A. Appendix

A.1. Proofs

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

Proof. Our proof starts by calculating the risks of the invariant predictor and a perturbed predictor that leverages the information introduced in the augmentation. Following prior works [1], the risk R_{inv} of the invariant predictor $W = W_{inv}$ can be expressed as:

$$\begin{aligned} R_{inv} &= \mathrm{E}[Y^e_{Aug} \oplus I(W \cdot G^e)] \\ &= \mathrm{E}[Y^e_{Aug} \oplus I(W_{inv} \cdot G^e_{inv})] \end{aligned} \tag{1}$$

The prediction relationship between G^e_{inv} and Y is stable across different environments. More importantly, G^e_{inv} is the causal subgraph that intrinsically affects the label $Y(G^e)$ of the training graph G^e. Without the loss of generality, we consider a two-class classification problem with balanced labels, *i.e.* $Y(G^e) \in \{0, 1\}$ and $p(Y(G^e) = 1) = p(Y(G^e) = 0) = 0.5$. When $Y(G^e) = 1$, R_{inv} takes the following form:

$$\begin{aligned} R_{inv} &= (1 - p_{aug}) \cdot p + p_{aug} \cdot (1 - p) \\ &= p + p_{aug} - 2pp_{aug}. \end{aligned} \tag{2}$$

When $Y(G^e) = 1$, R_{inv} is as follows:

$$\begin{aligned} R_{inv} &= (1 - p_{aug}) \cdot p + p_{aug} \cdot (1 - p) \\ &= p + p_{aug} - 2pp_{aug}. \end{aligned} \tag{3}$$

Notice that $p(Y(G^e) = 1) = p(Y(G^e) = 0) = 0.5$. Thus, we finally obtain $R_{inv} = p + p_{aug} - 2pp_{aug}$. We proceed to compute the risk R_{aug} of a perturbed predictor $W = W_{aug}$. Similarly, when $Y(G^e) = 1$, we have:

$$\begin{aligned} R_{aug} &= (1 - p) \cdot (1 - p_{aug}) + pp_{aug} \\ &= 1 - p - p_{aug}. \end{aligned} \tag{4}$$

Moreover, when $Y(G^e) = 0$, the corresponding risk is:

$$\begin{aligned} R_{aug} &= p \cdot (1 - p_{aug}) + (1 - p)p_{aug} \\ &= p + p_{aug} - 2pp_{aug} \end{aligned} \tag{5}$$

Hence, the risk of $W = W_{aug}$ is $R_{aug} = 0.5 - pp_{aug}$. We can easily obtain $R_{inv} \geq R_{aug}$ when $p_{aug} \in [\frac{0.5-q}{1-q}, 1]$ and $R_{inv} < R_{aug}$ when $p_{aug} \in [0, \frac{0.5-q}{1-q})$. QED

With the augmented environments, the invariant predictor is supposed to achieve a lower risk than the perturbed predictor and makes it easier for the GNN predictor to leverage an invariant predictive relationship. However, when the label shift occurs in augmentation *i.e.* $p_{aug} \in [\frac{0.5-q}{1-q}, 1]$, the GNN predictor can easily learn the perturbed predictive relationship to achieve lower risk and is hard to generalize to OOD graphs. This perturbed predictive relationship can be introduced during augmentation, as discussed in the above proof, or possibly embedded in the underlying data generation process. Therefore, it is essential to maintain the label-invariant augmentation for graph OOD generalization. Notice that in the above proof we consider the linear case while most GNNs are nonlinear. However, our empirical evidence in Section 5 of the submission shows that the label shift in augmentation could lead to unsatisfactory OOD performance.

A.2. Comparison between LiSA and Related Works.

When sufficient training environments are lacking, it is natural to consider generating new environments via data augmentation. Therefore, some works handle different OOD generalization problems via augmentation for generalization schemes. Specifically, they generate augmented environments with different graph edition policies and learn an invariant GNN on these environments.

EERM [10] is an OOD generalization method designed for the node classification task, which predicts the label of OOD nodes after training with in-distribution data. In node classification, the nodes are usually in the same graph or several large graphs. EERM employs the graph extrapolation method, which adds new edges to the whole training graph to generate augmented environments. To cover as much population as possible, EERM generates augmentations to maximize the loss variance of the GNN classifier with reinforcement learning. Although this variance regularization somehow improves diversity among augmentations, we find it insufficient to promote diversity in practice. As shown in Table 4 in the main submission, the augmented environments generated by EERM are close to each other, while the proposed LiSA can induce more diverse augmentations. Another concern is that variance regularization may also encourage generating augmented graphs with perturbed labels to enlarge the classification loss. Moreover, EERM is hard to deal with the OOD problem on the graph classification task, where the dataset contains many graphs, and the goal is to infer the graph label. Crucially, the graph extrapolation scheme may perturb the semantic information of graphs or even lead to invalid graphs. For example, the molecule graphs become chemically invalid after adding new edges. Differently, LiSA does not harm the graph validity by mining diverse label-invariant subgraphs, and thus handles the OOD graph classification problem.

DIR [11] is an OOD generalization method designed for graph-level tasks. DIR employs a graph generator to generate an invariant subgraph of the input, hoping the predic-

Table 1. Statistics of all the datasets.

Dataset	Task	Distribution Shift	Nodes	Edges	Classes	Metric
MUTAG [7]	Graph	Size	97.9k	202.5k	2	Accuracy
D&D [5]	Graph	Size	334.9k	1.7M	2	Accuracy
Spurious-Motif [12]	Graph	Spurious Correlation	760.3k-765.9k	1.1M	3	Accuracy
MNIST-75sp [5]	Graph	Noise Features	2.3M	18.9M	10	Accuracy
Twitch-Explicit [8]	Node	Cross Domain	1.9k-9.6k	31.3k-153.1k	2	ROC-AUC
Facebook-100 [9]	Node	Cross Domain	0.7k-41.5k	16.7k-1.6M	2	Accuracy
Elliptic [6]	Node	Temporal shift	203.8k	234.3k	2	F1 Score
OGB-Arxiv [4]	Node	Temporal shift	169.3k	1.2M	40	Accuracy

tion between invariant subgraphs and graph labels is stable across different environments. And the GNN classifier only takes the invariant subgraph as input. To encourage the subgraph to be invariant, DIR employs the graph intervention strategy. It first decomposes the training graphs into invariant and complementary subgraph pairs. Then, it permutes the invariant and complementary subgraph pairs to generate augmented graphs for training. The label of the augmented graph is supposed to be consistent with the invariant subgraph. However, exchanging the complementary subgraphs may also change the label [3] as the graph label is sensitive to the graph structure. Differently, LiSA directly discovers label-invariant subgraphs to construct the augmented environments and thus avoid the label shift problem. Moreover, DIR is hard to implement on the node classification task, while LiSA can adapt to both node and graph classification tasks.

SizeShiftReg [2] is a recently proposed augmentation-based graph OOD generalization method. It studies the size shift between training and testing graphs in the graph classification. During training, it randomly drops a portion of graph structures to generate the augmented graphs, which is a coarsen version of the original graph. However, SizeShiftReg is also likely to change the graph label by randomly dropping graph structures and is incapable of handling other shifts other than the size shift. Since the code is still unavailable, we cannot compare the empirical performance between LiSA and SizeShiftReg.

A.3. Experimental Details

A.3.1 Details on Datasets

We provide more detailed statistics of the datasets in Table 1.

A.3.2 Visualization

We visualize the discovered predictable subgraphs on the MUTAG dataset, which is shown in Figure 1. All the found subgraphs are different, but all contribute to the mutagenic effect of molecules. This indicates that LiSA can generate diverse augmented environments with consistent semantics with the source environment.

Algorithm 1 Optimization algorithm for LiSA.

```
Input: Training set $\{(G_i, Y_i)|i = 1, \cdots, N\}$, subgraph generators $\{g_j(\cdot;\phi_j)\}_{j=1}^K$, graph neural network $f(\cdot;\theta)$, inner-step $I$, outer-step $T$, hyperparameters $\alpha, \beta, \eta_1, \eta_2$.
Output: A generalizable GNN $f_\theta^*$
function LiSA
    $\theta \leftarrow \theta^0$;   $\phi_i \leftarrow \phi_j^0, j = 1, \cdots, K$
    for $i = 0 \rightarrow N$ do
        $\theta \leftarrow \theta^0$
        for $t = 0 \rightarrow T$ do
            for $j = 0 \rightarrow K$ do
                $\phi_j^{t+1} \leftarrow \phi_j^t - \eta_1 \nabla_{\phi_j^t} \mathcal{L}_{cls} + \alpha \mathcal{L}_{info} - \beta \mathcal{L}_e$
            end for
        end for
        $\theta^{i+1} \leftarrow \theta^i - \eta_2 \nabla_{\theta^i} \mathcal{L}_{cls} + Var_e(L_{cls})$
    end for
    return $f_\theta^*$
end function
```

A.3.3 Algorithm

We provide the pseudo code for bilevel optimization of LiSA objective in Eqn.10 in Algorithm 1. For simplicity, we denote the weight of subgraph generator g_i as ϕ_i and the weight of GNN f as θ. Notice that we maximize $\mathcal{L}_e(g_i)$ in the inner loop to improve diversity.

$$\begin{aligned} &\min_f \mathcal{L}_{cls}(f, \{g_i^*\}_{i=1}^n) + \mathrm{Var}_e(\mathcal{L}_{cls}(f, g_i^*)), i = 1 \sim n \\ &s.t. g_i^* = \arg\min_{g_i} \mathcal{L}_{cls}(f, g_i) + \alpha \mathcal{L}_{info}(g_i) - \beta \mathcal{L}_e(g_i). \end{aligned} \tag{6}$$

A.3.4 Sensitivity Study of Hyper-parameters

We study the sensitivity of hyper-parameters α, β and K, which are the weights of $\mathcal{L}_{kld}$, $\mathcal{L}_{dist}$ and the number of subgraph generators K. We report the average ROC-AUC on all testing environments of the Twitch-Explicit dataset in Figure 2. The performance of LiSA is stable on a wide range of hyper-parameters. Moreover, we observe a per-

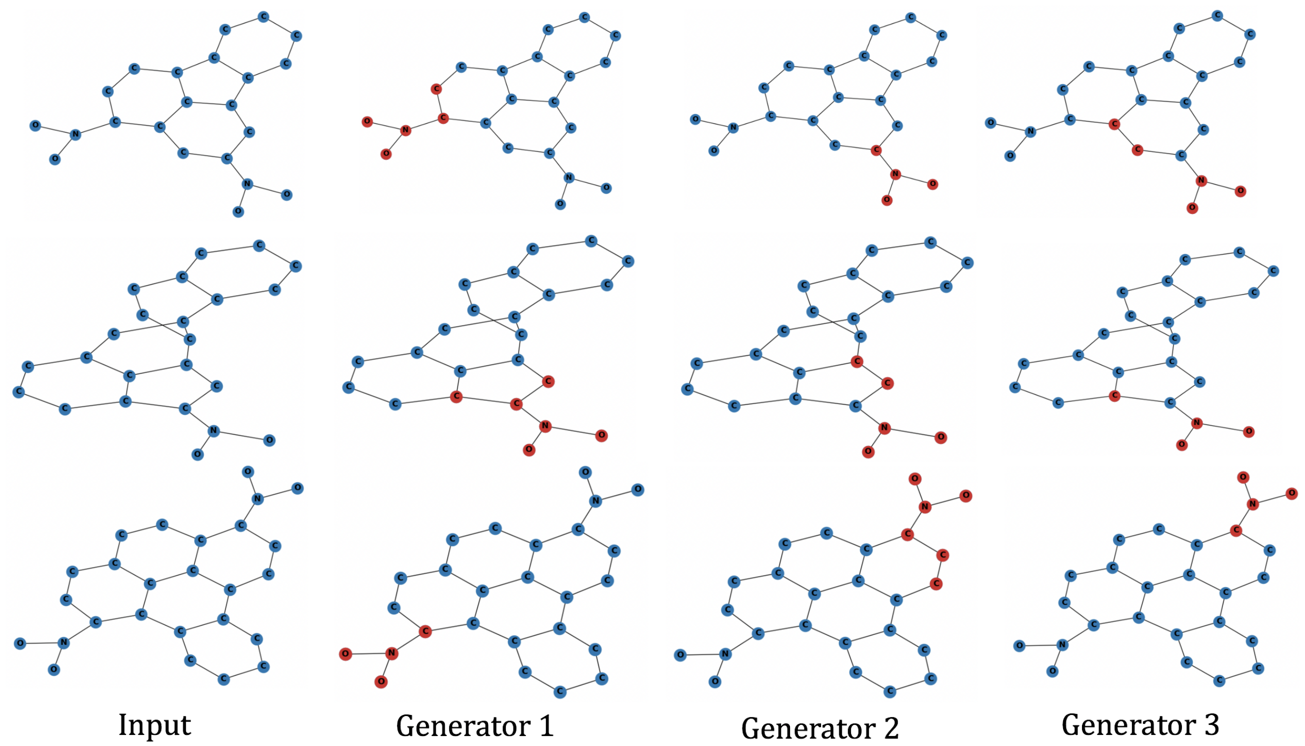

Figure 1. Visualization of the predictable subgraphs on MUTAG dataset. All the subgraph generators found different subgraphs which all contribute to the mutagenic effect. (Best view in color)

formance gain when we increase the number of subgraph generators since it can provide more diverse augmented environments. To extensively study the effect of K, we set $\alpha = 0.1$, $\beta = 0.1$, and vary K from 1 to 7. The results are shown in Table 2. Moreover, we compute the average distance between the augmented environments and source environment, denoted as d, to study the diversity of augmentations. The performance of LiSA increases as K increases from 1 to 5. After that, the performance drops since the diversity (d) of augmented environments decreases. The reason is that different generators may produce similar subgraphs at a large K, leading to the degraded performance of LiSA.

References

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

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

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

[4] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020. 2

[5] Boris Knyazev, Graham W. Taylor, and Mohamed R. Amer. Understanding attention and generalization in graph neural networks. In *NeurIPS*, pages 4204–4214, 2019. 2

[6] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5363–5370, 2020. 2

[7] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. 2

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

[9] Amanda L Traud, Peter J Mucha, and Mason A Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012. 2

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

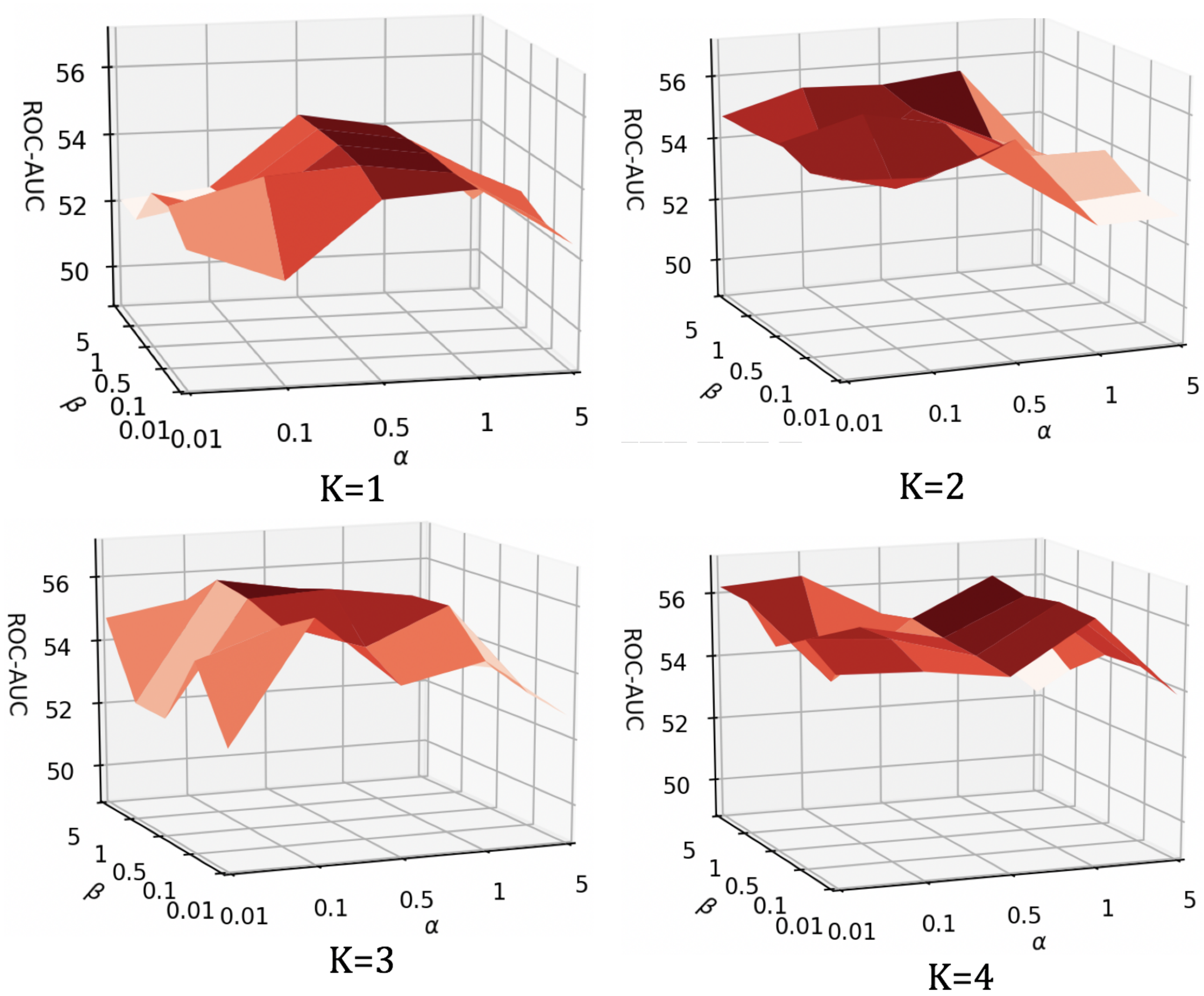

Figure 2. Sensitivity study of hyper-parameters on Twitch-Explicit dataset. We report average ROC-AUC on testing environments.

Table 2. Sensitivity study of the number of subgraph generators.

K	ES	FR	PTBR	RU	TW	d
1	53.72 ± 3.98	53.34 ± 1.64	54.59 ± 6.78	52.34 ± 1.08	52.07 ± 2.83	0.72
2	54.29 ± 1.56	52.07 ± 1.25	55.75 ± 5.29	50.52 ± 2.47	51.59 ± 2.62	0.72
3	57.97 ± 2.96	55.87 ± 2.66	59.96 ± 2.12	52.73 ± 0.67	52.60 ± 2.64	0.67
4	57.70 ± 5.87	54.26 ± 3.13	57.96 ± 7.96	52.45 ± 1.81	52.37 ± 2.93	0.63
5	59.32 ± 1.98	55.57 ± 0.94	59.53 ± 1.53	52.79 ± 0.82	52.80 ± 1.17	0.58
6	56.20 ± 1.98	53.29 ± 1.18	57.10 ± 0.94	52.13 ± 0.68	52.41 ± 2.49	0.60
7	56.77 ± 7.25	53.50 ± 4.74	56.81 ± 9.16	51.65 ± 2.47	50.59 ± 3.59	0.55

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

[12] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in neural information processing systems*, 2019. 2