for FIIF and PIIF, where we omit noises for simplicity (Pearl, 2009; Peters et al., 2017). Since MIIF is built upon FIIF and PIIF, we will focus on the axiom interaction modes (FIIF and PIIF) in this paper, while most of our discussions can be extended to MIIF or more complex interactions built upon FIIF and PIIF.

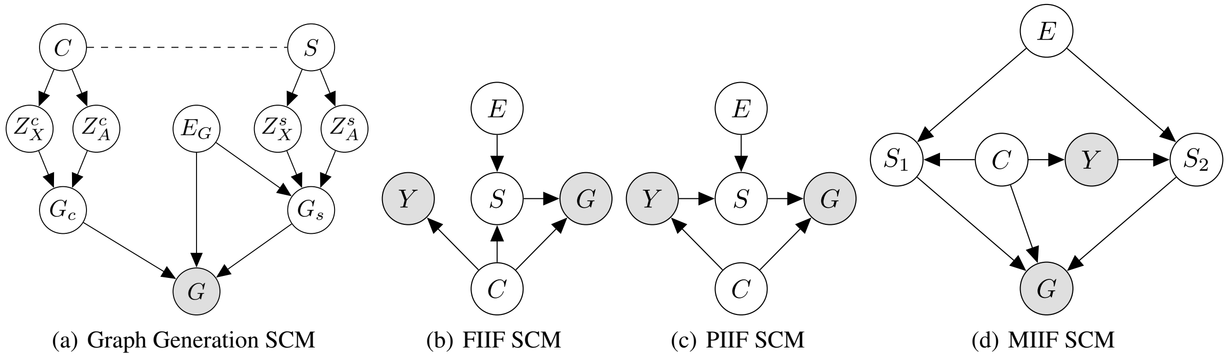


Figure 4. Full SCMs on Graph Distribution Shifts.

Among all of the interaction modes, f_{gen} corresponds to the graph generation process in Assumption A.1. f_{spu} is the mechanism describing how S is affected by C and E at the latent space. In FIIF, S is directly controlled by C while in PIIF, indirectly controlled by C through Y, which can exhibit different behaviors in practice (Ahuja et al., 2021; Nagarajan et al., 2021). Additionally, in MIIF, S is further partitioned into S_1 and S_2 depending on whether it is directly or indirectly controlled by C, respectively. Moreover, $f_{\text{inv}} : \mathcal{C} \to \mathcal{Y}$ indicates the labeling process, which assigns labels Y for the corresponding G merely based on C. Consequently, $\mathcal{C}$ is better clustered than $\mathcal{S}$ when given Y (Burshtein et al., 1992; Chapelle et al., 2006; Schölkopf, 2019; Schölkopf et al., 2021), which also serves as the necessary separation assumption for a classification task (Muller et al., 2001; Chen et al., 2005; Mika et al., 1999).

Assumption A.5 (Better Clustered Invariant Features)**.** $H(C|Y) \leq H(S|Y)$.

B. Full GOOD Framework

B.1. Invariant Graph Neural Networks

To start, we formulate the desired GNN that is able to generalize to OOD graphs under different distribution shifts as below.

Definition B.1 (Invariant GNN)**.** Given a set of graph datasets $\{\mathcal{D}^e\}_e$ and environments $\mathcal{E}_{\text{all}}$ that follow the same graph generation process in Sec. 2.1, considering a GNN $\rho \circ h$ that has a permutation invariant graph encoder $h : \mathcal{G} \to \mathbb{R}^h$ and a downstream classifier $\rho : \mathbb{R}^h \to \mathcal{Y}$, $\rho \circ h$ is an invariant GNN if it minimizes the worst case risk among all environments, i.e., $\min \max_{e \in \mathcal{E}_{\text{all}}} R^e$.

B.2. Invariance for OOD Generalization on Graphs

Towards extending the invariance principle to graphs under SCMs in Sec. 2.1, we need to identify a set of variables that have stable causal relationship with Y under both FIIF and PIIF (Assumption A.2, A.3). According to the ICM assumption (Peters et al., 2017), the labeling process $C \to Y$ is not informed nor influenced by other processes, implying that the conditional distribution $P(Y|C)$ remains invariant to the interventions on the environment latent variable E (Pearl, 2009). Consequently, for a GNN with a permutation invariant encoder $h : \mathcal{G} \to \mathbb{R}^h$ and a downstream classifier $\rho : \mathbb{R}^h \to \mathcal{Y}$, if h can recover the information of C from G in the learned graph representations, then the learning of ρ resembles traditional ERM (Vapnik, 1991) and can achieve the desired min-max optimality required by an invariant GNN (Def. B.1). However, recovering C from G is particularly difficult, since the generation of G from C involves two causal mechanisms $f_{\text{gen}}^{G_c}$ and f_{gen}^{G} in Assumption A.1. The unavailability of E further adds up the difficulty of enforcing the independence between the learned representations and E.

B.3. Invariant Graph Learning Framework

Causal algorithmic alignment. To enable a GNN to learn to extract the information about C from G, we propose the GOOD framework that *explicitly aligns with* the two causal mechanisms $f_{\text{gen}}^{G_c}$ and f_{gen}^{G} in Assumption A.1. The idea of alignment in GOOD is motivated by the algorithmic reasoning results that a neural network can learn a reasoning process better if its computation structure aligns with the process better (Xu et al., 2020; 2021b). Specifically, we realize the alignment by decomposing a GNN into two sub-components[1]: a) a featurizer GNN $g : \mathcal{G} \to \mathcal{G}_c$ aiming to identify the desired G_c; b) a classifier GNN $f_c : \mathcal{G}_c \to \mathcal{Y}$ that predicts the label Y based on the estimated G_c, where $\mathcal{G}_c$ refers to the

[1] The encoder of the GNN in GOOD can be regarded as the composition of g and the graph encoder in f_c.

space of subgraphs of G. Formally, the learning objectives of f_c and g can be formulated as:

$$\min_{f_c,\, g} \ R(f_c(\hat{G}_c)), \ \text{s.t.}\ \hat{G}_c \perp\!\!\!\perp E, \ \hat{G}_c = g(G), \tag{4}$$

where $R(f_c(\hat{G}_c))$ is the empirical risk of f_c that takes $\hat{G}_c$ as inputs to predict label Y for G, and $\hat{G}_c$ is the intermediate subgraph containing information about C and hence needs to be independent of E. Moreover, the extracted G_c can either share the same graph space with input G or has its own space with latent node and edge features, depending on the specific graph generation process. In practice, architectures from the literature of interpretable GNNs are compatible with GOOD (Yuan et al., 2020), hence can serve as practical choices for the implementation of GOOD. More details are given in Appendix F.

Although we can technically align with the two causal mechanisms with g and f_c, trivially optimizing this architecture can not satisfy the condition $\hat{G}_c \perp\!\!\!\perp E$. Formally, merely minimizing $R(f_c(\hat{G}_c))$ is equivalent to maximizing a variational lower bound of $I(\hat{G}_c; Y)$ (Alemi et al., 2017; Yu et al., 2021), which may include a subgraph from G_s in $\hat{G}_c$ since G_s also shares certain mutual information with Y. Moreover, the unavailability of E prevents the direct usage of E in enforcing the independence that is often adopted by previous objectives (Arjovsky et al., 2019; Krueger et al., 2021; Ahuja et al., 2021; Sagawa* et al., 2020; Ganin et al., 2016; Sun & Saenko, 2016; Dou et al., 2019; Mahajan et al., 2021), making the identification of G_c more challenging.

Optimization objective. To mitigate this issue, we need to translate other properties of G_c from SCMs in Sec. 2.1 into differentiable and equivalent conditions to the independence constraint $\hat{G}_c \perp\!\!\!\perp E$.

To start, consider a simplistic setting where all the invariant subgraphs G_c have the same size s_c, i.e., $|G_c| = s_c$[2]. When maximizing $I(\hat{G}_c; Y)$ by minimizing $R(f_c(\hat{G}_c))$ in Eq. 4, we are trying to extract all of the informative parts in G about Y into $\hat{G}_c$. For FIIF (Fig. 2(b)), as G_c already contains the maximal possible information in G about Y, G_c is a solution to $\max I(\hat{G}_c; Y)$. However, some subgraph of G_c can be replaced by some subgraph of G_s that is equally informative about Y. For PIIF (Fig. 2(c)), there also exists some subgraph of G_s that contains additional information about Y than G_c, hence $\hat{G}_c$ is more likely to involve some subgraph of G_s. Thus, the new condition needs to eliminate the auxiliary subgraphs of $\hat{G}_c$ from G_s such that the estimated $\hat{G}_c$ can only contain G_c.

Recall that for both FIIF and PIIF SCMs (Fig. 2), given two environments e_1 and e_2: if G_c appears in both e_1 and e_2, the correctly identified $\hat{G_c^{e_1}}, \hat{G_c^{e_2}}$ in e_1 and e_2 tend to have higher mutual information about the other, i.e., $(G_c, G_c) \in \arg\max I(\hat{G_c^{e_1}}; \hat{G_c^{e_2}})$. While for G_c and another $G_{c'}$ corresponding to a different $C' \neq C$, if they appear in the same environment, then including any subgraph from G_s in the estimated $\hat{G}_c, \hat{G}_{c'}$ for $G_c, G_{c'}$ will enlarge their mutual information, i.e., $(G_c, G_{c'}) \in \arg\min I(\hat{G}_c; \hat{G}_{c'})$. Thus, we can derive another important property about G_c:

$$G_c \in \arg\max_{\hat{G}_c} I(\hat{G}_c; \tilde{G}_c | C = c) - I(\hat{G}_c; \hat{G}_{c'} | C = c', c' \neq c), \tag{5}$$

where $\hat{G}_c$ and $\tilde{G}_c$ share the same C while $\hat{G}_{c'}$ corresponds to a different c'. In practice, we are also not given C. However, since C and Y shares a stable causal relationship in both FIIF and PIIF SCMs, Y can serve as an alternative of C in Eq. 5. Moreover, when both $I(\hat{G}_c; \tilde{G}_c | C = c)$ and $I(\hat{G}_c; Y)$ are maximized, $I(\hat{G}_c; \hat{G}_{c'} | C = c', c' \neq c)$ is automatically minimized, otherwise all classes will collapse to trivial solutions which is not possible given $I(\hat{G}_c; Y)$ being maximized. Thus, we can replace the independence condition in Eq. 4 and obtain the following objective (GOODv1):

$$\text{(GOODv1)} \qquad \max_{f_c, g} \ I(\hat{G}_c; Y), \ \text{s.t.}\ \hat{G}_c \in \arg\max_{\hat{G}_c = g(G), |\hat{G}_c| \leq s_c} I(\hat{G}_c; \tilde{G}_c | Y), \tag{6}$$

where $\hat{G}_c = g(G), \tilde{G}_c = g(\tilde{G})$ and $\tilde{G} \sim \mathbb{P}(G|Y)$, i.e., $\tilde{G}$ and G have the same label. In Theorem B.2, we discuss why Eq. 6 is equivalent to Eq. 4. Although being effective, Eq. 6 requires a strong assumption about the size of G_c. However, the size of G_c is usually unknown or changes for different Cs. In this circumstance, maximizing Eq. 6 without additional constraints would lead to the presence of subgraphs of G_s in $\hat{G}_c$. For instance, $\hat{G}_c = G$ is a trivial solution to Eq. 6 when $s_c = \infty$.

To circumvent this limitation, we further resort to the properties of G_s. In both FIIF and PIIF SCMs (Fig. 2), G_s and G_c share certain overlapped information about Y via some subgraphs. When maximizing $I(\hat{G}_c; \tilde{G}_c | Y)$ and $I(\hat{G}_c; Y)$, the appearances of these subgraphs in $\hat{G}_c$ will not affect the optimality. On the other hand, it can reduce the mutual information between the left part $\hat{G}_s = G - \hat{G}_c$ and Y, i.e., $I(\hat{G}_s; Y)$. In other words, by maximizing $I(\hat{G}_s; Y)$, we can avoid involving

[2] Throughout the paper, we use generalized set operators for the ease of understanding. They can have multiple implementations in terms of nodes, edges or attributes.

additional subgraphs from G_s into $\hat{G}_c$. Meanwhile, to avoid trivial solution that $G_c \subseteq \hat{G}_s$ during maximizing $I(\hat{G}_s;Y)$, we can leverage the better clustering property of G_c implied by Assumption A.5 to derive the constraint $I(\hat{G}_s;Y) \leq I(\hat{G}_c;Y)$. Thus, we can obtain a new objective GOODv2 as follows:

$$\begin{aligned}
&\max_{f_c,g} \ I(\hat{G}_c;Y)+I(\hat{G}_s;Y), \ \text{s.t.}\ \hat{G}_c \in \operatorname*{arg\,max}_{\hat{G}_c=g(G),\tilde{G}_c=g(\tilde{G})} \ I(\hat{G}_c;\tilde{G}_c|Y),\\
&\text{(GOODv2)} \qquad\qquad\qquad\qquad I(\hat{G}_s;Y) \leq I(\hat{G}_c;Y),\ \hat{G}_s = G - g(G),
\end{aligned} \tag{7}$$

where $\hat{G}_c = g(G), \tilde{G}_c = g(\tilde{G})$ and $\tilde{G} \sim \mathbb{P}(G|Y)$, i.e., $\tilde{G}$ and G have the same label. We also prove the equivalence between Eq. 7 and Eq. 4 in Theorem B.2.

B.4. Theoretical Analysis and Practical Discussions

Theorem B.2 (GOOD Induces Invariant GNNs). *Given a set of graph datasets $\{\mathcal{D}^e\}_e$ and environments $\mathcal{E}_{all}$ that follow the same graph generation process in Sec. 2.1, assuming that* (a) *f_{gen}^{G} and $f_{gen}^{G_c}$ in Assumption A.1 are invertible,* (b) *samples from each training environment are equally distributed, i.e.,$|\mathcal{D}_{\hat{e}}| = |\mathcal{D}_{\tilde{e}}|$, $\forall \hat{e}, \tilde{e} \in \mathcal{E}_{tr}$, then:*

(i). If $\forall G_c, |G_c| = s_c$, then each solution to Eq. 6, elicits an invariant GNN (Def. B.1).

(ii). Each solution to Eq. 7, elicits an invariant GNN (Def. B.1).

We prove Theorem B.2 (i) and (ii) in Appendix E.2, E.3, respectively.

Practical implementations of GOOD objectives. After showing the power of GOOD, we introduce the practical implementations of GOODv1 and GOODv2 objectives. Specifically, an exact estimate of the second term $I(\hat{G}_c;\tilde{G}_c|Y)$ could be highly expensive (van den Oord et al., 2018; Belghazi et al., 2018). However, contrastive learning with supervised sampling provides a practical solution for the approximation (Khosla et al., 2020; Chopra et al., 2005; Salakhutdinov & Hinton, 2007; van den Oord et al., 2018; Belghazi et al., 2018):

$$I(\hat{G}_c;\tilde{G}_c|Y) \approx \mathbb{E}_{\substack{\{\hat{G}_c,\tilde{G}_c\}\sim\mathbb{P}_g(G|\mathcal{Y}=Y)\\ \{G_c^i\}_{i=1}^M\sim\mathbb{P}_g(G|\mathcal{Y}\neq Y)}} \log \frac{e^{\phi(h_{\hat{G}_c},h_{\tilde{G}_c})}}{e^{\phi(h_{\hat{G}_c},h_{\tilde{G}_c})} + \sum_i^M e^{\phi(h_{\hat{G}_c},h_{G_c^i})}}, \tag{8}$$

where positive samples $(\hat{G}_c, \tilde{G}_c)$ are the extracted subgraphs of graphs that have the same label of G, negative samples are those with different labels, $\mathbb{P}_g(G|\mathcal{Y}=Y)$ is the push-forward distribution of $\mathbb{P}(G|\mathcal{Y}=Y)$ by featurizer g, $\mathbb{P}(G|\mathcal{Y}=Y)$ refers to the distribution of G given the label Y, $\mathbb{P}(G|\mathcal{Y}\neq Y)$ refers to the distribution of G given the label that is different from Y, $h_{\hat{G}_c}, h_{\tilde{G}_c}, h_{G_c^i}$ are the graph presentations of the estimated subgraphs, and ϕ is the similarity metric for graph presentations. As $M \to \infty$, Eq. 8 approximates $I(\hat{G}_c;\tilde{G}_c|Y)$, which can be regarded as a non-parameteric resubstitution entropy estimator via the von Mises-Fisher kernel density (Ahmad & Lin, 1976; Kandasamy et al., 2015; Wang & Isola, 2020). Thus, plugging it into Eq. 6 and Eq. 7 can relieve the issue of approximating $I(\hat{G}_c;\tilde{G}_c|Y)$ in practice.

For the implementation of $I(\hat{G}_s;Y)$ and the constraint $I(\hat{G}_s;Y) \leq I(\hat{G}_c;Y)$ in GOODv2, a practical choice is to follow the idea of hinge loss, $I(\hat{G}_s;Y) \approx \frac{1}{N} R_{\hat{G}_s} \cdot \mathbb{I}(R_{\hat{G}_s} \leq R_{\hat{G}_c})$, where N is the number of samples, $\mathbb{I}$ is an indicator function that outputs 1 when the inner condition is satisfied otherwise 0, and $R_{\hat{G}_s}$ and $R_{\hat{G}_c}$ are the empirical risk vector of the predictions for each sample based on the corresponding $\hat{G}_s$ and $\hat{G}_c$. More implementation details can be found in Appendix F.

Discussions and implications of GOOD. Although using contrastive learning to improve OOD generalization is not new in the literature (Dou et al., 2019; Mahajan et al., 2021; Zhang et al., 2022), previous methods cannot yield OOD guarantees in graph circumstances due to the highly non-linearity and the unavailability of domain labels E. In particular, GOOD can *be reduced to directly applying contrastive learning* when without the decomposition for causal algorithmic alignment. However, in the experiments we found that merely using the contrastive objective, i.e., CNC (Zhang et al., 2022), yields unsatisfactory OOD generalization performance, which further implies the necessity of the decomposition in GOOD.

Moreover, the architecture of GOOD can have multiple other implementations for both the featurizer and classifier, such as identifying G_c at the latent space (Schölkopf, 2019; Schölkopf et al., 2021). Since we cannot enumerate every possible implementation, in this work we choose interpretable GNN architectures as a prototype validation for GOOD and leave more sophisticated architectures as future works. In particular, when optimized with ERM objective, GOOD can *be reduced to interpretable GNNs*. However, merely using interpretable GNNs such as ASAP (Ranjan et al., 2020), GIB (Yu et al., 2021) or DIR (Wu et al., 2022c) cannot yield satisfactory OOD performance. As shown in Table 2 and discussed in Appendix. D.4, GIB can only work for FIIF, while DIR *cannot* yield OOD guarantees for neither FIIF and PIIF SCMs. These results are also empirically validated in the experiments. We provide more detailed discussions in Appendix C.

C. More Discussions on Related Works and Future Directions

C.1. More backgrounds

We give more background introduction about GNNs and Invariant Learning in this section.

Graph Neural Networks. Let $G = (A, X)$ denote a graph with n nodes and m edges, where $A \in \{0,1\}^{n\times n}$ is the adjacency matrix, and $X \in \mathbb{R}^{n\times d}$ is the node feature matrix with a node feature dimension of d. In graph classification, we are given a set of N graphs $\{G_i\}_{i=1}^N \subseteq \mathcal{G}$ and their labels $\{Y_i\}_{i=1}^N \subseteq \mathcal{Y} = \mathbb{R}^c$ from c classes. Then, we train a GNN $\rho \circ h$ with an encoder $h : \mathcal{G} \to \mathbb{R}^h$ that learns a meaningful representation h_G for each graph G to help predict their labels $y_G = \rho(r_G)$ with a downstream classifier $\rho : \mathbb{R}^h \to \mathcal{Y}$. The representation h_G is typically obtained by performing pooling with a READOUT function on the learned node representations:

$$h_G = \text{READOUT}(\{h_u^{(K)} | u \in V\}), \tag{9}$$

where the READOUT is a permutation invariant function (e.g., SUM, MEAN) (Xu et al., 2019; Ying et al., 2018; Murphy et al., 2019; Xu et al., 2019; Chen et al., 2020; Morris et al., 2021), and $h_u^{(K)}$ stands for the node representation of $u \in V$ at K-th layer that is obtained by neighbor aggregation:

$$h_u^{(K)} = \sigma(W_K \cdot a(\{r_v^{(K-1)}\} | v \in \mathcal{N}(u) \cup \{u\})), \tag{10}$$

where $\mathcal{N}(u)$ is the set of neighbors of node u, $\sigma(\cdot)$ is an activation function, e.g., ReLU, and $a(\cdot)$ is an aggregation function over neighbors, e.g., MEAN.

Invariant Learning. Invariant learning typically considers a supervised learning setting based on the data $\mathcal{D} = \{\mathcal{D}^e\}_e$ collected from multiple environments $\mathcal{E}_{\text{all}}$, where $\mathcal{D}^e = \{G_i^e, y_i^e\}$ is the dataset from environment $e \in \mathcal{E}_{\text{all}}$. (G_i^e, y_i^e) from a single environment e are considered as drawn independently from an identical distribution $\mathbb{P}^e$. The goal of OOD generalization is to train a GNN $\rho \circ h : \mathcal{G} \to \mathcal{Y}$ with data from training environments $\mathcal{D}_{\text{tr}} = \{\mathcal{D}^e\}_{e \in \mathcal{E}_{\text{tr}} \subseteq \mathcal{E}_{\text{all}}}$, and generalize well to all (unseen) environments, i.e., to minimize:

$$\min_{\rho, h} \max_{e \in \mathcal{E}_{\text{all}}} R^e(\rho \circ h), \tag{11}$$

where R^e is the empirical risk under environment e (Vapnik, 1991; Peters et al., 2016; Arjovsky et al., 2019). More details can be referred in (Ahuja et al., 2021).

C.2. Related work

We provide detailed related work discussion in this section in complementary to that in Introduction (Sec. 1). To begin with, we summarize the main differences between our solution and them in Table 2.

An overview of related works. On Euclidean data, Invariant Learning (Arjovsky et al., 2019; Creager et al., 2021; Ahuja et al., 2021), Group Distributionally Robust Optimization (GroupDro) (Krueger et al., 2021; Sagawa* et al., 2020; Zhang et al., 2022), Domain Adaption (DA) and Domain Generalization (DG) (Ganin et al., 2016; Sun & Saenko, 2016; Li et al., 2018b; Dou et al., 2019; Mahajan et al., 2021; Wang et al., 2021) are three widely adopted approaches to enable OOD generalization. However, they all have limitations when being applied to graphs. First, previous invariant learning methods are mostly developed and analyzed for Euclidean data (Arjovsky et al., 2019; Ahuja et al., 2021; Creager et al., 2021), or under specific SCM assumptions (Arjovsky et al., 2019), making the theoretical results hardly applicable to the complicated graph data (Rosenfeld et al., 2021) that can have multiple types of distribution shifts (Nagarajan et al., 2021). GroupDro that minimizes the gap between worst group risk and average risk (Krueger et al., 2021; Sagawa* et al., 2020; Zhang et al., 2022), and DA/DG methods that aim to learn class-conditional domain invariant representations (Ganin et al., 2016; Sun & Saenko, 2016; Li et al., 2018b; Dou et al., 2019; Wang et al., 2021), cannot guarantee a min-max optimal predictor without additional assumptions (Arjovsky et al., 2019; Gulrajani & Lopez-Paz, 2021; Ahuja et al., 2021). Moreover, most existing methods require environment labels that are however expensive to obtain in graphs, which limits their applications to graphs (Arjovsky et al., 2019; Krueger et al., 2021; Ahuja et al., 2021; Sagawa* et al., 2020; Ganin et al., 2016; Sun & Saenko, 2016; Dou et al., 2019; Mahajan et al., 2021). In contrast, we aim at a unified framework that are provably generalizable under different types of distribution shifts on graphs.

Another line of relevant works is about GNN explainability that aims to find a subgraph of the input as the explanation for a GNN prediction (Ying et al., 2019; Yuan et al., 2020). Although some may leverage causality to justify the generated explanation (Lin et al., 2021), they mostly focus on understanding the predictions of GNNs instead of for OOD generalization. The most close works to ours are two interpretable GNNs that aim to explicitly extract a subgraph for both predictions and explanations. However, they focus on graphs and shifts generated under a specific SCM. Although one of them can provide theoretical guarantee for OOD generalization (Yu et al., 2021) by using the information bottleneck criteria (Ahuja et al., 2021), they would inevitably fail to generalize to graphs generated under different SCMs. More discussions about the failure are given in Appendix D.4. Besides, Bevilacqua et al. (2021) also discuss OOD generalization on graphs but limited a specific graph family and graph size shifts. Wu et al. (2022b) propose OOD methods on graphs for the task of node classification but limited to shifts under a specific SCM.

Causality and OOD Generalization. Causality comes to the stage for demystifying and improving the huge success of machine learning algorithms to further advances (Pearl, 2019; Schölkopf, 2019; Schölkopf et al., 2021). One of the most widely applied concept from causality is the Independent Causal Mechanism (ICM) that assumes conditional distribution of each variable given its causes (i.e., its mechanism) does not inform or influence the other conditional distributions (Pearl, 2009; Peters et al., 2017). The invariance principle is also induced from the ICM assumption. Once proper assumptions about the underlying data generation process via Structural Causal Models (SCM) are established, it is promising to apply the invariance principle to machine learning models for finding an invariant representation about the causal relationship between the underlying causes and the label (Peters et al., 2016; Arjovsky et al., 2019). Consequently, models built upon the invariant representation can generalize to unseen environments or domains with guaranteed performance (Peters et al., 2016; Rojas-Carulla et al., 2018; Arjovsky et al., 2019; Sagawa* et al., 2020; Bengio et al., 2020; Koyama & Yamaguchi, 2020; Gulrajani & Lopez-Paz, 2021; Krueger et al., 2021; Creager et al., 2021; Ahuja et al., 2021). The arguably first formulation of invariance principle was introduced by Peters et al. (2016). Arjovsky et al. (2019) propose a novel formulation of learning causal invariance in representation learning, i.e., IRM, show how it connects with existing areas such as distributional robust optimization (Namkoong & Duchi, 2016) and generalization (Zhang et al., 2017), and prove its effectiveness in addressing PIIF spurious correlations (Assumption A.3). However, in practice, both PIIF and FIIF (Assumption A.2) can appear in data, while IRM can fail in these cases (Aubin et al., 2021; Nagarajan et al., 2021). Ahuja et al. (2021) then propose to add information bottleneck criteria into the IRM formulation to address the issue. However, their results are restricted to linear regime and also require environment partitions to distinguish the sources of distribution shifts. Recently, Creager et al. (2021) and Lin et al. (2022) propose new OOD objectives to relieve the needs for environment partitions, but limited to PIIF spurious types and linear regime.

In parallel invariant learning approaches, Sagawa* et al. (2020) propose to regularize the worst group in group distributionally robust optimization (GroupDro). Zhang et al. (2022) propose a contrastive approach to tackle GroupDro when the group partitions are not available. However, minimizing the gap between worst group risk and averaged risk can not yield a OOD generalizable predictors in our circumstances. Besides, traditional approaches to tackle OOD generalization also include Domain Adaption, Transfer Learning and Domain Generalization(Rojas-Carulla et al., 2018; Chuang et al., 2020; Ganin et al., 2016; Sun & Saenko, 2016; Li et al., 2018b; Dou et al., 2019; Mahajan et al., 2021; Wang et al., 2021), which aim to learn the class conditional invariant representation shared across source domain and target domain. However, they all require a stronger assumption on the availability of target domain data or the ground truth predictors (Gulrajani & Lopez-Paz, 2021; Ahuja et al., 2021), hence are not able to yield predictors with OOD generalization guarantees. We refer interested readers to Pearl (2019); Schölkopf (2019); Schölkopf et al. (2021) for an in-depth understanding, and Gulrajani & Lopez-Paz (2021); Ahuja et al. (2021) for a thorough overview.

GNN Explainability. Works in GNN explainability aim to find a subgraph of the input graph as the explanation for the prediction of a GNN model (Ying et al., 2019; Yuan et al., 2020). Although some may leverage causality in explanation generation (Lin et al., 2021), they mostly focus on understanding the predictions of GNNs in a post-hoc manner instead of OOD generalization. Recently there are two works aiming to provide robust explanations under distribution shifts, i.e., GIB (Yu et al., 2021) and DIR (Wu et al., 2022c), and both of them focus on tackling FIIF spurious correlations (Assumption A.2). The theoretical guarantees of GIB follows the theory of information bottleneck (Tishby et al., 1999), while GIB can not solve PIIF spurious correlations (Assumption A.3). As both FIIF and PIIF widely exist in realistic scenarios, failing to solve either of them could result in severe performance degradation in practice (Arjovsky et al., 2019;

Table 2. An overview of potential algorithms for OOD generalization on graphs.

Algorithm	OOD Guarantee	Regime	E Known	SCM Support
IRM (Arjovsky et al., 2019)	Yes	$\mathbb{R}$	Yes	PIIF
IB-IRM (Ahuja et al., 2021)	Yes	$\mathbb{R}$	Yes	PIIF&FIIF
EIIL (Creager et al., 2021)	Yes	$\mathbb{R}$	No	PIIF
DANN (Ganin et al., 2016)	N/A	$\mathbb{R}$	Yes	N/A
MatchDG (Mahajan et al., 2021)	N/A	$\mathbb{R}$	Yes	FIIF
GroupDro (Sagawa* et al., 2020)	N/A	$\mathbb{R}$	Yes	N/A
CNC (Zhang et al., 2022)	N/A	$\mathbb{R}$	No	N/A
GIB (Yu et al., 2021)	Yes	$\mathcal{G}$	No	FIIF
DIR (Wu et al., 2022c)	No	$\mathcal{G}$	No	FIIF
GOOD (Ours)	**Yes**	$\mathcal{G}$	**No**	**PIIF&FIIF**

Ahuja et al., 2021; Aubin et al., 2021; Nagarajan et al., 2021). While for DIR, though as a generalization of Chang et al. (2020) to graphs, can not provide any theoretical guarantees under FIIF spurious correlations as shown in Appendix D.4, nor under PIIF spurious correlations.

GNN Extrapolation. Recently there is a surge of attention in improving the extrapolation ability of GNNs and apply them to various applications, such as mathematical reasoning (Santoro et al., 2018; Saxton et al., 2019), physics (Battaglia et al., 2016; Sanchez-Gonzalez et al., 2018), and graph algorithms (Tang et al., 2020; Velickovic et al., 2020; Xu et al., 2020; Xhonneux et al., 2021). Xu et al. (2021b) study the neural network extrapolation ability from a geometrical perspective. Han et al. (2021) improve OOD drug discovery by mitigating the overconfident misprediction issue. Knyazev et al. (2019); Yehudai et al. (2021) focus on the extrapolation of GNNs in terms of graph sizes, while making additional assumptions on the knowledge about ground truth attentions and access to test inputs. Bevilacqua et al. (2021) study the graph size extrapolation problem of GNNs through a causal lens, while the induced invariance principle is built upon assumptions on the specific family of graphs. Different from these works, we consider the GNN extrapolation as a causal problem, establish generic SCMs that are compatible with several graph generation models, as well as, more importantly, different types of distribution shifts. Hence, the induced the invariance principle and provable algorithms built upon the SCMs in our work can generalize to multiple graph families and distribution shifts.

Additionally, Wu et al. (2022b) propose causal models as well as specialized objectives to extrapolate nodes with different neighbors. However, their formulation is limited to node classification task and specific spurious correlation type. In contrast, the induced invariance principle in Wu et al. (2022b), can be seen as a extension of GOOD for node classification, where we cab identify an invariant subgraph from the K-hop neighbor graph of each node, and making predictions based on it, i.e., $Y \perp\!\!\!\perp E | G_c^{\text{ego}} \subseteq G_u^{\text{ego}}$ for node u. We leave specific formulation and implementation to future works.

C.3. More discussions on connections of GOOD with existing work

Although primarily serving for graph OOD generalization problem, our theory complements the identifiability study on graphs through contrastive learning, and aligns with the discoveries in the image domain that contrastive learning learns to isolate the content (C) and style (S) (Zimmermann et al., 2021; Kügelgen et al., 2021). Moreover, our results also partially explain the success of graph contrastive learning (You et al., 2020; Ma et al., 2021; You et al., 2021), where GNNs may implicitly learn to identify the underlying invariant subgraphs for prediction.

On expressivity of graph encoder in GOOD. The expressivity of GOOD is essentially constrained by the encoders embedded for learning graph representations. During isolating G_c from G, if the encoder can not differentiate two isomorphic graphs G_c and $G_c \cup G_s^p$ where $G_s^p \subseteq G_s$, then the featurizer will fail to identify the underlying invariant subgraph. Moreover, the classifier will also fail if the encoder can not differentiate two non-isomorphic G_cs from different classes. Thus, adopting more powerful graph representation encoders into GOOD can improve the OOD generalization.

On GOOD and graph information bottleneck. Under the FIIF assumption on latent interaction, the independence condition derived from causal model can also be rewritten as $Y \perp\!\!\!\perp S|C$ (similar to that in DIR (Wu et al., 2022c) as they also focus on FIIF), which further implies $Y \perp\!\!\!\perp S|\hat{G}_c$. Hence it is natural to use Information Bottleneck (IB) objective (Tishby et al., 1999) to solve for G_c:

$$\min_{f_c,g} R_{G_c}(f_c(\hat{G}_c)),\\ \text{s.t. } G_c = \operatorname*{arg\,max}_{\hat{G}_c=g(G)\subseteq G} I(\hat{G}_c,Y) - I(\hat{G}_c,\mathcal{G}), \tag{12}$$

which explains the success of many existing works in finding predictive subgraph through IB (Yu et al., 2021). However, the estimation of $I(\hat{G}_c, G)$ is notoriously difficult due to the complexity of graph, which can lead to unstable convergence as observed in our experiments. In contrast, optimization with contrastive objective in GOOD as Eq. 8 induces more stable convergence.

On GOOD for node classifications. As the task of node classification can be viewed as graph classification based on the ego-graphs of a node, our analysis and discoveries can generalize to node classification. More specifically, the invariance principle for node classification can be implemented by identifying an invariant subgraph from the K-hop neighbor graph of each node, and making predictions based on it, i.e., $Y \perp\!\!\!\perp E|G_c^{\text{ego}} \subseteq G_u^{\text{ego}}$ for node u (Wu et al., 2022b).

D. More Details about Failure Case Studies in Sec. 2.2

In this section, we provide details on failure case studies in Sec. 2.2. We first elaborate the empirical evaluation setting where we construct a synthetic graph datasets to probe the behaviors of existing methods in OOD generalization on graphs.

D.1. More empirical details about failure case study in Sec. 2.2

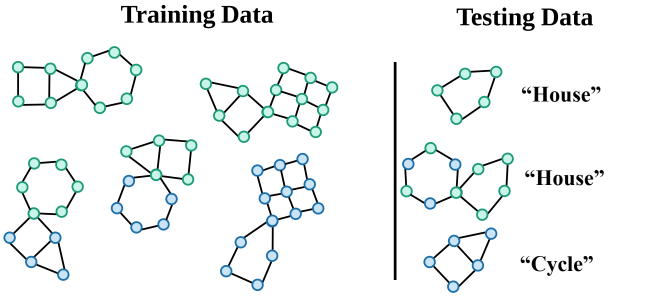


Figure 5. Failure cases for existing methods. GNNs are required to classify whether the graph contains a “house” or “cycle” motif, where the colors represent node features. However, distribution shifts in the training exists at both structure level (From left to right: “house” mostly co-occur with a hexagon), attribute level (From upper to lower: graphs nodes are mostly green colored if they contain “house”, or blued colored if they contain “cycle”), and graph sizes, making GNNs hard to capture the invariance. *ERM can fail* for leveraging the shortcuts and predict graphs that have a hexagon or have mostly green nodes as “house”. *IRM can fail* when testing data are not sufficiently supported by the training data.

To begin with, we construct 3-class synthetic datasets based on BAMotif (Luo et al., 2020) and follow Wu et al. (2022c) to inject spurious correlations between motif graph and base graph during the generation. In this graph classification task, the model needs to tell which motif the graph contains, e.g., “House” or “Cycle” motif, as shown in Fig. 5. We inject the distribution shifts in the training data while keeping the test data and validation data without the biases. For structure-level shifts, we introduce the artificial bias based on FIIF, where the motif and the base graph are spuriously correlated with a probability of various bias. For mixed shifts, we additionally introduced attribute-level shifts based on FIIF, where all of the node features are spuriously correlated with a probability of various bias. The number of training graphs is 600 for each class and the number of graphs in validation and test set is 200 for each class. More construction details are given in Appendix G.

For the GNN encoders, by default, we use 3-layer GCN (Kipf & Welling, 2017) with mean readout, a hidden dimension of 64, and JK jump connections (Xu et al., 2018) at the last layer. During training, we use a batch size of 32, learning rate of $1e-3$ with Adam optimizer (Kingma & Ba, 2015), and batch normalization between hidden layers (Ioffe & Szegedy, 2015). Meanwhile, to stabilize the training, we also use dropout (Srivastava et al., 2014) of 0.1 and early stop the training when the validation accuracy does not increase till 5 epoch after first 20 epochs. All of the experiments are repeated 5 times, and the mean accuracy as well as variance are reported and plotted. When using IRM objective (Arjovsky et al., 2019), as the environment partitions are not available, we generate 2 environments with random partitions.

D.2. More discussions about failure case study in Sec. 2.2

In Fig. 6, 7, 8, 9, we investigate whether existing training objectives (ERM and IRM), adding more message passing, as well as using expressive GNNs, can improve the OOD generalization ability on graphs. Here we also provide a additional discussion in complementary to the discussions on OOD generalization performance of ERM and IRM objectives in Sec. 2.2.

More concretely, we focus on answering the following question from multiple aspects.

Can better architectures improve OOD generalization of GNNs?

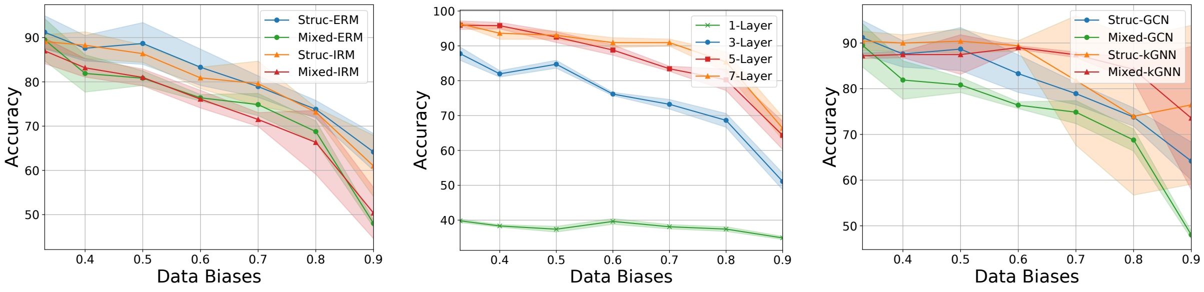
Training with ERM objective. As shown in Fig. 6, 7, 8, 9, we find that GNNs trained with the standard empirical risk minimization (ERM) algorithm (Vapnik, 1991) are not able to generalize to OOD graphs. As the data biases grows stronger, the performances of GNNs drop dramatically. Furthermore, when graph size shifts are mixed in the data, GNNs can have larger variance at low data biases, indicating the instability of learning the desired relationships for the task. The reason is that ERM tends to overfit to the shortcuts or spurious correlations presented in specific substructures or attributes in the graphs (Geirhos et al., 2020). This phenomenon has also been shown to exist in GNNs equipped with more sophisticated architectures such as attention mechanisms (Veličković et al., 2018), under graph size shifts (Knyazev et al., 2019).

Training with OOD objective. Meanwhile, as shown in Fig. 6, 7, 8, 9, OOD objectives primarily developed on Euclidean data such as invariant risk minimization (IRM) (Arjovsky et al., 2019) also cannot alleviate the problem. On the contrary, IRM can fail catastrophically at non-linear regime if without sufficient support overlap for the test environments, i.e., $\cup_{e\in\mathcal{E}_{\text{te}}}\text{supp}(\mathbb{P}^e) \not\subseteq \cup_{e\in\mathcal{E}_{\text{tr}}}\text{supp}(\mathbb{P}^e)$ (Rosenfeld et al., 2021). In addition to IRM, the failure would also happen for alternative objectives (Krueger et al., 2021; Bellot & van der Schaar, 2020; Ahuja et al., 2021) as proved by Rosenfeld et al. (2021). Besides, different distribution shifts on graphs can be nested with each other where each one can have distinct spurious correlation type, e.g., FIIF or PIIF. OOD objectives will also fail seriously if either of the correlation types is not supported (Aubin et al., 2021; Nagarajan et al., 2021). Moreover, non-trivial environment partitions or labels are required for performance guarantee of these OOD objectives (Arjovsky et al., 2019; Krueger et al., 2021; Sagawa* et al., 2020; Ahuja et al., 2021). However, collecting meaningful environment partitions of graphs requires expert knowledge about graph data. Thus, the environment labels can be expensive to obtain and are usually not available (Morris et al., 2020; Dwivedi et al., 2020; Hu et al., 2020). Alternative options such as random partitions tend not to alleviate the issue (Creager et al., 2021; Lin et al., 2022), as it can be trivially deemed as mini-batching.

Adding more message passing turns. It is a common practice in GNNs to denoise the signals by aggregating more neighbors with higher layers, or enhance the expressive power with more powerful readout functions (Xu et al., 2018; 2019; Yang et al., 2021). Aggregating neighbor information with more layers to denoise the input signal, or enhancing the expressivity with more powerful readout functions, are two common choices in GNNs to improve the generalization ability (Xu et al., 2018; Li et al., 2018a; Xu et al., 2019; Yang et al., 2021). However, in the experiments next, we empirically found that GCNs with more layers and more powerful readout operations are still sensitive to distribution shifts. In particular, stacking more layers helps denoising certain shifts, while the OOD performance would drop more sharply when the bias increases. Intuitively, if the spurious features from nodes cannot be eliminated by the denoising property of a deeper GNN, they would spread among the whole graph more widely, which in turn leads to stronger spurious correlations. Besides, the spurious correlations would be more difficult to be disentangled if there are distribution shifts at both structure-level and attribute-level. Since the node representations from hidden layers can also encode graph topology features (Xu et al., 2019), distribution shifts introduced through Z_A^s and Z_X^s will doubly mix at the learned features. In the worst case, the information about Z_A^c and Z_X^c could be partially covered by or even replaced by Z_A^s and Z_X^s. This will make OOD generalization of message passing GNNs trained through ERM much more difficult or even impossible. Besides, as the node representations of $1 \leq i \leq k$-th layer can also encode graph topology features (Xu et al., 2019), which, if spuriously correlated with labels through Z_A^s and entangled with part of invariant node features, i.e., Z_X^c, in the worst case, can greatly improve the difficulty or even make the OOD generalization impossible for neighbor aggregation GNNs trained with ERM.

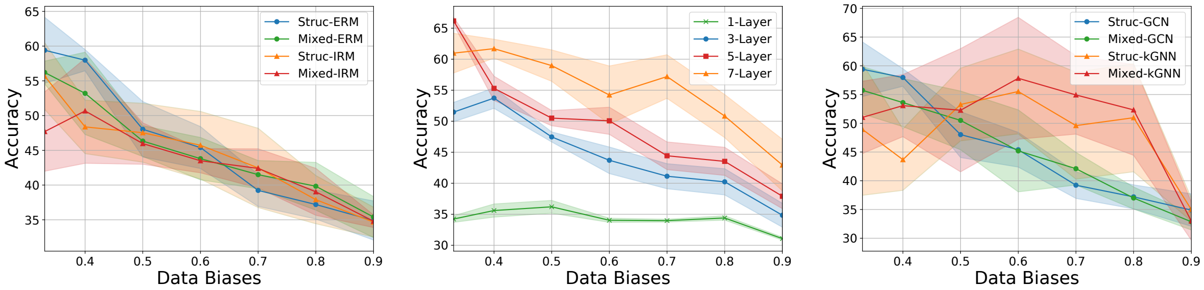
Using more expressive GNNs. Previous results on the expressivity of GNNs show that GNNs are limited to distinguish isomorphic graphs at most as 1-WL/2-WL test can distinguish (Xu et al., 2019). After that, many follow-up variants are proposed to improve the expressivity of GNNs (Morris et al., 2021). However, if the labels are spuriously correlated with certain subgraphs, even the GNN has high expressivity can still be prone to distribution shifts. In a idealistic case, when classifying a graph with a highly expressive GNN, it reduces to the linear or discrete feature case on the Euclidean regime. In this case, there exists many evidences showing that neural networks can fail to generalize to OOD data without a proper objective (Beery et al., 2018; DeGrave et al., 2021; Arjovsky et al., 2019; Sagawa* et al., 2020; Bengio et al., 2020; Krueger et al., 2021; Creager et al., 2021; Koyama & Yamaguchi, 2020; Ahuja et al., 2021). Empirically, we use k-GNNs (Morris et al., 2019) to verify the intuition and observe similar failures for this provably more expressive GNN as basic GNN variants.

D.3. More empirical results about failure case study in Sec. 2.2



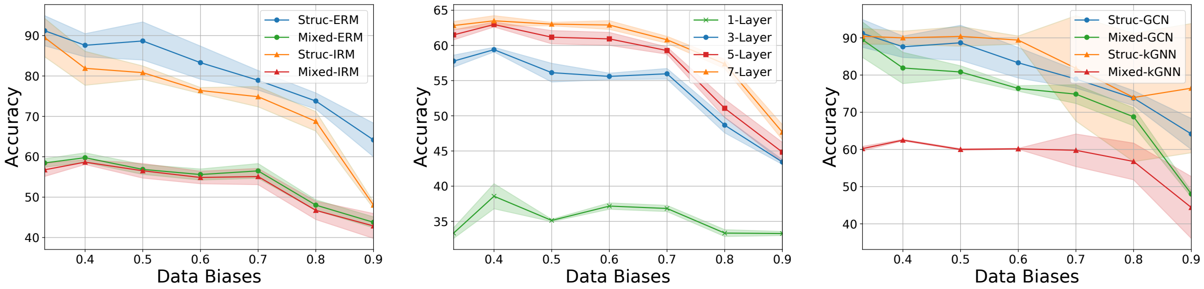
(a) Failures of training objectives. (b) Failures of deeper GNNs. (c) Failures of expressive GNNs.

Figure 6. Failure of existing methods on SPMotif with FIIF attribute shifts.



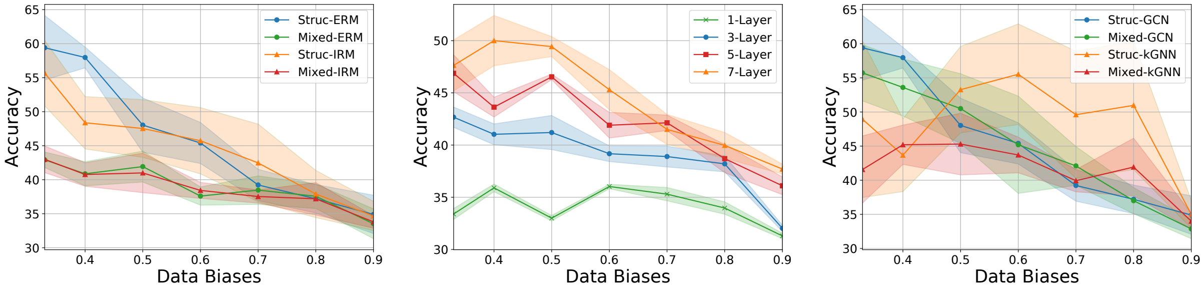
(a) Failures of training objectives. (b) Failures of deeper GNNs. (c) Failures of expressive GNNs.

Figure 7. Failure of existing methods on SPMotif with FIIF attribute shifts and graph size shifts.



(a) Failures of training objectives. (b) Failures of deeper GNNs. (c) Failures of expressive GNNs.

Figure 8. Failure of existing methods on SPMotif with PIIF attribute shifts.



(a) Failures of training objectives.

(b) Failures of deeper GNNs.

(c) Failures of expressive GNNs.

Figure 9. Failure of existing methods on SPMotif PIIF attribute shifts with graph size shifts.

To explore the behaviors of aforementioned methods against complicated distribution shifts on graphs, we first modify construction method in Wu et al. (2022c) to construct dataset for Fig. 6, where only FIIF structure-level spurious correlations are injected. Then we also inject FIIF attribute-level shifts, by setting the node attributes to constant vectors which is spuriously correlated with the labels. Furthermore, in Fig. 7, graph size shifts are added, which is exactly the SPMotif datasets used in DIR (Wu et al., 2022c). Besides, in Fig. 8, we can also change the FIIF attribute-level shifts to PIIF attribute-level shifts, where we flip the labels by a probability of 5% and let the flipped label to be spuriously correlated with the node features, following the PIIF SCM in Fig. 4. Graph size shifts can also be injected in this case, shown as Fig. 9. Next, we summarize our findings from the experiments.

Observation I: All existing methods are sensitive to distribution shifts. From the Fig. 6, 7, 8, 9, we can observe that *all* GNNs are sensitive to distribution shifts. As the intensity of spurious correlation grows, GNNs are more likely to overfit to shortcuts presented either in the structure-level or attribute-level, which is similar to general deep learning models (Geirhos et al., 2020).

Observation II: Higher variance also indicates unstable OOD performance. Although GNNs show certain robustness against single distribution shifts, e.g., performances do not decrease sharply at the beginning in Fig. 6, when the spurious correlation grows stronger, the OOD performance become more *unstable*, e.g., higher variance. The reason is that, GNNs sometimes can directly learn about the desired information at some random initializations, since the task is relatively simple compared to reality. Hence the performance will be highly sensitive to the quality of initialized points at the beginning. Consequently, the performances from multiple runs would exhibit high variance. However, when the task becomes more difficult, GNNs will consistently be prone to distribution shifts, and the variance will be smaller, as shown in experiments (Sec. 4).

Observation III: Entangling more distribution shifts can degenerate more GNN performance. As implied by the graph generation SCMs in Fig. 4, distribution shifts can happen at both structure-level and attribute-level, and each of them can have different type of spurious correlation with the label. In Fig. 6, we can find that, when the attribute-level distribution shifts are mixed, the performance will be worse and more unstable. When the graph size shifts are mixed, this phenomenon will be more obvious, as shown in Fig. 7. This phenomenon also verifies the observations in Knyazev et al. (2019) that attention mechanism in GNN is also sensitive to graph size shifts and can hardly learn the desired attention distributions without further guidance. Moreover, when the structure-level and attribute-level shifts have different spurious correlation types, i.e., when FIIF structure-level shifts and PIIF attribute-level shifts are both presented, the performance drop will be more serious, by comparing Fig. 6 to Fig. 8, as well as Fig. 7 to Fig. 9.

Observation IV: Using more powerful architectures can not improve the OOD performance. From the sub-figures (b) and (c) in Fig. 6, 7, 8, 9, we can also observe that neither adding more message passing turns nor using more expressive GNN architectures can be immune to distribution shifts. On the contrary, they also exhibit similar behaviors like basic GNN architectures. Specifically, adding more message passing runs show certain robustness against distribution shifts since they are more likely to learn the desired information during the optimization (Xu et al., 2021a). However, when the intensity of spurious correlation grows stronger, deeper GNNs are more likely to overfit to shortcuts hence their performances will drop more sharply. On the other hand, using provably more expressive GNN architectures can not improve the OOD performance, either. In Fig. 6, 7, 8, 9 we use 1-2-3-GNN following the algorithm of k-GNNs which is provably more expressive than 2-WL test (Morris et al., 2019). When there are no graph size shifts, k-GNNs will have higher performance at the beginning.

When there are graph size shifts, k-GNNs will have a lower initial performance at the beginning. Then, as the spurious strength grows, k-GNNs can suddenly become seriously unstable, though k-GNNs can have higher averaged performance, which reflects unsatisfactory OOD performance as Observation II implies. When the intensity of spurious correlations grows even stronger, similar to deeper GNNs, OOD performances of k-GNNs will be more unstable and go down to similar level as that of normal GNN architectures. Hence, it calls for better optimization objectives as well as a suitable architectures to help improve the OOD generalization performance.

Beyond the empirical studies in previous section, we aim to accompany more formal discussions for explaining the failures of existing optimization objectives and architectures in the next sections.

D.4. Theoretical discussions for failure case study in Sec. 2.2

A motivating example. To begin with, we follow Ahuja et al. (2021) to introduce a formal example on the failures of GNNs optimized with ERM or IRM (Vapnik, 1991; Arjovsky et al., 2019) via a linear binary classification problem:

Definition D.1 (Linear classification structural equation model (FIIF))**.**

$$\begin{aligned} &Y := (w^*_{\text{inv}} \cdot C) \oplus N,\ N \sim \text{Ber}(q),\ N \perp\!\!\!\perp (C,S), \\ &X \leftarrow S(C,S), \end{aligned}$$

where $w^*_{\text{inv}} \in \mathbb{R}^{n_c}$ with $\|w^*_{\text{inv}}\| = 1$ is the labeling hyperplane, $C \in \mathbb{R}^{n_c}$, $S \in \mathbb{R}^{n_s}$ are the corresponding invariant and varying latent variables, N is Bernoulli binary noise with a parameter of q and identical across all environments, $\oplus$ is the XOR operator, S is invertible.

Given data generation process as Assumption A.1, and latent space interaction as Assumption A.2 or A.3, and strictly separable invariant features A.5, consider a k-layer linearized GNN $\rho \circ h$ using mean as READOUT for binary graph classification, if $\cup_{e\in\mathcal{E}_{\text{te}}}\text{supp}(\mathbb{P}^e) \not\subseteq \cup_{e\in\mathcal{E}_{\text{tr}}}\text{supp}(\mathbb{P}^e)$:

(i) For graphs features generated as Definition D.1, $\rho \circ h$ optimized with ERM or IRM will fail to generalize OOD (Eq. 11) almost surely;

(ii) For graphs with more than two nodes, globally same node features generated as Definition D.1, and graph labels that are the same as global node labels, $\rho \circ h$ optimized with ERM or IRM will fail to generalize OOD (Eq. 11) almost surely;

For graph classification, if the number of nodes is fixed to one, it covers the linear classification as above. When $\cup_{e\in\mathcal{E}_{\text{te}}}\text{supp}(\mathbb{P}^e) \not\subseteq \cup_{e\in\mathcal{E}_{\text{tr}}}\text{supp}(\mathbb{P}^e)$, it implies the S from training environments $\mathcal{E}_{\text{tr}}$ does not cover S from testing environments, while C can be covered. Moreover, the condition of strictly separable training data now can be formulated as $\min_{C\in\cup_{e\in\mathcal{E}_{\text{tr}}}(C\subseteq G^e)} \text{sgn}(w^*_{\text{inv}} \cdot C)(w^*_{\text{inv}} \cdot C) > 0$. Recall that ERM trains the model by minimizing the empirical risk (e.g., 0-1 loss) over all training data, and IRM formulates OOD generalization as:

$$\begin{aligned} &\min_{\theta, f_c} \frac{1}{|\mathcal{E}_{\text{tr}}|} \sum_{e\in\mathcal{E}_{\text{tr}}} R^e(h \circ \rho) \\ &\text{s.t. } \rho \in \arg\min_{\hat{\rho}} R^e(h \circ \hat{\rho}),\ \forall e \in \mathcal{E}_{\text{tr}}. \end{aligned} \tag{13}$$

However, both ERM and IRM can not enable OOD generalization, i.e., finding the ground truth w^*_{inv}, following the Theorem 3 from Ahuja et al. (2021):

Theorem D.2 (Insufficiency of ERM and IRM)**.** *Suppose each $e \in \mathcal{E}_{all}$ follows Definition. D.1, C are strictly separable, bounded and satisfy the support overlap between $\mathcal{E}_{tr}$ and $\mathcal{E}_{te}$, and S are bounded, if S does not support the overlap, then both ERM and IRM fail at solving the OOD generalization problem.*

The reason is that, when C from all environments are strictly separable, there can be infinite many Bayes optimal solutions given training data $\{G^e, y^e\}_{e\in\mathcal{E}_{\text{tr}}}$, while there is only one optimal solution that does not rely on S. Hence, the probability of generalization to OOD (finding the optimal solution) tends to be 0 in probability.

As for case (ii), when the GNN uses mean readout to classify more than one node graphs, assuming the graph label is determined by the node label and all of the nodes have the same label that are determined as Definition D.1, then GNN optimized with ERM and IRM will also fail because of the same reasons as case (i).

Discussions on the failures of previous OOD related solutions. First of all, for IRM or similar objectives (Sagawa* et al., 2020; Krueger et al., 2021; Ahuja et al., 2021; Bellot & van der Schaar, 2020) that require environment information or non-trivial data partitions, they can hardly be applied to graphs due to the lack of such information. The reason is that obtaining such information can be expensive due to the abstraction of graphs. Moreover, as proved in Theorem 5.1 of Rosenfeld et al. (2021), when there is not sufficient support overlap between training environments and testing environments, the IRM or similar objectives can fail catastrophically when being applied to non-linear regime. The only OOD objective EIIL (Creager et al., 2021) that does not require environment labels, also rely on similar assumptions on the support overlap. We also empirically verify their failing behaviors in our experiments.

Moreover, since part of explainability works also try to find a subset of the inputs for interpretable prediction robustly against distribution shifts. Here we also provide a discussion for these works. The first work following this line is INVRAT (Chang et al., 2020), which develops an information-theoretic objective (we re-formulate it to suit with OOD generalization problem on graphs):

$$\min_{g,f_c}\max_{f_s} R(g\circ f_c,Y)+\lambda h(R(g\circ f_c,Y)-R_e(g\circ f_s,Y,E)). \tag{14}$$

However, it also requires extra environment labels for optimization that are often unavailable in graphs. Besides, the corresponding assumption on the data generation for guaranteed performance is essentially PIIF if applied to our case, while it can not provide any theoretical guarantee on FIIF.

We also notice a recent work, DIR (Wu et al., 2022c), as a generalization of INVRAT to graphs while studying FIIF spurious correlations, that proposes an alternative objective which does not require environment label:

$$\min \mathbb{E}_s[R(h,Y|\text{do}(S=s))]+\lambda\text{var}_s(\{R(h,Y|\text{do}(S=s))\}). \tag{15}$$

However, the theoretical justification established for DIR (Theorem 1 to Corollary 1 in Wu et al. (2022c)) essentially depends on the quality of the generator g which can be prone to spurious correlations. Thus, DIR can hardly provide any theoretical guarantees when applied to our case, neither for FIIF nor PIIF. In experiments, we empirically find the unstable and relatively high sensitivity of DIR to spurious correlations, which verifies our finding. More details about empirical behaviors of DIR can be found in Appendix G.

In contrast to DIR, GIB (Yu et al., 2021) that focuses on discovering a informative subgraph for explanation, essentially can provide theoretical guarantees for FIIF spurious correlations. Theoretically, (we copy the discussion in Appendix F here to provide an overview of relationships between GIB and DIR.) Under the FIIF assumption on latent interaction, the independence condition derived from causal model can also be rewritten as $Y \perp\!\!\!\perp S|C$ (similar to that in DIR (Wu et al., 2022c) as they also focus on FIIF), which further implies $Y \perp\!\!\!\perp S|\hat{G}_c$. Hence it is natural to use Information Bottleneck (IB) objective (Tishby et al., 1999) to solve for G_c:

$$\begin{aligned}&\min_{f_c,g} R_{G_c}(f_c(\hat{G}_c)),\\ &\text{s.t. } G_c=\operatorname*{arg\,max}_{\hat{G}_c=g(G)\subseteq G} I(\hat{G}_c,Y)-I(\hat{G}_c,\mathcal{G}),\end{aligned} \tag{16}$$

which explains the success of many existing works in finding predictive subgraph through IB (Yu et al., 2021). However, the estimation of $I(\hat{G}_c,G)$ is notoriously difficult due to the complexity of graph, which can lead to unstable convergence as observed in our experiments. In contrast, optimization with contrastive objective in GOOD as Eq. 8 induces more stable convergence.

E. Theory and Discussions

In this section, we provide proofs for propositions and theorems mentioned in the main paper.

E.1. Challenges of OOD generalization on graphs.

From the aforementioned analysis, we can summarize some key challenges revealed by the failures of both existing optimization objectives and GNN architectures. In particular, we are facing two main challenges a) Distribution shifts on graphs are more complicated where different types of spurious correlations can be entangled via different graph properties; b) Environment labels are usually not available due to the abstract graph data structure.

E.2. Proof for theorem B.2 (i)

Theorem E.1 (GOODv1 Induces Invariant GNNs)**.** *Given a set of graph datasets $\{\mathcal{D}^e\}_e$ and environments $\mathcal{E}_{all}$ that follow the same graph generation process in Sec. 2.1, assuming that* (a) f_{gen}^{G} *and* $f_{gen}^{G_c}$ *in Assumption A.1 are invertible,* (b) *samples from each training environment are equally distributed, i.e.,*$|\mathcal{D}_{\hat{e}}| = |\mathcal{D}_{\tilde{e}}|,\ \forall \hat{e}, \tilde{e} \in \mathcal{E}_{tr}$*, if* $\forall G_c, |G_c| = s_c$*, then a GNN* $f_c \circ g$ *solves Eq. 7, is an invariant GNN (Def. B.1).*

Proof. We re-write the objective as follows:

$$\max_{f_c, g} I(\hat{G}_c; Y), \text{ s.t. } \hat{G}_c \in \underset{\hat{G}_c = g(G), |\hat{G}_c| \leq s_c}{\arg\max} I(\hat{G}_c; \tilde{G}_c | Y), \tag{17}$$

where $\hat{G}_c = g(G), \tilde{G}_c = g(\tilde{G})$ and $\tilde{G} \sim \mathbb{P}(G|Y)$, i.e., $\tilde{G}$ and G have the same label.

The proof of Theorem E.1 is essentially to show the estimated $\hat{G}_c$ through Eq. 17 is the underlying G_c, then the maximizer of $I(\hat{G}_c; Y)$ in Eq. 17 can produce most informative and stable predictions about Y based on G, hence is an invariant GNN according to Definition. B.1.

In the next, we are going to take a information-theoretic view of the first term $I(\hat{G}_c; Y)$ and the second term $I(\hat{G}_c; \tilde{G}_c | Y)$ to conclude the proof. We begin by introducing the following lemma:

Lemma E.2. *Given the same conditions as Thm. E.1,* $I(\hat{G}_c; Y)$ *is maximized if and only if* $I(\hat{G}_c; Y | E = e)$ *is maximized,* $\forall e \in \mathcal{E}_{tr}$.

The proof for Lemma E.2 is straightforward, given the condition that samples from each training environment are equally distributed, i.e.,$|\mathcal{D}_{\hat{e}}| = |\mathcal{D}_{\tilde{e}}|,\ \forall \hat{e}, \tilde{e} \in \mathcal{E}_{\text{tr}}$. Obviously, $\hat{G}_c = G_c$ is a maximizer of $I(\hat{G}_c; Y) = I(C; Y) = H(Y)$, since $f_{\text{gen}}^{c} : \mathcal{C} \to \mathcal{G}_c$ is invertible and C causes Y. However, there might be some subset $G_s^p \subseteq G_s$ from the underlying G_s that entail the same information about label, i.e., $I(G_c^p \cup G_s^p; Y) = I(G_c; Y)$ where $\hat{G}_c = G_c^p \cup G_s^p$ and $G_c^p = G_c \cap \hat{G}_c$. For FIIF (Assumption 4(b)), it can not happen otherwise, let $G_c^l = G_c - G_c^p$, then we have:

$$\begin{aligned} I(\hat{G}_c; Y) = I(G_c^p \cup G_s^p; Y) &= I(G_c^p \cup G_c^l; Y) = I(G_c; Y) \\ I(G_c^p; Y) + I(G_s^p; Y | G_c^p) &= I(G_c^p; Y) + I(G_c^l; Y | G_c^p) \\ I(G_s^p; Y | G_c^p) &= I(G_c^l; Y | G_c^p) \\ H(Y|G_c^p) - H(Y|G_c^p, G_s^p) &= H(Y|G_c^p) - H(Y|G_c^p, G_c^l) \\ H(Y|G_c^p) - H(Y|G_c^p, G_s^p) &= H(Y|G_c^p), \\ H(Y|G_c^l, G_s^p) &= 0, \end{aligned} \tag{18}$$

where the second last equality is due to $C \to Y$ and the invertibility of $f_{\text{gen}}^{c} : \mathcal{C} \to \mathcal{G}_c$ in FIIF, i.e., $H(Y|C) = H(Y|G_c) = H(Y|G_c^p, G_c^l) = 0$. However, in PIIF, it can not hold since conditioning on G_c^p, G_s^p can not determine Y, since $S \not\perp Y | C$. In other words, $G_s \not\perp Y | G_c$, which means G_s can imply some information about Y that is equivalent to $I(G_c^l; Y | G_c^p)$.

To avoid the presence of spuriously correlated G_s in $\hat{G}_c$, we will use the second term to eliminate it:

$$\begin{aligned} &\max_{f_c, g} I(\hat{G}_c; \tilde{G}_c | Y), \\ &\quad = H(\hat{G}_c | Y) - H(\hat{G}_c | \tilde{G}_c, Y), \end{aligned} \tag{19}$$

where $\hat{G}_c = g(G)$, $\tilde{G}_c = g(\tilde{G})$ are two positive samples drawn from the same class (i.e., condition on the same Y). Since the all of the training environments are equally distributed, maximizing $I(\hat{G}_c; \tilde{G}_c | Y)$ is essentially maximizes $I(\hat{G}_c, E = \hat{e}; \tilde{G}_c, E = \tilde{e} | Y), \forall \hat{e}, \tilde{e} \in \mathcal{E}_{\text{tr}}$. Hence, we have:

$$\begin{aligned} &\max_{f_c, g} I(\hat{G}_c; \tilde{G}_c | Y), \\ &\quad = I(\hat{G}_c, E = \hat{e}; \tilde{G}_c, E = \tilde{e} | Y) \\ &\quad = H(\hat{G}_c, E = \hat{e} | Y) - H(\hat{G}_c, E = \hat{e} | \tilde{G}_c, E = \tilde{e}, Y). \end{aligned} \tag{20}$$

We claim Eq. 20 can eliminate any potential subsets in the estimated $\hat{G}_c$.

Otherwise, suppose there are some subsets $\hat{G_s^p} \subseteq \hat{G}_s$ and $\tilde{G_s^p} \subseteq \tilde{G}_s$ contained in the estimated $\hat{G}_c, \tilde{G}_c$, where $\hat{G}_s, \tilde{G}_s$ be the corresponding underlying G_ss for $\hat{G}_c, \tilde{G}_c$. Let $\hat{G}_c^*$ and $\tilde{G}_c^*$ be the ground truth invariant subgraph G_cs of $\hat{G}$ and $\tilde{G}$, $\hat{G_c^l} = \hat{G}_c^* - \hat{G}_c$ and $\tilde{G_c^l} = \tilde{G}_c^* - \tilde{G}_c$ be the **l**eft (un-estimated) subsets from corresponding ground truth G_cs, and $\hat{G_c^p} = \hat{G}_c^* - \hat{G_c^l}$ and $\tilde{G_c^p} = \tilde{G}_c^* - \tilde{G_c^l}$ be the complement, or equivalently, the **p**artial $\hat{G}_c^*, \tilde{G}_c^*$ that are estimated in $\hat{G}_c, \tilde{G}_c$, respectively. We can also define similar counterparts for G_s: $\hat{G_s^p}, \tilde{G_s^p}$ are the partial $\hat{G}_s, \tilde{G}_s$s contained in the estimated $\hat{G}_c, \tilde{G}_c$ while $\hat{G_s^l}, \tilde{G_s^l}$ are the left subsets $\hat{G}_s, \tilde{G}_s$, respectively.

Recall the constraint that $|G_c| = s_c$, hence if $\hat{G_c^p} \subseteq \hat{G}_c$, then a corresponding $\hat{G_c^l} = \hat{G}_c^* - \hat{G_c^p}$ will be replaced by $\hat{G_s^p}$ in $\hat{G}_c$. In this case, we have:

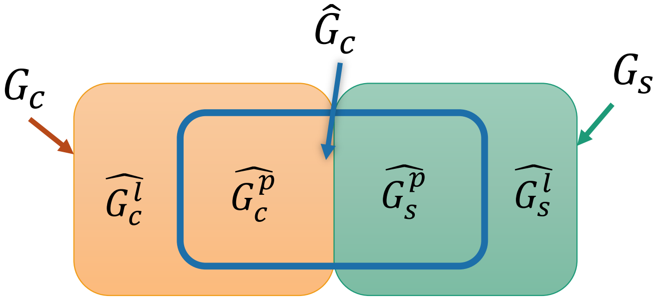


Figure 10. Illustration of the notation. G_c and G_s are two disjoint sets. $\hat{G}_c$ may contain certain subsets from G_c and G_s. The subsets from G_c and G_s contained in $\hat{G}_c$ are denoted as $\hat{G_c^p}$ and $\hat{G_s^p}$, respectively. While the left subsets in G_c and G_s are denoted as $\hat{G_c^l}$ and $\hat{G_s^l}$, respectively.

$$\begin{aligned} H(\hat{G}_c, E=\hat{e}|Y) &= H(E=\hat{e}|\hat{G}_c, Y) + H(\hat{G}_c|E=\hat{e}, Y) \\ &= H(\hat{G_c^p} \cup \hat{G_s^p}|E=\hat{e}, Y) \\ &= H(\hat{G_c^p}|E=\hat{e}, Y) + H(\hat{G_s^p}|\hat{G_c^p}, E=\hat{e}, Y) \end{aligned} \tag{21}$$

where the second equality is due to $E=\hat{e}$ is determined so that $H(E=\hat{e}|\hat{G}_c, Y) = 0$. Compared Eq. 21 to that when $\hat{G}_c = \hat{G}_c^*$, we have the entropy change as:

$$\begin{aligned} \Delta H(\hat{G}_c, E=\hat{e}|Y) &= H(\hat{G}_c, E=\hat{e}|Y) - H(\hat{G}_c^*, E=\hat{e}|Y), \\ &= H(\hat{G_s^p}|\hat{G_c^p}, E=\hat{e}, Y) - H(\hat{G_c^l}|\hat{G_c^p}, E=\hat{e}, Y). \end{aligned} \tag{22}$$

Let $\epsilon = H(\hat{G_s^p}|\hat{G_c^p}, E=\hat{e}, Y)$. In a idealistic setting, when the noise of the generation process $S := f_{\text{spu}}(Y, E)$ in PIIF tends to be 0, i.e., $\epsilon \to 0$, S is determined conditioned on E, Y, hence G_s and any subsets of G_s are all determined. Then, it suffices to know that in Eq. 22, $H(\hat{G_s^p}|\hat{G_c^p}, E=\hat{e}, Y) = 0$ while $H(\hat{G_c^l}|\hat{G_c^p}, E=\hat{e}, Y) > 0$ since $\hat{G_c^l}$ can not be determined when given $\hat{G_c^p}, E=\hat{e}, Y$. Thus, when some subset from G_s is included in $\hat{G}_c$, it will minimize $H(\hat{G}_c, E=\hat{e}|Y)$. In the next, we will show how $\epsilon = H(\hat{G_s^p}|\hat{G_c^p}, E=\hat{e}, Y)$ can be cancelled thus leading to a smaller $H(\hat{G}_c, E=\hat{e}|Y)$, by considering the second term $H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y)$.

As for $H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y)$, without loss of generality, we can divide all of the possible cases into two:

(i) One of $\hat{G}_c$ and $\tilde{G}_c$ contains some subset of G_s, i.e., $\hat{G}_c$ contains some $\hat{G_s^p} \subseteq \hat{G}_s$;

(ii) Both $\hat{G}_c$ and $\tilde{G}_c$ contain some $\hat{G_s^p} \subseteq \hat{G}_s$ and $\tilde{G_s^p} \subseteq \tilde{G}_s$, respectively.

For (i), we have:

$$\begin{aligned} H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y) &= H(\hat{G_c^p}, \hat{G_s^p}, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y) \\ &= H(\hat{G_s^p}|\tilde{G}_c, E=\tilde{e}, Y, \hat{G_c^p}, E=\hat{e}) + H(\hat{G_c^p}, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y), \end{aligned} \tag{23}$$

Thus, $H(\tilde{G_s^p}|\hat{G}_c, E=\tilde{e}, Y, \tilde{G_c^l}, E=\hat{e}) = 0$ and Thus, we have:

$$\begin{aligned} \Delta H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y) &= H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y) - H(\hat{G}_c^*, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y), \\ &= H(\tilde{G_s^p}|\hat{G}_c, E=\tilde{e}, Y, \tilde{G_c^l}, E=\hat{e}) \\ &\quad - H(\hat{G_c^l}|\tilde{G}_c, E=\tilde{e}, Y, \hat{G_c^p}, E=\hat{e}). \end{aligned} \tag{24}$$

Combing $\Delta H(\hat{G}_c, E=\hat{e}|Y)$, we have:

$$\begin{aligned}\Delta I(\hat{G}_c, E=\hat{e}; \tilde{G}_c, E=\tilde{e}|Y) &= \Delta H(\hat{G}_c, E=\hat{e}|Y) - \Delta H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y) \\ &= \left\{ H(\hat{G}_s^p|\hat{G}_c^p, E=\hat{e}, Y) - H(\hat{G}_s^p|\tilde{G}_c, E=\tilde{e}, Y, \hat{G}_c^p, E=\hat{e}) \right\} \\ &\quad + \left\{ -H(\hat{G}_c^l|\hat{G}_c^p, E=\hat{e}, Y) + H(\hat{G}_c^l|\tilde{G}_c, E=\tilde{e}, Y, \hat{G}_c^p, E=\hat{e}) \right\}, \\ &= -H(\hat{G}_c^l|\hat{G}_c^p, E=\hat{e}, Y) + H(\hat{G}_c^l|\tilde{G}_c, E=\tilde{e}, Y, \hat{G}_c^p, E=\hat{e}),\end{aligned} \tag{25}$$

where the last equality is because of the independence of $\hat{G}_s^p$ between $\tilde{G}_c, E=\tilde{e}$ conditioned on $Y, E=\hat{e}$. Since conditioning will lower the entropy for both discrete and continuous variables (Cover & Thomas, 2006; Yeung, 2008), we have:

$$\Delta I(\hat{G}_c, E=\hat{e}; \tilde{G}_c, E=\tilde{e}|Y) < 0, \tag{26}$$

which implies the existence of $\hat{G}_s^p$ in $\hat{G}_c$ will lower down the second term in Eq. 17 for the case (i).

For (ii), we have:

$$\begin{aligned}H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y) &= H(\hat{G}_c^p, \hat{G}_s^p, E=\hat{e}|\tilde{G}_c^p, \tilde{G}_s^p, E=\tilde{e}, Y) \\ &= H(\hat{G}_s^p|\tilde{G}_c^p, \tilde{G}_s^p, E=\tilde{e}, Y, \hat{G}_c^p, E=\hat{e}) \\ &\quad + H(\hat{G}_c^p, E=\hat{e}|\tilde{G}_c^p, \tilde{G}_s^p, E=\tilde{e}, Y),\end{aligned} \tag{27}$$

Similar to (i), $H(\hat{G}_s^p|\tilde{G}_c^p, \tilde{G}_s^p, E=\tilde{e}, Y, \hat{G}_c^p, E=\hat{e})$ can be cancelled out with $H(\hat{G}_s^p|\hat{G}_c^p, E=\hat{e}, Y)$. Then, we have:

$$\begin{aligned}\Delta I(\hat{G}_c, E=\hat{e}; \tilde{G}_c, E=\tilde{e}|Y) &= \Delta H(\hat{G}_c, E=\hat{e}|Y) - \Delta H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y) \\ &= -H(\hat{G}_c^l|\hat{G}_c^p, E=\hat{e}, Y) + H(\hat{G}_c^l|\tilde{G}_c^p, \tilde{G}_s^p, E=\tilde{e}, \hat{G}_c^p, Y, E=\hat{e}).\end{aligned} \tag{28}$$

Since additionally conditioning on $\hat{G}_s^p$ in $H(\hat{G}_c^l, E=\hat{e}|\tilde{G}_c^p, \tilde{G}_s^p, E=\tilde{e}, Y)$ can not lead to new information about $\hat{G}_c^l$, we have:

$$\begin{aligned}H(\hat{G}_c^l|\tilde{G}_c^p, \tilde{G}_s^p, E=\tilde{e}, \hat{G}_c^p, Y, E=\hat{e}) &= H(\hat{G}_c^l|\tilde{G}_c^p, E=\tilde{e}, \hat{G}_c^p, Y, E=\hat{e}) \\ &< H(\hat{G}_c^l|\hat{G}_c^p, Y, E=\hat{e}),\end{aligned} \tag{29}$$

which follows that $\Delta I(\hat{G}_c, E=\hat{e}; \tilde{G}_c, E=\tilde{e}|Y) < 0$.

To summarize, the ground truth G_c is the only maximizer of the objective (Eq. 17), hence solving for the objective (Eq. 17) can elicit an invariant GNN.

E.3. Proof for theorem B.2 (ii)

Theorem E.3 (GOODv2 Induces Invariant GNNs)**.** *Given a set of graph datasets $\{\mathcal{D}^e\}_e$ and environments $\mathcal{E}_{all}$ that follow the same graph generation process in Sec. 2.1, assuming that* (a) *f_{gen}^G and $f_{gen}^{G_c}$ in Assumption A.1 are invertible,* (b) *samples from each training environment are equally distributed, i.e.,$|\mathcal{D}_{\hat{e}}| = |\mathcal{D}_{\tilde{e}}|$, $\forall \hat{e}, \tilde{e} \in \mathcal{E}_{tr}$, a GNN $f_c \circ g$ solves Eq. 7, is an invariant GNN (Def. B.1).*

Proof. We re-write the objective as follows:

$$\begin{aligned}\max_{f_c, g} \ I(\hat{G}_c; Y) + I(\hat{G}_s; Y), \ \text{s.t.}\ \hat{G}_c \in & \mathop{\arg\max}_{\hat{G}_c = g(G), \tilde{G}_c = g(\tilde{G})} \ I(\hat{G}_c; \tilde{G}_c|Y), \\ & I(\hat{G}_s; Y) \leq I(\hat{G}_c; Y), \ \hat{G}_s = G - g(G).\end{aligned} \tag{30}$$

where $\hat{G}_c = g(G), \tilde{G}_c = g(\tilde{G})$ and $\tilde{G} \sim \mathbb{P}(G|Y)$, i.e., $\tilde{G}$ and G have the same label.

Similar to the proof for Theorem E.1, to prove Theorem E.3 is essentially to show the estimated $\hat{G}_c$ through Eq. 30 is the underlying G_c, hence the minimizer of Eq. 30 elicits an invariant GNN predictor.

In the next, we also begin with a lemma:

Lemma E.4. *Given data generation process as Theorem E.3, for both FIIF and PIIF, we have:*

$$I(C;Y) \geq I(S;Y),$$

hence $I(G_c;Y) \geq I(G_s;Y)$.

Proof for Lemma E.4. For both FIIF and PIIF, Assumption A.5 implies that $H(C|Y) \leq H(S|Y)$. It follows that $I(C;Y) = H(Y) - H(C|Y) \geq H(Y) - H(S|Y) = I(S;Y)$. Then, since $f_{\text{gen}}^{G_c} : \mathcal{C} \to \mathcal{G}_c$ is invertible, we have $I(G_c;Y) = I(C;Y) \geq I(S;Y) \geq I(G_s;Y)$. □

Given Lemma E.4, we know $\hat{G}_c$ at least contains some subset of the underlying G_c, otherwise the constraint $I(\hat{G}_s;Y) \leq I(\hat{G}_c;Y)$ will be violated since $G_c \subseteq \hat{G}_s$ in this case.

Assuming there are some subset of G_s contained in $\hat{G}_c$, without loss of generality, we can divide all of the possible cases about $\hat{G}_c$ into two:

(i) $\hat{G}_c$ only contains a subset of the underlying G_c;

(ii) $\hat{G}_c$ contains a subset of the underlying G_c as well as part of the underlying G_s;

Before the discussion, let us inherit the notations of subsets of G_c, G_s from the proof for Theorem E.1: Let $\hat{G}_c^*$ and $\tilde{G}_c^*$ be the ground truth invariant subgraph G_cs of $\hat{G}$ and $\tilde{G}$, $\hat{G}_c^l = \hat{G}_c^* - \hat{G}_c$ and $\tilde{G}_c^l = \tilde{G}_c^* - \tilde{G}_c$ be the **l**eft (un-estimated) subsets from corresponding ground truth G_cs, and $\hat{G}_c^p = \hat{G}_c^* - \hat{G}_c^l$ and $\tilde{G}_c^p = \tilde{G}_c^* - \tilde{G}_c^l$ be the complement, or equivalently, the **p**artial $\hat{G}_c^*, \tilde{G}_c^*$ that are estimated in $\hat{G}_c, \tilde{G}_c$, respectively. Similarly, $\hat{G}_s^p, \tilde{G}_s^p$ are the partial $\hat{G}_s, \tilde{G}_s$s contained in the estimated $\hat{G}_c, \tilde{G}_c$ while $\hat{G}_s^l, \tilde{G}_s^l$ are the left subsets $\hat{G}_s, \tilde{G}_s$, respectively.

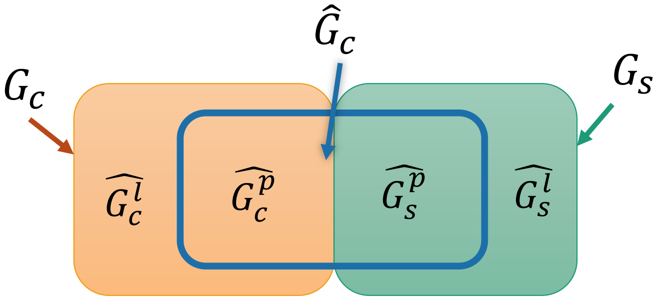


Figure 11. Illustration of the notation for estimated $\hat{G}_c$ from G. G_c and G_s are two disjoint sets. $\hat{G}_c$ may contain certain subsets from G_c and G_s. The subsets from G_c and G_s contained in $\hat{G}_c$ are denoted as $\hat{G}_c^p$ and $\hat{G}_s^p$, respectively. While the left subsets in G_c and G_s are denoted as $\hat{G}_c^l$ and $\hat{G}_s^l$, respectively. Similar notations are also applicable for the estimated $\tilde{G}_c$ from $\tilde{G}$.

First of all, case (i) cannot hold because, when maximizing $I(\hat{G}_c;\tilde{G}_c|Y)$, if $\exists \hat{G}_c^l = \hat{G}_c^* - \hat{G}_c$, as shown in the proof for Theorem E.1, including $\hat{G}_c^l$ into $\hat{G}_c$ can always enlarge $I(\hat{G}_c;\tilde{G}_c|Y)$, while not affecting the optimality of $I(\hat{G}_s;Y) + I(\hat{G}_c;Y)$ by re-distributing $\hat{G}_c^l$ from $\hat{G}_s$ to $\hat{G}_c$. Consequently, $\hat{G}_c^*$ must be included in $\hat{G}_c$, i.e., $\hat{G}_c^* \subseteq \hat{G}_c$.

As for case (ii), recall that, by the condition of equally distributed training samples from each training environment, maximizing $I(\hat{G}_c;\tilde{G}_c|Y)$ is essentially maximizing $I(\hat{G}_c, E=\hat{e};\tilde{G}_c, E=\tilde{e}|Y), \forall \hat{e},\tilde{e} \in \mathcal{E}_{\text{tr}}$, hence, we have:

$$\begin{aligned}
&\max_{g,f_c} I(\hat{G}_c;\tilde{G}_c|Y), \\
&\quad = I(\hat{G}_c, E=\hat{e};\tilde{G}_c, E=\tilde{e}|Y) \\
&\quad = H(\hat{G}_c, E=\hat{e}|Y) - H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y).
\end{aligned} \tag{31}$$

We claim Eq. 31 can eliminate any potential subsets in the estimated $\hat{G}_c$. Similarly, we have:

$$\begin{aligned}
H(\hat{G}_c, E=\hat{e}|Y) &= H(E=\hat{e}|\hat{G}_c, Y) + H(\hat{G}_c|E=\hat{e}, Y) \\
&= H(\hat{G}_c^* \cup \hat{G}_s^p|E=\hat{e}, Y) \\
&= H(\hat{G}_c^*|E=\hat{e}, Y) + H(\hat{G}_s^p|\hat{G}_c^*, E=\hat{e}, Y) \\
&= H(\hat{G}_c^*|Y) + H(\hat{G}_s^p|\hat{G}_c^*, E=\hat{e}, Y)
\end{aligned} \tag{32}$$

where the second equality is due to $E=\hat{e}$ is determined. Compared to the case that $\hat{G}_c=\hat{G}_c^*$, we have:

$$\begin{aligned}\Delta H(\hat{G}_c, E=\hat{e}|Y) &= H(\hat{G}_c, E=\hat{e}|Y) - H(\hat{G}_c^*, E=\hat{e}|Y),\\ &= H(\hat{G}_s^p|\hat{G}_c^*, E=\hat{e}, Y).\end{aligned} \tag{33}$$

Then, as for $H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y)$, without loss of generality, we can divide all of the possible cases into two:

(a) $\hat{G}_c$ contains some $\hat{G}_s^p \subseteq \hat{G}_s$;

(b) Both $\hat{G}_c$ and $\tilde{G}_c$ contain some $\hat{G}_s^p \subseteq \hat{G}_s$ and $\tilde{G}_s^p \subseteq \tilde{G}_s$, respectively.

For (a), we have:

$$\begin{aligned}H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y) &= H(\hat{G}_c^*, \hat{G}_s^p, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y)\\ &= H(\hat{G}_s^p|\tilde{G}_c, E=\tilde{e}, Y, \hat{G}_c^*, E=\hat{e}) + H(\hat{G}_c^*, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y),\end{aligned} \tag{34}$$

Similarly to the proof for Theorem E.1, when considering $\Delta I(\hat{G}_c;\tilde{G}_c|Y)$, the effects of $H(\hat{G}_s^p|\tilde{G}_c, E=\tilde{e}, Y, \hat{G}_c^*, E=\hat{e})$ is cancelled out by $H(\hat{G}_s^p|\hat{G}_c^*, E=\hat{e}, Y)$. Hence, we have:

$$\Delta I(\hat{G}_c;\tilde{G}_c|Y)=0.$$

For (b), we have:

$$\begin{aligned}H(\hat{G}_c, E=\hat{e}|\tilde{G}_c, E=\tilde{e}, Y) &= H(\tilde{G}_c^*, \tilde{G}_s^p, E=\hat{e}|\tilde{G}_c^*, \tilde{G}_s^p, E=\tilde{e}, Y)\\ &= H(\hat{G}_s^p|\tilde{G}_c^*, \tilde{G}_s^p, E=\tilde{e}, Y, \hat{G}_c^*, E=\hat{e})\\ &\quad + H(\hat{G}_c^*|\tilde{G}_c^*, \tilde{G}_s^p, E=\tilde{e}, Y, E=\hat{e}),\end{aligned} \tag{35}$$

Similarly, $H(\hat{G}_s^p|\tilde{G}_c^*, \tilde{G}_s^p, E=\tilde{e}, Y, \hat{G}_c^*, E=\hat{e})=0$ can also be cancelled out by $H(\hat{G}_s^p|\hat{G}_c^*, E=\hat{e}, Y)$. Moreover, for $H(\hat{G}_c^*|\tilde{G}_c^*, \tilde{G}_s^p, E=\tilde{e}, Y, E=\hat{e})$, $\tilde{G}_s^p$ can not bring no additional information about $\hat{G}_c^*$, when conditioning on $\tilde{G}_c^*, Y, E=\tilde{e}$. Hence, we also have:

$$\Delta I(\hat{G}_c;\tilde{G}_c|Y)=0.$$

To summarize, when maximizing $I(\hat{G}_c;\tilde{G}_c|Y)$, including any $\hat{G}_s^p \subseteq \hat{G}_s^*$ can not bring additional benefit while affecting the optimality of $I(\hat{G}_s;Y)+I(\hat{G}_c;Y)$. More specifically, when considering the changes to $I(\hat{G}_s;Y)+I(\hat{G}_c;Y)$, $\forall G_s^p \subseteq G_s$, we have

$$I(G-\hat{G}_c^*-G_s^p;Y) \leq I(G-\hat{G}_c^*;Y),\ \forall G_s^p \subseteq G_s,$$

while $I(Y;\hat{G}_c^*, G_s^p)=I(Y;\hat{G}_c^*)+I(Y;\hat{G}_s^p|\hat{G}_c^*),\ \forall e \in \mathcal{E}_{\text{tr}}$. Consequently,

$$\begin{aligned}\Delta I(\hat{G}_s;Y)+I(\hat{G}_c;Y) &= -I(\hat{G}_s^p;Y|\hat{G}_s^l)+I(\hat{G}_s^p;Y|\hat{G}_c^*)\\ &= -I(\hat{G}_s^p;Y)+I(\hat{G}_s^p;Y|\hat{G}_c^*) \leq 0.\end{aligned} \tag{36}$$

Hence, only the underlying G_c is the solution to Eq. 30, which implies that solving for the objective (Eq. 30) can elicit an invariant GNN.

F. Details of Prototypical GOOD Implementation

In fact, the GOOD framework introduced in Sec. 3 can have multiple implementations. We choose interpretable architectures in our experiments for the purpose of concept verification. More sophisticated architectures can be incorporated. Experimental results in Sec. 4 also demonstrates that, even equipped with basic GNN architectures, GOOD already has the excellent OOD generalization ability, hence it is promising to incorporate more advanced architectures from the prosperous GNN literature.

We now introduce the details of the architectures used in our experiments. Recall that GOOD decomposes a GNN model for graph classification into two modules, i.e., a featurizer: $g : \mathcal{G} \to \mathcal{G}_c$ and a classifier $f_c : \mathcal{G}_c \to \mathcal{Y}$. Specifically, for the implementation of Featurizer, we choose one of the common practices GAE (Kipf & Welling, 2016) for calculating the sampled weights for each edge. More formally, the soft mask is predicted through the following equation:

$$Z = \text{GNN}(G) \in \mathbb{R}^{n\times h},\ M = \sigma(ZZ^T) \in \mathbb{R}^{n\times n}.$$

If a sampling ratio s_c is predetermined, we sample s_c of total edges with the largest predicted weights as a soft estimation of $\hat{G}_c$. Then, the estimated $\hat{G}_c$ will be forwarded to the classifier f_c for predicting the labels of the original graph. Although Theorem E.1 assumes s_c is known, in real applications we do not know the specific s_c. Hence, in experiments, we select s_c according to the validation performance. To thoroughly study the effects of $I(\hat{G}_s;Y)$ comparing to GOODv1, we stick to using the same s_c and sampling process for GOODv2, while GOODv2 essentially requires less specific knowledge about ground truth r_c hence achieving better empirical performance. Moreover, once the sampled edges are determined, the classifier GNN can take either the original feature of the input graph or the learned feature from the featurizer as the new node attributes for $\hat{G}_c$. We select the architecture according to the validation performance from some random runs.

For the implementation of the information theoretic objectives, we will use GOODv2 for elaboration while the implementation of GOODv1 can be obtained via removing the third term from GOODv2. Recall that GOODv2 has the following formulation:

$$\begin{aligned}\max_{f_c,g}\ I(\hat{G}_c;Y) + I(\hat{G}_s;Y),\ \text{s.t. } \hat{G}_c \in &\mathop{\arg\max}_{\hat{G}_c=g(G),\tilde{G}_c=g(\tilde{G})} I(\hat{G}_c;\tilde{G}_c|Y),\\ &I(\hat{G}_s;Y) \le I(\hat{G}_c;Y),\ \hat{G}_s = G - g(G).\end{aligned} \tag{37}$$

where $\hat{G}_c = g(G), \tilde{G}_c = g(\tilde{G})$ and $\tilde{G} \sim P(G|Y)$, i.e., $\tilde{G}$ and G have the same label. In Sec. B.4, we introduce a contrastive approximation for $I(\hat{G}_c;\tilde{G}_c|Y)$:

$$I(\hat{G}_c;\tilde{G}_c|Y) \approx \mathbb{E}_{\substack{\{\hat{G}_c,\tilde{G}_c\}\sim\mathbb{P}_g(G|\mathcal{Y}=Y)\\ \{G_c^i\}_{i=1}^M\sim\mathbb{P}_g(G|\mathcal{Y}\neq Y)}} \log \frac{e^{\phi(h_{\hat{G}_c},h_{\tilde{G}_c})}}{e^{\phi(h_{\hat{G}_c},h_{\tilde{G}_c})} + \sum_i^M e^{\phi(h_{\hat{G}_c}h_{G_c^i})}}, \tag{38}$$

where positive samples $(\hat{G}_c, \tilde{G}_c)$ are the extracted subgraphs of graphs that have the same label of G, negative samples are those with different labels, $\mathbb{P}_g(G|\mathcal{Y} = Y)$ is the pushforward distribution of $\mathbb{P}(G|\mathcal{Y} = Y)$ by featurizer g, $\mathbb{P}(G|\mathcal{Y} = Y)$ refers to the distribution of G given the label Y, $h_{\hat{G}_c}, h_{\tilde{G}_c}, h_{G_c^i}$ are the graph presentations of the estimated subgraphs, and ϕ is the similarity metric for the graph presentations. As $M \to \infty$, Eq. 38 approximates $I(\hat{G}_c;\tilde{G}_c|Y)$ which can be regarded as a non-parameteric resubstitution entropy estimator via the von Mises-Fisher kernel density (Ahmad & Lin, 1976; Kandasamy et al., 2015; Wang & Isola, 2020).

While for the third term $I(\hat{G}_s;Y)$ and the constraint $I(\hat{G}_s;Y) \le I(\hat{G}_c;Y)$, a straightforward implementation is to imitate the hinge loss:

$$I(\hat{G}_s;Y) \approx \frac{1}{N} R_{\hat{G}_s} \cdot \mathbb{I}(R_{\hat{G}_s} \le R_{\hat{G}_c}), \tag{39}$$

where N is the number of samples, $\mathbb{I}$ is a indicator function that outputs 1 when the interior condition is satisfied otherwise 0, and $R_{\hat{G}_s}$ and $R_{\hat{G}_c}$ are the empirical risk vector of the predictions for each sample based on $\hat{G}_s$ and $\hat{G}_c$ respectively. One can also formulate Eq. 37 from game-theoretic perspective (Chang et al., 2020).

Finally, we can derive the specific loss for the optimization of GOODv2 combining Eq. 38 and Eq. 39:

$$\begin{aligned}&R_{\hat{G}_c} + \alpha\mathbb{E}_{\substack{\{\hat{G}_c,\tilde{G}_c\}\sim\mathbb{P}_g(G|\mathcal{Y}=Y)\\ \{G_c^i\}_{i=1}^M\sim\mathbb{P}_g(G|\mathcal{Y}\neq Y)}} \log \frac{e^{\phi(h_{\hat{G}_c},h_{\tilde{G}_c})}}{e^{\phi(h_{\hat{G}_c},h_{\tilde{G}_c})} + \sum_i^M e^{\phi(h_{\hat{G}_c}h_{G_c^i})}}\\ &+ \beta\frac{1}{N} R_{\hat{G}_s} \cdot \mathbb{I}(R_{\hat{G}_s} \le R_{\hat{G}_c}),\end{aligned} \tag{40}$$

where $R_{\hat{G}_c}, R_{\hat{G}_s}$ are the empirical risk when using $\hat{G}_c, \hat{G}_s$ to predict Y through the classifier. Typically, we use a additional MLP downstream classifier ρ_s for $\hat{G}_s$ in the classifier GNN. $h_{\hat{G}_c}$ is the graph representation of $\hat{G}_c$ which can be induced from the GNN encoder either in the featurizer or in the classifier. α, β are the weights for $I(\hat{G}_c;\tilde{G}_c|Y)$ and $I(\hat{G}_s;Y)$, and ϕ is implemented as cosine similarity. The optimization loss for GOODv1 merely contains the first two terms in Eq. 40.

The detailed algorithm for GOOD is given in the Algorithm 1, assuming the $h_{\hat{G}_c}$ is obtained via the graph encoder in f_c.

Algorithm 1 Pseudo code for GOOD framework.

```
Input: Training graphs and labels $\mathcal{D}_{\text{tr}} = \{G_i, Y_i\}_{i=1}^N$; learning rate $l$; loss weights $\alpha, \beta$ required by Eq. 40; training epochs $e$; batch size $b$;
Randomly initialize parameters of $g, f_c$;
for $i = 1$ to $e$ do
    Sample a batch of graphs $\{G^j, Y^j\}_{j=1}^b$;
    Estimate the invariant subgraph for the batch: $\{\hat{G}_c^j\}_{j=1}^b = g(\{G^j, y^j\}_{j=1}^b)$;
    Make predictions based the estimated invariant subgraph: $\{\hat{Y}^j\}_{j=1}^b = f_c(\{\hat{G}_c^j\}_{j=1}^b)$;
    Calculate the empirical loss $R_{\hat{G}_c}$ with $\{\hat{y^j}\}_{j=1}^b$;
    Fetch the graph representations of invariant subgraphs from $f_c$ as $\{h_{\hat{G}_c^j}\}_{j=1}^b$;
    Calculate the contrastive loss $R_c$ with Eq. 38, where positive samples and negative samples are constructed from the batch;
    Obtain $\hat{G}_s$ for the batch: $\{\hat{G}_s^j\}_{j=1}^b = \{G^j - \hat{G}_c^j\}_{j=1}^b$;
    Make predictions based the $\hat{G}_s$: $\{\hat{Y}_s^j\}_{j=1}^b = f_c(\{\hat{G}_s^j\}_{j=1}^b)$;
    Calculate the empirical loss $R_{\hat{G}_s}$ with $\{\hat{Y}_s^j\}_{j=1}^b$, and weighted as Eq. 39;
    Update parameters of $g, f_c$ with respect to $R_{\hat{G}_c} + \alpha R_c + \beta R_{\hat{G}_s}$ as Eq. 40;
end for
```

G. Full Experimental Results and Settings

In this section, we provide more details about the experimental results and settings in complementary to Sec. 4, including the full empirical results and analysis, dataset preparation, dataset statistics, implementations of baselines, selection of models and hyperparameters as well as evaluation protocols.

G.1. Full experimental results and analysis

Datasets. We use the SPMotif datasets from DIR (Wu et al., 2022c) where artificial structural shifts and graph size shifts are nested (SPMotif-Struc). Besides, we construct a harder version mixed with attribute shifts (SPMotif-Mixed). To examine GOOD in real-world scenarios with more complicated relationships and distribution shifts, we also use DrugOOD (Ji et al., 2022) from AI-aided Drug Discovery with Assay, Scaffold, and Size splits, convert the ColoredMNIST from IRM (Arjovsky et al., 2019) using the algorithm from Knyazev et al. (2019) to inject attribute shifts, and split Graph-SST (Yuan et al., 2020) to inject degree biases. To compare with previous specialized OOD methods for graph size shifts (Yehudai et al., 2021; Bevilacqua et al., 2021), we use the datasets in Bevilacqua et al. (2021) that are converted from TU benchmarks (Morris et al., 2020).

Baselines and our methods. Besides the ERM, we also compare with SOTA interpretable GNNs, GIB (Yu et al., 2021), ASAP Pooling (Ranjan et al., 2020), and DIR (Wu et al., 2022c), to validate the effectiveness of the optimization objective in GOOD. We use the same selection ratio (i.e., s_c) for all models. On the other hand, to validate the effectiveness of the decomposition in GOOD, we compare GOOD with SOTA OOD objectives including IRM (Arjovsky et al., 2019), v-Rex (Krueger et al., 2021) and IB-IRM (Ahuja et al., 2021), for which we apply random environment partitions. We also compare GOOD with EIIL (Creager et al., 2021) and CNC (Zhang et al., 2022) that does not require environment labels, where CNC (Zhang et al., 2022) has a more sophisticated contrastive sampling strategy for combating subpopulation shifts.

OOD performance on structure and mixed shifts. In Table 3, we report the test accuracy of each method, where we omit GIB due to its poor convergence. Different biases indicate different strengths of the distribution shifts. Although the training accuracy of most methods converge to more than 99%, the test accuracy decreases dramatically as the bias increases and as more distribution shifts are mixed, which concurs with our discussions in Sec. 2.2 and Appendix D. Due to the simplicity of the task as well as the relatively high support overlap between training and test distributions, interpretable GNNs and OOD objectives can improve certain OOD performance, while they can have *high variance* since they donot have OOD generalization guarantees. In contrast, GOODv1 and GOODv2 outperform all of the baselines by a significant margin up to 10% with *lower variance*, which demonstrates the effectiveness and excellent OOD generalization ability of GOOD. More analysis and results on real-world datasets are given in Appendix G.

Table 3. OOD generalization performance on structure and mixed shifts for synthetic graphs.

	SPMotif-Struc†			SPMotif-Mixed†			
	bias=0.33	bias=0.60	bias=0.90	bias=0.33	bias=0.60	bias=0.90	Avg
ERM	59.49 (3.50)	55.48 (4.84)	49.64 (4.63)	58.18 (4.30)	49.29 (8.17)	41.36 (3.29)	52.24
ASAP	64.87 (13.8)	64.85 (10.6)	**57.29 (14.5)**	66.88 (15.0)	59.78 (6.78)	50.45 (4.90)	60.69
DIR	58.73 (11.9)	48.72 (14.8)	41.90 (9.39)	67.28 (4.06)	51.66 (14.1)	38.58 (5.88)	51.14
IRM	57.15 (3.98)	61.74 (1.32)	45.68 (4.88)	58.20 (1.97)	49.29 (3.67)	40.73 (1.93)	52.13
V-Rex	54.64 (3.05)	53.60 (3.74)	48.86 (9.69)	57.82 (5.93)	48.25 (2.79)	43.27 (1.32)	51.07
EIIL	56.48 (2.56)	60.07 (4.47)	55.79 (6.54)	53.91 (3.15)	48.41 (5.53)	41.75 (4.97)	52.73
IB-IRM	58.30 (6.37)	54.37 (7.35)	45.14 (4.07)	57.70 (2.11)	50.83 (1.51)	40.27 (3.68)	51.10
CNC	70.44 (2.55)	**66.79 (9.42)**	50.25 (10.7)	65.75 (4.35)	59.27 (5.29)	41.58 (1.90)	59.01
GOODv1	**71.07 (3.60)**	63.23 (9.61)	51.78 (7.29)	**74.35 (1.85)**	**64.54 (8.19)**	49.01 (9.92)	**62.33**
GOODv2	**77.33 (9.13)**	**69.29 (3.06)**	**63.41 (7.38)**	**72.42 (4.80)**	**70.83 (7.54)**	**54.25 (5.38)**	**67.92**

†Higher accuracy and lower variance indicate better OOD generalization ability.

Table 4. OOD generalization performance on complex distribution shifts for real-world graphs.

Datasets	Drug-Assay	Drug-Sca	Drug-Size	CMNIST-sp	Graph-SST5	Twitter	Avg (Rank)†
ERM	71.79 (0.27)	68.85 (0.62)	66.70 (1.08)	13.96 (5.48)	43.89 (1.73)	60.81 (2.05)	54.33 (6.00)
ASAP	70.51 (1.93)	66.19 (0.94)	64.12 (0.67)	10.23 (0.51)	44.16 (1.36)	60.68 (2.10)	52.65 (8.33)
GIB	63.01 (1.16)	62.01 (1.41)	55.50 (1.42)	15.40 (3.91)	38.64 (4.52)	48.08 (2.27)	47.11 (10.0)
DIR	68.25 (1.40)	63.91 (1.36)	60.40 (1.42)	15.50 (8.65)	41.12 (1.96)	59.85 (2.98)	51.51 (9.33)
IRM	72.12 (0.49)	68.69 (0.65)	66.54 (0.42)	31.58 (9.52)	43.69 (1.26)	63.50 (1.23)	57.69 (4.50)
V-Rex	72.05 (1.25)	68.92 (0.98)	66.33 (0.74)	10.29 (0.46)	43.28 (0.52)	63.21 (1.57)	54.01 (6.17)
EIIL	72.60 (0.47)	68.45 (0.53)	66.38 (0.66)	30.04 (10.9)	42.98 (1.03)	62.76 (1.72)	57.20 (5.33)
IB-IRM	72.50 (0.49)	68.50 (0.40)	66.64 (0.28)	**39.86 (10.5)**	40.85 (2.08)	61.26 (1.20)	58.27 (5.33)
CNC	72.40 (0.46)	67.24 (0.90)	65.79 (0.80)	12.21 (3.85)	42.78 (1.53)	61.03 (2.49)	53.56 (7.50)
GOODv1	**72.71 (0.52)**	**69.04 (0.86)**	**67.24 (0.88)**	19.77 (17.1)	**44.71 (1.14)**	**63.66 (0.84)**	**56.19 (2.50)**
GOODv2	**73.17 (0.39)**	**69.70 (0.27)**	**67.78 (0.76)**	**44.91 (4.31)**	**45.25 (1.27)**	**64.45 (1.99)**	**60.88 (1.00)**

†Averaged rank is also reported in the blankets because of dataset heterogeneity. Lower rank is better.

OOD generalization performance on realistic shifts. In Table 4 and Table 5, we examine the effectiveness of GOOD in real-world data and more complicated distribution shifts. Both averaged accuracy and ranks are reported because of the dataset heterogeneity. Since the tasks are harder than synthetic ones, interpretable GNNs and OOD objectives perform similar to or even under-perform the ERM baselines, which is also consistent to the observations in non-linear benchmarks (Gulrajani & Lopez-Paz, 2021; Ji et al., 2022). However, both GOODv1 and GOODv2 consistently and significantly outperform previous methods, including previous specialized methods Γ GNNs (Bevilacqua et al., 2021) for combating graph size shifts, demonstrating the generality and superiority of GOOD.

OOD generalization performance on graph size shifts. In Table 5, we additionally compare GOOD to the specialized designed methods for OOD generalization in terms of graph sizes (Γ GNNs) (Bevilacqua et al., 2021), where we include the author reported results for both kernel methods and Γ GNNs. It can be found that GOOD consistently and significantly outperforms the previous SOTA methods, which further demonstrates the generality of GOOD.

Comparisons with advanced ablation variants. As discussed in Sec. B.4, GOOD can be reduced to interpretable GNNs and contrastive learning approaches. However, across all experiments, we can observe that neither the advanced interpretable GNNs (DIR) nor sophisticated contrastive objectives with specialized sampling strategy (CNC) can yield satisfactory OOD performance, which serves as *strong evidence* for the necessities of the decomposition as well as the objective in GOOD. Furthermore, although GOODv1 can outperform GOODv2 when we may have a relatively accurate s_c, the improvements in GOODv1 are not as stable as GOODv2 or even unsatisfactory when the assumption is violated. This phenomenon also reveals the advances of GOODv2 in practical scenarios.

Table 5. OOD generalization performance on graph size shifts for real-world graphs in terms of Matthews correlation coefficient.

Datasets	NCI1	NCI109	PROTEINS	DD	Avg
ERM	0.15 (0.05)	0.16 (0.02)	0.22 (0.09)	0.27 (0.09)	0.20
ASAP	0.16 (0.10)	0.15 (0.07)	0.22 (0.16)	0.21 (0.08)	0.19
GIB	0.13 (0.10)	0.16 (0.02)	0.19 (0.08)	0.01 (0.18)	0.12
DIR	0.21 (0.06)	0.13 (0.05)	0.25 (0.14)	0.20 (0.10)	0.20
IRM	0.17 (0.02)	0.14 (0.01)	0.21 (0.09)	0.22 (0.08)	0.19
V-Rex	0.15 (0.04)	0.15 (0.04)	0.22 (0.06)	0.21 (0.07)	0.18
EIIL	0.14 (0.03)	0.16 (0.02)	0.20 (0.05)	0.23 (0.10)	0.19
IB-IRM	0.12 (0.04)	0.15 (0.06)	0.21 (0.06)	0.15 (0.13)	0.16
CNC	0.16 (0.04)	0.16 (0.04)	0.19 (0.08)	0.27 (0.13)	0.20
WL kernel	**0.39 (0.00)**	0.21 (0.00)	0.00 (0.00)	0.00 (0.00)	0.15
GC kernel	0.02 (0.00)	0.00 (0.00)	0.29 (0.00)	0.00 (0.00)	0.08
$\Gamma_{\text{1-hot}}$	0.17 (0.08)	0.25 (0.06)	0.12 (0.09)	0.23 (0.08)	0.19
Γ_{GIN}	0.24 (0.04)	0.18 (0.04)	0.29 (0.11)	**0.28 (0.06)**	0.25
Γ_{RPGIN}	0.26 (0.05)	0.20 (0.04)	0.25 (0.12)	0.20 (0.05)	0.23
GOODv1	0.22 (0.07)	**0.23 (0.09)**	**0.40 (0.06)**	**0.29 (0.08)**	**0.29**
GOODv2	**0.27 (0.07)**	**0.22 (0.05)**	**0.31 (0.12)**	0.26 (0.08)	**0.27**

G.2. Details about the datasets

We provide more details about the motivation and construction method of the datasets that are used in our experiments. Statistics of the datasets are presented in Table 6.

SPMotif datasets. We construct 3-class synthetic datasets based on BAMotif (Ying et al., 2019; Luo et al., 2020) following (Wu et al., 2022c), where the model needs to tell which one of three motifs (House, Cycle, Crane) that the graph contains. For each dataset, we generate 3000 graphs for each class at the training set, 1000 graphs for each class at the validation set and testing set, respectively. During the construction, we merely inject the distribution shifts in the training data while keep the testing data and validation data without the biases. For structure-level shifts (**SPMotif-Struc**), we introduce the bias based on FIIF, where the motif and one of the three base graphs (Tree, Ladder, Wheel) are artificially (spuriously) correlated with a probability of various biases, and equally correlated with the other two. Specifically, given a predefined bias b, the probability of a specific motif (e.g., House) and a specific base graph (Tree) will co-occur is b while for the others is $(1-b)/2$ (e.g., House-Ladder, House-Wheel). We use random node features for SPMotif-Struc, in order to study the influences of structure level shifts. Moreover, to simulate more realistic scenarios where both structure level and topology level have distribution shifts, we also construct **SPMotif-Mixed** for mixed distribution shifts. We additionally introduced FIIF attribute-level shifts based on SPMotif-Struc, where all of the node features are spuriously correlated with a probability of various biases by setting to the same number of corresponding labels. Specifically, given a predefined bias b, the probability that all of the node features of a graph has label y (e.g., $y=0$) being set to y (e.g., $\boldsymbol{X}=\boldsymbol{0}$) is b while for the others is $(1-b)/2$ (e.g., $P(\boldsymbol{X}=\boldsymbol{1})=P(\boldsymbol{X}=\boldsymbol{2})=(1-b)/2$). More complex distribution shift mixes can be studied following our construction approach, which we will leave for future works.

TU datasets. To study the effects of graph sizes shifts, we follow Yehudai et al. (2021); Bevilacqua et al. (2021) to study the OOD generalization abilities of various methods on four of TU datasets (Morris et al., 2020), i.e., **PROTEINS, DD, NCI1, NCI109**. Specifically, we use the data splits generated by Yehudai et al. (2021) and use the Matthews correlation coefficient as evaluation metric following (Bevilacqua et al., 2021) due to the class imbalance in the splits. The splits are generated as follows: Graphs with sizes smaller than the 50-th percentile are assigned to training, while graphs with sizes larger than the 90-th percentile are assigned to test. A validation set for hyperparameters tuning consists of 10% held out examples from training. We also provide a detailed statistics about these datasets in table 7.

Graph-SST datasets. Inspired by the data splits generation for studying distribution shifts on graph sizes, we split the data curated from sentiment graph data (Yuan et al., 2020), that converts sentiment sentence classification datasets **SST5** and **SST-Twitter** (Socher et al., 2013; Dong et al., 2014) into graphs, where node features are generated using BERT (Devlin et al., 2019) and the edges are parsed by a Biaffine parser (Gardner et al., 2018). Our splits are created according to the averaged degrees of each graph. Specifically, we assign the graphs as follows: Those that have smaller or equal than 50-th percentile averaged degree are assigned into training, those that have averaged degree large than 50-th percentile while smaller than 80-th percentile are assigned to validation set, and the left are assigned to test set. For SST5 we follow the above process while for Twitter we conduct the above split in an inversed order to study the OOD generalization ability of GNNs trained on large degree graphs to small degree graphs.

CMNIST-sp. To study the effects of PIIF shifts, we select the ColoredMnist dataset created in IRM (Arjovsky et al., 2019). We convert the ColoredMnist into graphs using super pixel algorithm introduced by Knyazev et al. (2019). Specifically, the original Mnist dataset are assigned to binary labels where images with digits $0-4$ are assigned to $y=0$ and those with digits $5-9$ are assigned to $y=1$. Then, y will be flipped with a probability of 0.25. Thirdly, green and red colors will be respectively assigned to images with labels 0 and 1 an averaged probability of 0.15 (since we do not have environment splits) for the training data. While for the validation and testing data the probability is flipped to 0.9.

DrugOOD datasets. To evaluate the OOD performance in realistic scenarios with realistic distribution shifts, we also include three datasets from DrugOOD benchmark. DrugOOD is a systematic OOD benchmark for AI-aided drug discovery, focusing on the task of drug target binding affinity prediction for both macromolecule (protein target) and small-molecule (drug compound). The molecule data and the notations are curated from realistic ChEMBL database (Mendez et al., 2019). Complicated distribution shifts can happen on different assays, scaffolds and molecule sizes. In particular, we select `DrugOOD-lbap-core-ic50-assay`, `DrugOOD-lbap-core-ic50-scaffold`, and `DrugOOD-lbap-core-ic50-size`, from the task of Ligand Based Affinity Prediction which uses `ic50` measurement type and contains `core` level annotation noises. For more details, we refer interested readers to Ji et al. (2022).

Table 6. Information about the datasets used in experiments. The number of nodes and edges are taking average among all graphs. MCC indicates the Matthews correlation coefficient.

DATASETS	# TRAINING	# VALIDATION	# TESTING	# CLASSES	# NODES	# EDGES	METRICS
SPMOTIF	9,000	3,000	3,000	3	44.96	65.67	ACC
PROTEINS	511	56	112	2	39.06	145.63	MCC
DD	533	59	118	2	284.32	1,431.32	MCC
NCI1	1,942	215	412	2	29.87	64.6	MCC
NCI109	1,872	207	421	2	29.68	64.26	MCC
SST5	6,090	1,186	2,240	5	19.85	37.70	ACC
TWITTER	3,238	694	1,509	3	21.10	40.20	ACC
CMNIST-SP	40,000	5,000	15,000	2	56.90	373.85	ACC
DRUGOOD-ASSAY	34,179	19,028	19,032	2	32.27	70.25	ROC-AUC
DRUGOOD-SCAFFOLD	21,519	19,041	19,048	2	29.95	64.86	ROC-AUC
DRUGOOD-SIZE	36,597	17,660	16,415	2	30.73	66.90	ROC-AUC

Table 7. Detailed statistics of selected TU datasets. Table from Yehudai et al. (2021); Bevilacqua et al. (2021).

	NCI1			NCI109		
	ALL	SMALLEST 50%	LARGEST 10%	ALL	SMALLEST 50%	LARGEST 10%
CLASS A	49.95%	62.30%	19.17%	49.62%	62.04%	21.37%
CLASS B	50.04%	37.69%	80.82%	50.37%	37.95%	78.62%
NUM OF GRAPHS	4110	2157	412	4127	2079	421
AVG GRAPH SIZE	29	20	61	29	20	61

	PROTEINS			DD		
	ALL	SMALLEST 50%	LARGEST 10%	ALL	SMALLEST 50%	LARGEST 10%
CLASS A	59.56%	41.97%	90.17%	58.65%	35.47%	79.66%
CLASS B	40.43%	58.02%	9.82%	41.34%	64.52%	20.33%
NUM OF GRAPHS	1113	567	112	1178	592	118
AVG GRAPH SIZE	39	15	138	284	144	746

G.3. Training and Optimization in Experiments

During the experiments, we do not tune the hyperparameters exhaustively while following the common recipes for optimizing GNNs. Details are as follows.

GNN encoder. For fair comparison, we use the same GNN architecture as graph encoders for all methods. By default, we use 3-layer GNN with Batch Normalization (Ioffe & Szegedy, 2015) between layers and JK residual connections at last layer (Xu et al., 2018). For the architectures we use the GCN with mean readout (Kipf & Welling, 2017) for all datasets except Proteins where we empirically observe better validation performance with a GIN and max readout (Xu et al., 2019), and for DrugOOD datasets where we follow the backbone used in the paper (Ji et al., 2022), i.e., 4-layer GIN with sum readout. The hidden dimensions are fixed as 32 for SPMotif, TU datasets, CMNIST-sp, and 128 for SST5, Twitter and DrugOOD datasets.

Optimization and model selection. By default, we use Adam optimizer (Kingma & Ba, 2015) with a learning rate of $1e-3$ and a batch size of 32 for all models at all datasets. Except for DrugOOD datasets, we use a batch size of 128 following the original paper (Ji et al., 2022). To avoid underfitting, we pretrain models for 20 epochs for all datasets, except for CMNIST and Twitter where we pretrain 5 epochs and for SST5 we pretrain 10 epochs, because of the dataset size and the difficulty of the task. To avoid overfitting, we also employ an early stopping of 5 epochs according to the validation performance. Meanwhile, dropout (Srivastava et al., 2014) is also adopted for some datasets. Specifically, we use a dropout rate of 0.5 for CMNIST, SST5, Twitter, DrugOOD-Assay and DurgOOD-Scaffold, 0.1 for DrugOOD-Size according to the validation performance, and 0.3 for TU datasets following the practice of Bevilacqua et al. (2021).

Implementations of baselines. For implementations of the interpretable GNNs, we use the author released codes (Yu et al., 2021; Ranjan et al., 2020), where we use the codes provided by the authors[3] for DIR c(Wu et al., 2022c) which is the same as the author released codes. During the implementation, we use the same s_c for all interpretable GNN baselines, chosen from $\{0.1, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ according to the validation performances, and set to 0.25 for SPMotif following Wu et al. (2022c), 0.3 for Proteins and DD, 0.6 for NCI1, 0.7 for NCI109, 0.8 for CMNIST-sp, 0.5 for SST5 and Twitter, and 0.8 for DrugOOD datasets, respectively. Empirically, we observe that the optimization process in GIB can be unstable during its nested optimization for approximating the mutual information of the predicted subgraph and the input graph. We use a larger batch size of 128 or reduce the nested optimization steps to be lower than 20 for stabilizing the performance. If the optimization failed due to the instability during training, we will select the results

[3] https://anonymous.4open.science/r/DIR/

with best validation accuracy as the final outcomes. Although SPMotif-Struc is also evaluated in DIR, we find the results are inconsistent to the results reported by the author, because DIR adopts `Last Epoch Model Selection` which is *different* from the claim that they select models according to `the validation performance`, i.e., `line 264` to `line 278` in `train/spmotif_dir.py` from the commit `4b975f9b3962e7820d8449eb4abbb4cc30c1025d` of `https://github.com/Wuyxin/DIR-GNN`. We select the hyperparamter for the proposed DIR regularization from $\{0.01, 0.1, 1, 10\}$ according to the validation performances at the datasets, while we stick to the authors claimed hyperparameters for the datasets they also experimented with.

For invariant learning, we refer to the implementations in DomainBed (Gulrajani & Lopez-Paz, 2021) for IRM (Arjovsky et al., 2019), V-Rex (Krueger et al., 2021) and IB-IRM (Ahuja et al., 2021). Since the environment information is not available, we perform random partitions on the training data to obtain two equally large environments for these objectives. Moreover, we select the weights for the corresponding regularization from $\{0.01, 0.1, 1, 10, 100\}$ for these objectives according to the validation performances of IRM and stick to it for others, since we empirically observe that they perform similarly with respect to the regularization weight choice. For EIIL (Creager et al., 2021), we use the author released implementations about assigning different samples the weights for being put in each environment and calculating the IRM loss.

Besides, for CNC (Zhang et al., 2022), we follow the algorithm description to modify the sampling strategy in supervised contrastive loss (Khosla et al., 2020) based on a pretrained GNN optimized with ERM, and choose the weight for contrastive loss using the same grid search as for GOOD.

Implementations of GOOD. For fair comparison, GOOD uses the same GNN architecture for GNN encoders as the baseline methods. We did not do exhaustive hyperparameters tuning for the loss Eq. 40. By default, we fix the temperature to be 1 in the contrastive loss, and merely search α from $\{0.5, 1, 2, 4, 8, 16, 32\}$ and β from $\{0.5, 1, 2, 4\}$ according to the validation performances. For CMNIST-sp, we find larger β are required to get rid of intense spurious node features hence we expand the search range for β to $\{0.5, 1, 2, 4, 16, 32\}$, For Graph-SST datasets, we search α from $\{0.5, 1, 2, 4\}$ as we empirically find that increasing α does not help increase the performance with few random runs. Besides, we also have various implementation options for obtaining the features in $\hat{G}_c$, for obtaining $h_{\hat{G}_c}$, as well as for obtaining predictions based on $\hat{G}_s$. By default, we feed the graph representations of featurizer GNN to the classifier GNN, as well as to the contrastive loss. For classifying G based on $\hat{G}_s$, we use a separate MLP downstream classifier in the classifier GNN f_c. The only exception is for the CMNIST-sp dataset where the spurious correlation is stronger than the invariant signal. Directly feeding the graph representations from the featurizer GNN can easily overfit to the shortcuts hence we instead feed the original features to the downstream classifier GNN. There can be more other options, such as using separate graph convolutions on $\hat{G}_s$ or $\hat{G}_c$, which we leave for future work. Options for obtaining the features in $\hat{G}_c$ are: $\{$from g, from the raw features$\}$. Options for obtaining $h_{\hat{G}_c}$ are: $\{$from the GNN encoder of the classifier f_c with the same pooling as the classifier, from the GNN encoder of the featurizer g with a SUM global pooling, $\}$. Options for obtaining predictions based on $\hat{G}_s$ are:$\{$from another classifier with shared GNN encoder of f_c, from another classifier with shared GNN encoder of f_c while without gradients backwards to the encoder, from a single GNN convolution and a same pooling as $f_c\}$. We select the corresponding options according to the validation performance with several runs of random α and β, and stick to one for each dataset. As a result, we empirically find using the raw node features for $\hat{G}_c$, obtaining $h_{\hat{G}_c}$ via a global ADD pooling with the featurizer outputs, and obtaining predictions based on $\hat{G}_s$ from another classifier with shared GNN encoder of f_c while without gradients backwards to the encoder, has better validation performances. Except for TU datasets where we use the outputs of g as the features of $\hat{G}_c$, and obtain predictions with one GNN layer for the prediction of $\hat{G}_s$ empirically has better validation performances.

Evaluation protocol. We run each experiment 10 on TU datasets and 5 times for others where the random seeds start from 1 to the number of total repeated times. During each run, we select the model according to the validation performance and report the mean and standard deviation of the corresponding metrics.

G.4. Software and hardware

We implement our methods with PyTorch (Paszke et al., 2019) and PyTorch Geometric (Fey & Lenssen, 2019). We ran our experiments on Linux Servers with 40 cores Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz, 256 GB Memory, and Ubuntu 18.04 LTS installed. GPU environments are varied from 4 NVIDIA RTX 2080Ti graphics cards with CUDA 10.2, 2 NVIDIA RTX 2080Ti and 2 NVIDIA RTX 3090Ti graphics cards with CUDA 11.3, and NVIDIA TITAN series with CUDA 11.3.