Graph Structure and Feature Extrapolation for Out-of-Distribution Generalization Supplementary Material

A Background and Related Works

Out-of-Distribution (OOD) Generalization. Out-of-Distribution (OOD) Generalization (Shen et al., 2021; Duchi and Namkoong, 2021; Shen et al., 2020; Liu et al., 2021a) addresses the challenge of adapting a model, trained on one distribution (source), to effectively process unseen data from a potentially different distribution (target). It shares strong ties with various areas such as transfer learning (Weiss et al., 2016; Torrey and Shavlik, 2010; Zhuang et al., 2020), domain adaptation (Wang and Deng, 2018), domain generalization (Wang et al., 2022), causality (Pearl, 2009; Peters et al., 2017), and invariant learning (Arjovsky et al., 2019). As a form of transfer learning, OOD generalization is especially challenging when the target distribution substantially differs from the source distribution. OOD generalization, also known as distribution or dataset shift (Quiñonero-Candela et al., 2008; Moreno-Torres et al., 2012), encapsulates several concepts including covariate shift (Shimodaira, 2000), concept shift (Widmer and Kubat, 1996), and prior shift (Quiñonero-Candela et al., 2008). Both Domain Adaptation (DA) and Domain Generalization (DG) can be viewed as specific instances of OOD, each with its own unique assumptions and challenges.

Domain Generalization (DG). DG (Wang et al., 2022; Li et al., 2017; Muandet et al., 2013; Deshmukh et al., 2019; Gui et al., 2020) strives to predict samples from unseen domains without the need for pre-collected target samples, making it more practical than DA in many circumstances. However, generalizing without additional information is logically implausible, a conclusion also supported by the principles of causality (Pearl, 2009; Peters et al., 2017). As a result, contemporary DG methods have proposed the use of domain partitions (Ganin et al., 2016; Zhang et al., 2022) to generate models that are domain-invariant. Yet, due to the ambiguous definition of domain partitions, many DG methods lack robust theoretical underpinning.

Causality & Invariant Learning. Causality (Peters et al., 2016; Pearl, 2009; Peters et al., 2017) and invariant learning (Arjovsky et al., 2019; Rosenfeld et al., 2020; Ahuja et al., 2021) provide a theoretical foundation for the above concepts, offering a framework to model various distribution shift scenarios as structural causal models (SCMs). SCMs, which bear resemblance to Bayesian networks (Heckerman, 1998), are underpinned by the assumption of independent causal mechanisms, a fundamental premise in causality. Intuitively, this supposition holds that causal correlations in SCMs are stable, independent mechanisms akin to unchanging physical laws, rendering these causal mechanisms generalizable. An assumption of a data-generating SCM equates to the presumption that data samples are generated through these universal mechanisms. Hence, constructing a model with generalization ability requires the model to approximate these invariant causal mechanisms. Given such a model, its performance is ensured when data obeys the underlying data generation assumption. Peters et al. (2016) initially proposed optimal predictors invariant across all environments (or interventions). Motivated by this work, Arjovsky et al. (2019) proposed framing this invariant prediction concept as an optimization process, considering one of the most popular data generation assumptions, PIIF. Consequently, numerous subsequent works (Rosenfeld et al., 2020; Ahuja et al., 2021; Chen et al., 2022a; Lu et al., 2021)—referred to as invariant learning—considered the initial intervention-based environments (Peters et al., 2016) as an environment variable in SCMs. When these environment variables are viewed as domain indicators, it becomes evident that this SCM also provides theoretical support for DG, thereby aligning many invariant works with the DG setting. Besides PIIF, many works have considered FIIF and anti-causal assumptions (Rosenfeld et al., 2020; Ahuja et al., 2021; Chen et al., 2022a), which makes these assumptions as popular basics of causal theoretical analyses.

OOD generalization for graph. Extrapolating on non-Euclidean data has garnered increased attention, leading to a variety of applications (Sanchez-Gonzalez et al., 2018; Barrett et al., 2018; Saxton et al., 2019; Battaglia et al., 2016; Tang et al., 2020; Veličković et al., 2019; Xu et al., 2019b). Inspired by Xu et al. (2020), Yang et al. (2022a) proposed that GNNs intrinsically possess superior generalization capability. Several prior works (Knyazev et al., 2019; Yehudai et al., 2021; Bevilacqua et al., 2021) explored graph generalization in terms of graph sizes, with Bevilacqua

et al. (2021) being the first to study this issue using causal models. Recently, causality modeling-based methods have been proposed for both graph-level tasks (Wu et al., 2022b; Miao et al., 2022; Chen et al., 2022b; Fan et al., 2022; Yang et al., 2022b) and node-level tasks (Wu et al., 2022a). To solve OOD problems in graph, DIR (Wu et al., 2022b) selects graph representations as causal rationales and conducts causal intervention to create multiple distributions. EERM (Wu et al., 2022a) generates environments with REINFORCE algorithm to maximize loss variance between environments while adversarially minimizing the loss. SRGNN (Zhu et al., 2021) aims at pushing biased training data to the given unbiased distribution, performed through central moment discrepancy and kernel matching. To improve interpretation and prediction, GSAT (Miao et al., 2022) learns task-relevant subgraphs by constraining information with stochasticity in attention weights. CIGA (Chen et al., 2022b) models the graph generation process and learns subgraphs to maximally preserve invariant intra-class information. GREA (Liu et al., 2022) performs rationale identification and environment replacement to augment virtual data examples. GIL (Li et al., 2022) proposes to identify invariant subgraphs and infer latent environment labels for variant subgraphs through joint learning. However, except for CIGA (Chen et al., 2022b), their data assumptions are less comprehensive compared to traditional OOD generalization. CIGA, while recognizing the importance of diverse data generation assumptions (SCMs), attempts to fill the gap through non-trivial extra assumptions without environment information. Additionally, environment inference methods have gained traction in graph tasks, including EERM (Wu et al., 2022a), MRL (Yang et al., 2022b), and GIL (Li et al., 2022). However, these methods face two undeniable challenges. First, their environment inference results require environment exploit methods for evaluation, but there are no such methods that perform adequately on graph tasks according to the synthetic dataset results in GOOD benchmark (Gui et al., 2022a). Second, environment inference is essentially a process of injecting human assumptions to generate environment partitions, but these assumptions are not well compared.

Graph data augmentation for generalization. Some data augmentation methods, not limited to graph methods, empirically show improvements in OOD generalization tasks. Mixup (Zhang et al., 2017), which augments samples by interpolating two labeled training samples, is reported to benefit generalization. LISA (Yao et al., 2022) selectively interpolates intra-label or intra-domain samples to further improve OOD robustness. In the graph area, following Mixup, Graph Mixup (Wang et al., 2021) mixes the hidden representations in each GNN layer, while ifMixup (Guo and Mao, 2021) directly applies Mixup on the graph data instead of the latent space. Graph Transplant (Park et al., 2022) employs node saliency information to select a substructure from each graph as units to mix. G-Mixup (Han et al., 2022) interpolates the graph generator of each class and mixes on class-level to improve GNN robustness. DPS (Yu et al., 2022) extracts multiple label-invariant subgraphs with a set of subgraph generators to train an invariant GNN predictor. However, few works target OOD problems, and no prior work generates OOD samples that can provably generalize over graph distribution shifts. In contrast, we offer a graph augmentation method to extrapolate in structure and feature for OOD generalization.

Technical comparisons with prior methods. We discuss in detail the technical differences between existing works and ours. DIR and GREA algorithms are much alike by design, identifying causal subgraphs and switching non-causal subgraphs between graphs. With this localized strategy, their augmented environments can only cover local base shifts, leaving the global structural extrapolation unexplored. EERM exclusively considers node-level tasks, and only performs edge addition/deletion to cover minor shifts on graph base. GDA methods GMixup and Graph Transplant provide no guarantee for solving OOD related tasks, and can not deal with global structure shifts such as size. LiSA (Yu et al., 2023) extracts multiple subgraphs and AdvCA (Sui et al., 2022) masks certain nodes/edges from given graphs to generate graph augmentations. SizeShiftReg (Buffelli et al., 2022) uses coarsening to extracts multiple subgraphs from given graphs, obtaining slightly smaller augmented graphs (80% or 90% of the original graph in actual implementation). These strategies result in augmented graphs that only contain smaller substructures, restricting their potential extrapolation to one instead of both distribution directions. In this case, a common test scenario where test graphs are larger than the training graphs is not covered. Mixup and ExtraMix (Kwon et al., 2022) apply strategies on feature levels without designs for graph structure. In contrast, we study non-Euclidean space extrapolation in a far more systematic way. Our method considers the completeness of achieving both feature and structural extrapolation, and further cover structural global/local extrapolation (or size/base shifts by example) in both distribution directions. This substantially sets the difference between our method and the existing works. Moreover, our novel theoretical contributions include proposing non-Euclidean space linear extrapolation with definitions, analyses,

and guarantees. Considering techniques, our design of graph splice serves global extrapolation and avoids add-on nodes to preserve graph structures, divergent from linker design approaches for molecules (Huang et al., 2022; Igashov et al., 2022). In addition, we design subgraph extraction by label-environment-aware pair learning, a novel technique over previous studies.

B Computational Complexity Analysis

We provide the theoretical analysis regarding the time and space computational complexity as follows. For computation, we generally use one NVIDIA GeForce RTX 2080 Ti for each single experiment.

The time complexity of our G-Splice is $O((|V|^2d + |V|d^2)|B|)$, where $|V|$ denotes the number of nodes, $|B|$ is the batch size, and d is the dimensionality of the representations. The time complexity of our FeatX is $O((|E|d + |V|d^2)|B|)$, where $|E|$ denotes the number of edges. Specifically, message-passing GNNs has a complexity of $O((|E|d + |V|d^2)|B|)$, which we adopt to instantiate our GNN components. For G-Splice, the time complexity of obtaining GNN representations is $O((|E|d + |V|d^2)|B|)$, and that of pair-wise similarity matching and bridge generation are both $O(|V|^2d|B|)$. Since $O(|E|) \leq O(|V|^2)$, the overall time complexity of G-Splice is $O((|E|d + |V|^2d + |V|d^2)|B|)$=$O((|V|^2d + |V|d^2)|B|)$. For FeatX, the time complexity of non-causal feature selection and feature extrapolation are both $O(|V|d_V|B|)$, where d_V is the node feature dimension, which is far smaller than the time complexity of GNN representations. In comparison, the time complexity of most GNN-based graph representation methods are $O(|E|d+|V|d^2)$, including simple algorithms like ERM, IRM, and VRex implemented with GNNs. More complicated graph methods such as CIGA has time complexity $O((|V|^2d + |V|d^2)|B|)$, which is also the case for G-Splice. Therefore, the time complexity of our proposed methods is on par with the existing methods.

The space complexity of G-Splice and FeatX is $O(|V|dL + |E|d)$, where L is the number of GNN layers. This is the space complexity of the message-passing GNNs we use, as well as the space complexity of most GNN-based methods.

C Technical Details

We complete the technical details of causal and environmental subgraph extractions here. Let G_{ε_1} and G_{ε_2} be two graphs with the same label y_1 but different environments ε_1 and ε_2. As we have discussed, G_{inv} should be the subgraph both graphs contain and have most in common. Since graph neural networks aggregate information of an ego graph, i.e., the local subgraph within k-hop of a node, to the embedding of that node through message passing, nodes with similar ego graphs should have similar embeddings. Therefore, in the node embedding space, nodes from G_{ε_1} and G_{ε_2} with similar representations should be a part of G_{inv}. We encode both graphs into node embeddings with a GNN and calculate their weighted similarity matrix $\boldsymbol{S}^{\boldsymbol{w}}$, each element of which is the weighted cosine similarity of a pair of nodes from G_{ε_1} and G_{ε_2}, *i.e.*,

$$\boldsymbol{S}^{\boldsymbol{w}}_{ij} = w * S_c(\boldsymbol{z}_i, \boldsymbol{z}_j), \text{ for } v_i \sim G_{\varepsilon_1}, v_j \sim G_{\varepsilon_2}, \tag{5}$$

where w is a trainable parameter and $S_c(\cdot)$ is the cosine similarity calculation. The scores in the weighted similarity matrix $\boldsymbol{S}^{\boldsymbol{w}}$ are considered as probabilities to sample the causal subgraph from either G_{ε_1} or G_{ε_2}. We use label-invariant and environment-variant graph pairs to pre-train a causal subgraph searching network, which is optimized for the sampled causal subgraphs to be capable of predicting the label Y solely.

Similarly, we perform similarity matching for environment-invariant graph pairs to extract environmental subgraphs G_{env} that are determined by the environment $\mathcal{E}$. Graphs from the same environment should contain similar subgraphs, and we aim at extracting these environmental subgraphs. Environment-invariant and label-variant graph pairs are used to pre-train the environmental subgraph searching network. We calculate the weighted similarity scores from embeddings and sample subgraphs with probabilities. The network is optimized using the environment label ε for the sampled subgraphs to predict the environment.

D FURTHER DISCUSSIONS

D.1 CONNECTIONS BETWEEN SLE AND FLE

The connection between feature extrapolation and structural extrapolation is implicitly described in Sec 2's Graph structure and feature distribution shifts and Sec 3 Linear Extrapolation in Graph Space. Firstly, Linear Extrapolation, which constructs samples beyond the known range while maintaining the same direction and magnitude of known sample differences, is a central concept in our approach. Graph data are complex in that it contains features as well as topological structures. We propose to define Linear Extrapolation for both feature and structure (Sec 3.2), and together they form the complete definition of Graph Linear Extrapolation. Though their definition have different variables, the formulas share the common form of mathematical organization as linear extrapolations. Secondly, graph distribution shifts can happen on both features and structures, which possesses different properties and should be handled separately. While feature extrapolation and structural extrapolation can be used to solve respective shifts, when combined they can address complex feature-structure shifts. Therefore, they complement each other in a systematic solution of graph OOD problems. Thirdly, feature and structural extrapolation share similar logic in causal analyses and theoretical justification. Combined causal analyses are given in Sec 3.1. In Sec 3.3, we provide theoretical guarantees that structural linear extrapolation has the capability to generate OOD samples that are both plausible and diverse, while the justification of feature linear extrapolation is relatively straightforward and provided in Section 5.3.

D.2 APPLICABILITY OF THE CAUSAL ADDITIVITY ASSUMPTION

Our work builds upon the principle of Causal Additivity, a causal assumption widely applicable in graph classification tasks. This assumption can be subjectively verified through the logic for common natural language sentimental analysis datasets such as SST2 and Twitter, as well as synthetic dataset GOOD-Motif, where labels are determined by certain structures. The spliced graph contains combined causal structures; therefore, it forms a causally valid sample when given the mixed label of all component graphs. For molecule/protein datasets with chemical property tasks, the assumption is strongly underpinned by experimental results, as evidenced by the improved or comparable results even when whole samples are randomly combined in Appendix E.1.1. Although we acknowledge that our assumption may not encompass all cases, it does make headway in addressing a substantial class of problems. As graph OOD generalization is a complex issue in practice, different techniques are required for varying domains and problems. No single method can be expected to resolve all unknown cases, and our future work aims to expand the scope of tasks addressed.

D.3 STRENGTHS OF LINEAR EXTRAPOLATION OVER INTERPOLATION

Data augmentation methods can introduce additional samples not covered by the training database to benefit model learning. Since we focus on tasks that are out-of-distribution instead of in-distribution, models are expected to extrapolate instead of interpolate to make predictions outside the training range. However, the distribution area where models cannot generalize to is also hardly reachable when generating augmentation samples using traditional interpolation techniques. Interpolation methods cannot provide any guarantees regarding solving graph distribution shifts. In contrast, theoretical and empirical analyses show that linear extrapolation can generalize over certain shifts. Specifically, it can be reasoned based on our theoretical studies. Theorem 5.1 establishes that feature linear extrapolation can achieve invariant prediction and generalize over distribution shifts regarding selected variant features under certain environment conditions. Theorem 3.1 establishes that structural linear extrapolation can create OOD samples covering at least two environments in opposite directions of the distribution, respecting size and base shifts each. Contrarily, in the case of interpolation, the constructions in the proofs would not hold, thus failing to build these guarantees.

D.4 METHOD APPLICABILITY WHEN THE DISTRIBUTION SHIFT TYPE IS UNKNOWN

Firstly, as we have evidenced, our two strategies for feature and structure can be combined to solve comprehensive shifts. Also, our proposed method is applicable when OOD knowledge of a dataset is not fully known, since we are able to choose the techniques as hyperparameter selection. This

allows the framework to automatically decide on using either method separately or combined, thus covering OOD tasks with structural, feature or complex shifts. Secondly, although we use specific expressions of base and size shifts in theoretical studies of structural shifts, discussions for base and size extrapolation are actually applicable to general local and global structural extrapolation, respectively, in the field of graph. Therefore, G-Splice can cover both local and global structural shifts, making it applicable towards various unknown distribution shifts. One potential evidence is that it performs favorably on multiple real-world datasets, which inevitably contain natural and unknown distribution shifts.

E BROADER IMPACTS

Addressing out-of-distribution (OOD) generalization presents a formidable challenge, particularly in the realm of graph learning. This issue is acutely exacerbated when conducting scientific experiments becomes cost-prohibitive or impractical. In many real-world scenarios, data collection is confined to certain domains, yet extrapolating this knowledge to broader areas, where experiment conduction proves difficult, is crucial. In focusing on a data-centric approach to the OOD generalization problem, we pave the way for the integration of graph data augmentation with graph OOD, a strategy with substantial potential for broad societal and scientific impact.

Our research adheres strictly to ethical guidelines and does not raise any ethical issues. It neither involves human subjects nor gives rise to potential negative social impacts or privacy and fairness issues. Furthermore, we foresee no potential for malicious or unintended usage of our work. Nonetheless, we acknowledge that all technological progress inherently carries risks. Consequently, we advocate for ongoing evaluation of the broader implications of our methodology across a range of contexts.

F LIMITATIONS

Our work builds upon the principle of Causal Additivity, a causal assumption widely applicable in graph classification tasks. This assumption can be verified theoretically and experimentally for a variety of graph classification tasks. However, we acknowledge that our assumption may not encompass all cases. As graph OOD generalization is a complex issue in practice, different domains and problems may required varying techniques and our method might not resolve all unknown cases. Our future work aims to expand the scope of tasks addressed.

For another, our work discusses shifts on both graph structure and feature. By respective considerations, while G-splice can solve structure shifts, it augments structural OOD samples, which creates additional shift when facing feature-OOD situations. FeatX stands in the similar situation, introducing extra shifts for structural OOD problems. When combined, G-Splice and FeatX can solve both types of shifts and is suitable for addressing complex OOD situations. As "all medicine has its side effects", their concurrent use would create extra shifts if the problem does not involve both shifts. Given that an OOD dataset only contain one type between structure/feature shifts, the performance gain might not be so ideal when using two methods concurrently. However, this does not impair their applicability when OOD knowledge of the dataset is not fully known, since we are able to choose the techniques similarly as hyperparameter selection.

In addition, our current work does not discuss link prediction, which is an important task in graph learning. Thus our future work aims to expand the scope of tasks addressed. Furthermore, the proposed methods are designed to cover complex shifts of multiple types, therefore the hyperparameter selections including selection of techniques require certain amounts of pre-computation, which sets prerequisites in computational resources.

G THEORETICAL PROOFS

This section presents comprehensive proofs for all the theorems mentioned in this paper, along with the derivation of key intermediate results and necessary discussions.

Theorem 3.1 *Given an N-sample training dataset $D\{G_{tr}\}$, its N-dimension structural linear extrapolation can generate sets $D\{G_1\}$ and $D\{G_2\}$ s.t. $(G_1)_{env} < (G_{tr})_{env} < (G_2)_{env}$ for $\forall G_{tr}, G_1, G_2$,*

where $<$ denotes "less in size" for size extrapolation and "lower base complexity" for base extrapolation.

Proof. Considering size extrapolation, we prove that 1.sets $D\{G_1\}$ and $D\{G_2\}$ contain graph sizes achievable by N-dimension structural linear extrapolation; 2.$|\boldsymbol{X}|_{G_1} < |\boldsymbol{X}|_{G_{tr}} < |\boldsymbol{X}|_{G_2}$ holds for $\forall G_{tr}, G_1, G_2$.

For N-dimension structural linear extrapolation on training data $D\{G_{tr}\}$, we have Eq. 2:

$$G_{sle}^N = \sum_{i=1}^{N} a_i \cdot G_i + \sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij} \cdot (G_j - G_i) = \mathbf{a}^\top \mathbf{G} + \langle B, \mathbf{1}\mathbf{G}^\top - \mathbf{G}\mathbf{1}^\top \rangle_F.$$

Let the largest and smallest graph G_{ma} and G_{mi} in $D\{G_{tr}\}$ be indexed $i = ma$ and $i = mi$. We generate $D\{G_2\}$ using Eq. 2 with the condition that $a_{ma} = 1$ and $\sum_{i=1}^{N} a_i \geq 2$. We generate $D\{G_1\}$ with the condition that $\sum_{i=1}^{N} a_i = 0$, $\sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij} = 1$ and $b_{(mi)j} = 1$. By Definition 3, $D\{G_1\}$ and $D\{G_2\}$ contain graph sizes achievable by N-dimension structural linear extrapolation.

For $\forall G_2 \in D\{G_2\}$, since $a_{ma} = 1$ and $\sum_{i=1}^{N} a_i \geq 2$, G_2 contains multiple graphs spliced together; then we have

$$|\boldsymbol{X}|_{G_2} > |\boldsymbol{X}|_{G_{ma}} \geq |\boldsymbol{X}|_{G_{tr}} \tag{6}$$

for $\forall G_{tr} \in D\{G_{tr}\}$. For $\forall G_1 \in D\{G_1\}$, since $\sum_{i=1}^{N} a_i = 0$, $\sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij} = 1$ and $b_{(mi)j} = 1$, G_1 contains only one single subgraph extracted from G_{mi} and another graph; then we have

$$|\boldsymbol{X}|_{G_1} < |\boldsymbol{X}|_{G_{mi}} \leq |\boldsymbol{X}|_{G_{tr}} \tag{7}$$

for $\forall G_{tr} \in D\{G_{tr}\}$. Therefore, $|\boldsymbol{X}|_{G_1} < |\boldsymbol{X}|_{G_{tr}} < |\boldsymbol{X}|_{G_2}$ holds for $\forall G_{tr}, G_1, G_2$.

Considering base extrapolation, we prove that 1.sets $D\{G_1\}$ and $D\{G_2\}$ contain graph bases achievable by N-dimension structural linear extrapolation; 2.$\mathcal{B}_{G_1} < \mathcal{B}_{G_{tr}} < \mathcal{B}_{G_2}$ holds for $\forall G_{tr}, G_1, G_2$, where $\mathcal{B}$ denotes the base graph and "$<$" denotes less complex in graph base. Note that graph bases can be numerically indexed for ordering and comparisons, such as the Bemis-Murcko scaffold algorithm (Bemis and Murcko, 1996).

For N-dimension structural linear extrapolation on training data $D\{G_{tr}\}$, following Eq. 2, let the graphs with the most and least complex graph base G_{mo} and G_{le} in $D\{G_{tr}\}$ be indexed $i = mo$ and $i = le$. We generate $D\{G_2\}$ using Eq. 2 with the condition that $a_{mo} = 1$ and $\sum_{i=1}^{N} a_i \geq 2$. We generate $D\{G_1\}$ with the condition that $\sum_{i=1}^{N} a_i = 0$, $\sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij} = 1$ and $b_{(le)j} = 1$, with $(G_j - G_i)$ being a causal graph extraction. By Definition 4, $D\{G_1\}$ and $D\{G_2\}$ contain graph bases achievable by N-dimension structural linear extrapolation.

For $\forall G_2 \in D\{G_2\}$, since $a_{mo} = 1$ and $\sum_{i=1}^{N} a_i \geq 2$, G_2 contains multiple graphs spliced together including the most complex base; then we have

$$\mathcal{B}_{G_2} > \mathcal{B}_{G_{mo}} \geq \mathcal{B}_{G_{tr}} \tag{8}$$

for $\forall G_{tr} \in D\{G_{tr}\}$, adding upon $\mathcal{B}_{G_{mo}}$ to create more complex base graphs. For $\forall G_1 \in D\{G_1\}$, since $\sum_{i=1}^{N} a_i = 0$, $\sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij} = 1$ and $b_{(le)j} = 1$ with $(G_j - G_i)$ being a causal graph extraction, G_1 contains only a single causal subgraph extracted from G_{le}; then we have

$$\mathcal{B}_{G_1} < \mathcal{B}_{G_{le}} \leq \mathcal{B}_{G_{tr}} \tag{9}$$

for $\forall G_{tr} \in D\{G_{tr}\}$, essentially creating structural linear extrapolations containing no base graphs. Therefore, $\mathcal{B}_{G_1} < \mathcal{B}_{G_{tr}} < \mathcal{B}_{G_2}$ holds for $\forall G_{tr}, G_1, G_2$.

This completes the proof. □

Theorem 3.2 *Given an N-sample training dataset $D\{G_{tr}\}$ and its true labeling function for the target classification task $f(G)$, if $D\{G_{sle}^N\}$ is a graph set sampled from the N-dimension structural linear extrapolation of $D\{G_{tr}\}$ and Assumption 1 holds, then for $\forall (G_{sle}^N, y) \in D\{G_{sle}^N\}$, $y = f(G_{sle}^N)$.*

Proof. By Definition 2, for N-dimension structural linear extrapolation on training data $D\{G_{tr}\}$, for $\forall(G^N_{sle}, y)$ we have G^N_{sle}

$$G^N_{sle} = \sum_{i=1}^{N} a_i \cdot G_i + \sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij} \cdot (G_j - G_i) = \mathbf{a}^\top \mathbf{G} + \langle B, \mathbf{1}\mathbf{G}^\top - \mathbf{G}\mathbf{1}^\top \rangle_F,$$

and the label y for G^N_{sle}

$$\begin{aligned} y &= (\sum_{i=1}^{N} a_i \cdot y_i + \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}b_{ij} \cdot y_j)/(\sum_{i=1}^{N} a_i + \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}b_{ij}) \\ &= (\mathbf{a}^\top \mathbf{y} + \langle C \circ B, \mathbf{1}\mathbf{y}^\top \rangle_F)/(\mathbf{a}^\top \mathbf{1} + \langle C, B \rangle_F). \end{aligned}$$

$\mathbf{a}^\top \mathbf{G}$ splices $\sum_{i=1}^{N} a_i$ graphs together, and since $[G_1, \ldots, G_N]$ are the N graphs from $D\{G_{tr}\}$, each of G_i contains one and only one causal graph. Under the causal additivity of Assumption 1, given $G' = G_1 + G_2$, we have $f(G') = ay_1 + (1-a)y_2$. With a fair approximation of $a = 1 - a = 1/2$, we can feasibly obtain $f(G') = (y_1 + y_2)/2$. Recursively, for $\mathbf{a}^\top \mathbf{G}$ we can derive

$$f(\mathbf{a}^\top \mathbf{G}) = (\sum_{i=1}^{N} a_i \cdot y_i)/(\sum_{i=1}^{N} a_i). \tag{10}$$

$\langle B, \mathbf{1}\mathbf{G}^\top - \mathbf{G}\mathbf{1}^\top \rangle_F$ splices $\sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij}$ extracted subgraphs together. Among them, $\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}b_{ij}$ are causal subgraphs, while the others are environmental subgraphs. Similarly, using the causal additivity of Assumption 1 in a recursive manner, we can derive

$$f(\langle B, \mathbf{1}\mathbf{G}^\top - \mathbf{G}\mathbf{1}^\top \rangle_F) = (\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}b_{ij} \cdot y_i)/(\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}b_{ij}). \tag{11}$$

Combining the results of Eq. 10 and Eq. 11, using Assumption 1 in a recursive manner, we can derive for $\mathbf{a}^\top \mathbf{G} + \langle B, \mathbf{1}\mathbf{G}^\top - \mathbf{G}\mathbf{1}^\top \rangle_F$:

$$f(\mathbf{a}^\top \mathbf{G} + \langle B, \mathbf{1}\mathbf{G}^\top - \mathbf{G}\mathbf{1}^\top \rangle_F) = (\sum_{i=1}^{N} a_i \cdot y_i + \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}b_{ij} \cdot y_j)/(\sum_{i=1}^{N} a_i + \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}b_{ij}). \tag{12}$$

By Definition 2, we have

$$\begin{aligned} f(G^N_{sle}) &= f(\mathbf{a}^\top \mathbf{G} + \langle B, \mathbf{1}\mathbf{G}^\top - \mathbf{G}\mathbf{1}^\top \rangle_F) \\ &= (\sum_{i=1}^{N} a_i \cdot y_i + \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}b_{ij} \cdot y_j)/(\sum_{i=1}^{N} a_i + \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}b_{ij}) = y. \end{aligned}$$

Therefore, for $\forall(G^N_{sle}, y) \in D\{G^N_{sle}\}$, we have $y = f(G^N_{sle})$.

This completes the proof. □

Theorem 4.1 *Given the causal graph (Figure 1(b)) and assuming a bijective causal mapping between C and Y, for two same-class different-environment graphs $G = G_{y,\epsilon}$ and $G' = G_{y,\epsilon'}$, let $G_s \subseteq G$ and $G'_s \subseteq G'$ represent the subgraphs of G and G'. Let $I(\cdot,\cdot) \in [0,1]$ a similarity function in the graph hidden feature space and $f_z : \mathcal{G} \to \mathbb{R}^f$ is a feature mapping that reversely infers hidden features,* e.g., *C and S_1, then:*

(1) Given the invariant subgraphs of G and G', G_{inv} and G'_{inv}, the similarity of the subgraphs defined in the corresponding graph feature space can be represented as $I(f_z(G_{inv}), f_z(G'_{inv}))$. It follows that the value of this similarity reaches its maximum 1;

(2) Given a subgraph set $\mathbf{G}_s = \{G_s | \exists G'_s,\ I(f_z(G_s), f_z(G'_s)) = 1\}$, the invariant subgraph G_{inv} of G can be obtained by optimizing the objective: $G_{inv} = \text{argmax}_{G_s \in \mathbf{G}_s} |G_s|$.

Proof. The invariant subgraphs G_{inv} in the same class Y share the same features inferred by f_z because of the bijective causal mapping assumption. This implies that the similarity of invariant subgraphs in the same class is able to reach the maximal value of 1. In contrast, subgraphs affected by S_1 and S_2 do not have this property since S_1 and S_2 are noises fluctuating frequently, leading to diverse G_{s1} and G_{s2} that are assumed not to be matched in different graphs. Concretely, this mismatching is caused by the differences in feature dimensions corresponding to S_1 and S_2, which leads to similarities strictly lower than 1, *i.e.*, $\forall G_s, \exists G_n \subseteq G_s, G_n \neq \emptyset$ s.t. $G_n \in G_{s1} \cup G_{s2} \Leftrightarrow \forall G'_s, I(f_z(G_s), f_z(G'_s)) < 1$. Note that our causal graph is the general case covering the common SCMs of covariate shift, FIIF, and PIIF assumptions when latent variable S_2 is not considered.

Proof of (1): According to the aforementioned assumption, since G and G' have the same label, it follows that $f_z(G_{\text{inv}}) = f_z(G'_{\text{inv}})$, which directly results in $I(f_z(G_{\text{inv}}), f_z(G'_{\text{inv}})) = 1$

Proof of (2): First, given the subgraph set $\mathbf{G}_s$, it follows that $\forall G_s \in \mathbf{G}_s, G_s \subseteq G_{\text{inv}}$. Otherwise if G_s contains subgraphs of G_{env}, G_{s1}, or G_{s2} (Figure 1(b)), the different G_{env} caused by ϵ, ϵ' and the fluctuations in S_1, S_2 will lead to $I(f(G_s), f(G'_s)) < 1$ as discussed above. Therefore, $I(f(G_s), f(G'_s)) = 1 \Rightarrow G_s \subseteq G_{\text{inv}}$. In addition, according to (1), G_{inv} is also included in the set $\mathbf{G}_s$ since $I(f_z(G_{\text{inv}}), f_z(G'_{\text{inv}})) = 1$, which satisfies the definition of $\mathbf{G}_s$. Therefore, the solution for both "$\forall G_s \in \mathbf{G}_s, G_s \subseteq G_{\text{inv}}$" and "$G_{\text{inv}} \in \mathbf{G}_s$" reveals that G_{inv} is the largest subgraph in $\mathbf{G}_s$, i.e., $G_{\text{inv}} = \text{argmax}_{G_s \in \mathbf{G}_s} |G_s|$. □

Theorem 5.1 *If (1) $\exists(\boldsymbol{X}_1, \cdots, \boldsymbol{X}_j) \in P^{train}$ from at least 2 environments, s.t. $(\boldsymbol{X}_{1var}, \cdots, \boldsymbol{X}_{jvar})$ span $\mathbb{R}^j$, and (2) $\forall \boldsymbol{X}_1 \neq \boldsymbol{X}_2$, the GNN encoder of f_ψ maps $G_1 = (\boldsymbol{X}_1, \boldsymbol{A}, \boldsymbol{E})$ and $G_2 = (\boldsymbol{X}_2, \boldsymbol{A}, \boldsymbol{E})$ to different embeddings, then with $\hat{y} = f_\psi(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{E})$, $\hat{y} \perp\!\!\!\perp \boldsymbol{X}_{var}$ as $n_A \to \infty$.*

We theoretically prove the statements and Theorem 5.1 for FeatX. We propose to learn and apply a mask $\boldsymbol{M}$ and perturb the non-causal node features to achieve extrapolation w.r.t. $\boldsymbol{x}_{\text{var}}$, without altering the topological structure of the graph. Let the domain for $\boldsymbol{x}$ be denoted as $\mathcal{D}$, which is assumed to be accessible. Valid extrapolations must generate augmented samples with node feature $\boldsymbol{X}_A \in \mathcal{D}$ while $\boldsymbol{X}_A \nsim P^{\text{train}(\boldsymbol{X})}$. Since $\boldsymbol{x}$ is a vector, $\mathcal{D}$ is also a vector, in which each element gives the domain of an element in $\boldsymbol{x}$. We ensure the validity of extrapolation with the generalized modulo operation mod, which we define as

$$\boldsymbol{X} \bmod \mathcal{D} = \boldsymbol{X} + i * abs(\mathcal{D}), s.t. \boldsymbol{X} \bmod \mathcal{D} \in \mathcal{D}, \tag{13}$$

where i is any integer and $abs(\mathcal{D})$ calculates the range length of $\mathcal{D}$. Therefore, $\forall \boldsymbol{X} \in \mathbb{R}^p$, $(\boldsymbol{X} \bmod \mathcal{D}) \in \mathcal{D}$. Given each pair of samples $\boldsymbol{D}_{\varepsilon_1}$, $\boldsymbol{D}_{\varepsilon_2}$ with the same label y but different environments ε_1 and ε_2, FeatX produces

$$\begin{aligned} \boldsymbol{X}_A &= \boldsymbol{M} \times ((1+\lambda)\boldsymbol{X}_{\varepsilon_1} - \lambda' \boldsymbol{x}_{\varepsilon_2}) \bmod \mathcal{D} + \overline{\boldsymbol{M}} \times \boldsymbol{X}_{\varepsilon_1}, \\ (\boldsymbol{A}, \boldsymbol{E}) &= (\boldsymbol{A}_{\varepsilon_1}, \boldsymbol{E}_{\varepsilon_1}), \end{aligned}$$

where $\lambda, \lambda' \sim \mathcal{N}(a, b)$ is sampled for each data pair. During the process, the augmented samples form a new environment.

We prove Theorem 5.1, showing that, under certain conditions, FeatX substantially solves feature shifts on the selected variant features for node-level tasks. The proof also evidences that our extrapolation spans the feature space outside $P^{\text{train}}(\boldsymbol{X})$ for $\boldsymbol{x}_{\text{var}}$, transforming OOD areas to ID. Let n_A be the number of samples FeatX generates and f_ψ be the well-trained network with FeatX applied.

Proof. Condition is given that

$$\exists(\boldsymbol{X}_1, \cdots, \boldsymbol{X}_j) \in P^{\text{train}}, (\boldsymbol{X}_{1\,\text{var}}, \cdots, \boldsymbol{X}_{j\,\text{var}}) \text{span} \mathbb{R}^j. \tag{14}$$

Therefore, by definition,

$$\forall \boldsymbol{u} \in \mathbb{R}^j, \exists \boldsymbol{t} = (t_1, t_2, \cdots, t_j), t_1, t_2, \cdots, t_j \in \mathbb{R}^j, s.t. \boldsymbol{u} = t_1 \boldsymbol{X}_{1\,\text{var}} + \cdots + t_j \boldsymbol{X}_{j\,\text{var}}. \tag{15}$$

The operation to generate $\boldsymbol{X}_A$ gives

$$\boldsymbol{X}_A = \boldsymbol{M} \times ((1+\lambda)\boldsymbol{X}_{\varepsilon_1} - \lambda' \boldsymbol{X}_{\varepsilon_2}) \bmod \mathcal{D} + \overline{\boldsymbol{M}} \times \boldsymbol{X}_{\varepsilon_1}, \tag{16}$$

so we have

$$\boldsymbol{X}_{A\,\text{var}} = ((1+\lambda)\boldsymbol{X}_{\varepsilon_1\,\text{var}} - \lambda' \boldsymbol{X}_{\varepsilon_2\,\text{var}}) \bmod \mathcal{D}. \tag{17}$$

For $\forall \boldsymbol{u}, \exists \boldsymbol{t} = (t_1, t_2, \cdots, t_j)$ and $(\boldsymbol{X}_1, \cdots, \boldsymbol{X}_j) \in P^{\text{train}}$ from at least 2 environments. Without loss of generality, we assume that $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are from different environments ε_1 and ε_2. With $n_A \to \infty$, there will exist an augmentation sampled between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, and since $\lambda \sim \mathcal{N}(a, b), \lambda \in \mathbb{R}$,

$$\begin{aligned}\exists \boldsymbol{X}_A^1 \quad s.t. \boldsymbol{X}_{A\,\text{var}}^1 &= ((1+\lambda)\boldsymbol{X}_{1\,\text{var}} - \lambda' \boldsymbol{X}_{2\,\text{var}}) \bmod \mathcal{D} \\ &= (1+\lambda)\boldsymbol{X}_{1\,\text{var}} - \lambda' \boldsymbol{X}_{2\,\text{var}} + n_1 * abs(\mathcal{D}), 1+\lambda = t_1, -\lambda' = t_2,\end{aligned}$$

where n_1 is an integer. Equivalently,

$$\boldsymbol{X}_{A\,\text{var}}^1 = t_1 \boldsymbol{X}_{1\,\text{var}} + t_2 \boldsymbol{X}_{2\,\text{var}} + n_1 * abs(\mathcal{D}). \tag{18}$$

The augmentation sample $\boldsymbol{X}_A^1$ belongs to a new environment, thus in a different environment from $\boldsymbol{X}_1, \cdots, \boldsymbol{X}_j$. Similarly, with $n_A \to \infty$, there will exist an augmentation sampled between $\boldsymbol{X}_A^1$ and $\boldsymbol{X}_3$,

$$\begin{aligned}\exists \boldsymbol{X}_A^2 \quad s.t. \boldsymbol{X}_{A\,\text{var}}^2 &= ((1+\lambda)\boldsymbol{X}_{A\,\text{var}}^1 - \lambda' \boldsymbol{X}_{3\,\text{var}}) \bmod \mathcal{D} \\ &= (1+\lambda)\boldsymbol{X}_{A\,\text{var}}^1 - \lambda' \boldsymbol{X}_{3\,\text{var}} + n_2 * abs(\mathcal{D}), \lambda = 0, -\lambda' = t_3\end{aligned}$$

where n_2 is an integer. Equivalently,

$$\boldsymbol{X}_{A\,\text{var}}^2 = \boldsymbol{X}_{A\,\text{var}}^1 + t_3 \boldsymbol{X}_{3\,\text{var}} + n_2 * abs(\mathcal{D}) = t_1 \boldsymbol{X}_{1\,\text{var}} + t_2 \boldsymbol{X}_{2\,\text{var}} + t_3 \boldsymbol{X}_{3\,\text{var}} + (n_1 + n_2) * abs(\mathcal{D}). \tag{19}$$

The augmentation sample $\boldsymbol{X}_A^2$ also belongs to the new environment.

Recursively, with $n_A \to \infty$, there will exist an augmentation

$$\exists \boldsymbol{X}_A^{j-1} \quad s.t. {\boldsymbol{X}_A^{j-1}}_{\text{var}} = t_1 \boldsymbol{X}_{1\,\text{var}} + t_2 \boldsymbol{X}_{2\,\text{var}} + \cdots + t_j \boldsymbol{X}_{j\,\text{var}} + (n_1 + n_2 + \cdots + n_{j-1}) * abs(\mathcal{D}). \tag{20}$$

Since for $\boldsymbol{u}$, we have $\boldsymbol{u} = t_1 \boldsymbol{X}_{1\,\text{var}} + \cdots + t_j \boldsymbol{X}_{j\,\text{var}}$, therefore, ${\boldsymbol{X}_A^{j-1}}_{\text{var}} = \boldsymbol{u} + (n_1 + n_2 + \cdots + n_{j-1}) * abs(\mathcal{D})$. With $\boldsymbol{u} \in \mathbb{R}^j$ and ${\boldsymbol{X}_A^{j-1}}_{\text{var}} = ((1+\lambda){\boldsymbol{X}_A^{j-2}}_{\text{var}} - \lambda' \boldsymbol{X}_{j\,\text{var}}) \bmod \mathcal{D} \in \mathbb{R}^j$ by the definition of $\bmod \mathcal{D}$, we have

$$(n_1 + n_2 + \cdots + n_{j-1}) * abs(\mathcal{D}) = 0 \text{ and } {\boldsymbol{X}_A^{j-1}}_{\text{var}} = \boldsymbol{u}. \tag{21}$$

Therefore, we prove that with $n_A \to \infty$,

$$\forall \boldsymbol{u} \in \mathbb{R}^j, \text{ there exists an augmentation sample } \boldsymbol{X}_A^{j-1} \quad s.t. {\boldsymbol{X}_A^{j-1}}_{\text{var}} = \boldsymbol{u}. \tag{22}$$

That is, the extrapolation strategy of FeatX spans the feature space for $\boldsymbol{x}_{\text{var}}$.

With the above result, for the feature space of $\boldsymbol{x}_{\text{var}}$, every data point is reachable. As $n_A \to \infty$, every data point of $\boldsymbol{x}_{\text{var}}$ is reached at least once. Let a group of samples with selected and preserved causal features $\boldsymbol{X}_{\text{inv}^*}$ be $\boldsymbol{M} \times \boldsymbol{X}_{\text{var}} + \overline{\boldsymbol{M}} \times \boldsymbol{X}_{\text{inv}^*}$, where $\boldsymbol{X}_{\text{var}} \leftarrow \forall \boldsymbol{x}_{\text{var}} \in \mathbb{R}^j$. Since $\forall \boldsymbol{X}_1 \neq \boldsymbol{X}_2$, the GNN encoder maps $G_1 = (\boldsymbol{X}_1, \boldsymbol{A}, \boldsymbol{E})$ and $G_2 = (\boldsymbol{X}_2, \boldsymbol{A}, \boldsymbol{E})$ to different embeddings, all different samples from $\boldsymbol{M} \times \boldsymbol{X}_{\text{var}} + \overline{\boldsymbol{M}} \times \boldsymbol{X}_{\text{inv}^*}$ are encoded into different embeddings, while all having the same label y. For the well-trained network f_ψ, the group of embeddings $\boldsymbol{Z}|(\boldsymbol{M} \times \boldsymbol{X}_{\text{var}} + \overline{\boldsymbol{M}} \times \boldsymbol{X}_{\text{inv}^*})$ are all predicted into class $\hat{y} = y$. In this case,

$$\forall \boldsymbol{X}_{\text{var}} \in \mathbb{R}^j, \quad \hat{y} = f_\psi(\boldsymbol{M} \times \boldsymbol{X}_{\text{var}} + \overline{\boldsymbol{M}} \times \boldsymbol{X}_{\text{inv}^*}, \boldsymbol{A}, \boldsymbol{E}) = y, \tag{23}$$

therefore

$$\hat{y} \perp\!\!\!\perp \boldsymbol{X}_{\text{var}} \quad \text{as} \quad n_A \to \infty. \tag{24}$$

This completes the proof. □

Theorem 5.1 states that, given sufficient diversity in environment information and expressiveness of GNN, FeatX can achieve invariant prediction regarding the selected variant features. Therefore, FeatX possesses the capability to generalize over distribution shifts on the selected variant features. Extending on the accuracy of non-causal selection, if $\boldsymbol{x}_{\text{var}^*} = \boldsymbol{x}_{\text{var}}$, we achieve causally-invariant prediction in feature-based OOD tasks. Thus, FeatX possesses the potential to solve feature distribution shifts.

H Experimental Details

We further describe experimental details in the following sections.

H.1 DATASET DETAILS

To evidence the generalization improvements of structure extrapolation, we evaluate G-Splice on 8 graph-level OOD datasets with structure shifts. We adopt 5 datasets from the GOOD benchmark (Gui et al., 2022a), GOODHIV-size, GOODHIV-scaffold, GOODSST2-length, GOODMotif-size, and GOODMotif-base, using the covariate shift split from GOOD. GOOD-HIV is a real-world molecular dataset with shift domains scaffold and size. The first one is Bemis-Murcko scaffold (Bemis and Murcko, 1996) which is the two-dimensional structural base of a molecule. The second one is the number of nodes in a molecular graph. GOOD-SST2 is a real-world natural language sentimental analysis dataset with sentence lengths as domain, which is equivalent to the graph size. GOOD-Motif is a synthetic dataset specifically designed for structure shifts. Each graph is generated by connecting a base graph and a motif, with the label determined by the motif solely. The shift domains are the base graph type and the graph size. We construct another natural language dataset Twitter (Yuan et al., 2020) following the OOD splitting process of GOOD, with length as the shift domain. In addition, we adopt protein dataset DD and molecular dataset NCI1 following Bevilacqua et al. (2021), both with size as the shift domain. All datasets possess structure shifts as we have discussed, thus proper benchmarks for structural OOD generalization.

To show the OOD the generalization improvements of feature extrapolation, we evaluate FeatX on 5 graph OOD datasets with feature shifts. We adopt 5 datasets of the covariate shift split from the GOOD benchmark. GOOD-CMNIST is a semi-artificial dataset designed for node feature shifts. It contains image-transformed graphs with color features manually applied, thus the shift domain color is structure-irrelevant. The other 4 datasets are node-level. GOOD-Cora is a citation network dataset with "word" shift, referring to the word diversity feature of a node. The input is a small-scale citation network graph, in which nodes represent scientific publications and edges are citation links. The shift domain is word, the word diversity defined by the selected-word-count of a publication. GOOD-Twitch is a gamer network dataset, with the node feature "language" as shift domain. The nodes represent gamers and the edge represents the friendship connection of gamers. The binary classification task is to predict whether a user streams mature content. The shift domain of GOOD-Twitch is user language. GOOD-WebKB is a university webpage network dataset. A node in the network represents a webpage, with words appearing in the webpage as node features. Its 5-class prediction task is to predict the owner occupation of webpages, and the shift domain is university, which is implied in the node features. GOOD-CBAS is a synthetic dataset. The input is a graph created by attaching 80 house-like motifs to a 300-node Barabási–Albert base graph, and the task is to predict the role of nodes. It includes colored features as in GOOD-CMNIST so that OOD algorithms need to tackle node color differences, which is also typical as feature shift. All shift domains are structure-irrelevant and provide specific evaluation for feature extrapolation.

Following prior works (Wu et al., 2022b; Gui et al., 2022a), we also create another synthetic dataset FSMotif. The GOOD benchmark we use for major evaluation in the paper does not contain OOD datasets with shifts on both structure and feature, which cannot provide evaluation for the combined effectiveness of FLE and SLE. We create FSMotif, with complex shifts on both structure and feature, to prove the superiority of our methods when used concurrently. FSMotif is a synthetic dataset where each graph is generated by connecting a base graph and a motif, with the label determined by the motif solely and all nodes given color features. The shift domains are 1.the base graph type and the color feature, and 2.the graph size and the color feature. Specifically, we generate graphs using seven colors, five label irrelevant base graphs (wheel, tree, ladder, star, and path), and three label determining motifs (house, cycle, and crane).

H.2 SETUP DETAILS

We conduct experiments on 8 datasets with 16 baseline methods to evaluate G-Splice, and on 5 datasets with 16 baselines for FeatX. As a common evaluation protocol, datasets for OOD tasks provides OOD validation/test sets (Gui et al., 2022a; Bevilacqua et al., 2021) to evaluate the model's OOD generalization abilities. Some datasets also provide ID validation/test sets for comparison (Gui et al., 2022a). For all experiments, we select the best checkpoints for OOD tests according to results on OOD validation sets; ID validation and ID test are also used for comparison if available. For graph prediction and node prediction tasks, we respectively select strong and commonly acknowledged GNN backbones. For each dataset, we use the same GNN backbone for all baseline methods for fair comparison. For graph prediction tasks, we use GIN-Virtual Node (Xu et al., 2019a; Gilmer et al.,

2017) as the GNN backbone. As an exception, for GOOD-Motif we adopt GIN (Xu et al., 2019a) as the GNN backbone, since we observe from experiments that the global information provided by virtual nodes would interrupt the training process here. For node prediction tasks, we adopt GraphSAINT (Zeng et al., 2020) and use GCN (Kipf and Welling, 2017) as the GNN backbone. For all the experiments, we use the Adam optimizer, with a weight decay tuned from the set {0, 1e-2, 1e-3, 1e-4} and a dropout rate of 0.5. The number of convolutional layers in GNN models for each dataset is tuned from the set {3, 5}. We use mean global pooling and the RELU activation function, and the dimension of the hidden layer is 300. We select the maximum number of epochs from {100, 200, 500}, the initial learning rate from {1e-3, 3e-3, 5e-3, 1e-4}, and the batch size from {32, 64, 128} for graph-level and {1024, 4096} for node-level tasks. All models are trained to converge in the training process. For computation, we generally use one NVIDIA GeForce RTX 2080 Ti for each single experiment.

H.3 Hyperparameter Selection

In all experiments, we perform hyperparameter search to obtain experimental results that can well-reflect the performance potential of models. For each dataset and method, we search from a hyperparameter set and select the optimal one based on OOD validation metric scores.

For each baseline method, we tune one or two algorithm-specific hyperparameters. For IRM and Deep Coral, we tune the weight for penalty loss from {1e-1, 1, 1e1, 1e2} and {1, 1e-1, 1e-2, 1e-3}, respectively. For VREx, we tune the weight for VREx's loss variance penalty from {1, 1e1, 1e2, 1e3}. For GroupDRO, we tune the step size from {1e-1, 1e-2, 1e-3}. For DANN, we tune the weight for domain classification penalty loss from {1, 1e-1, 1e-2, 1e-3}. For Graph Mixup, we tune the alpha value of its Beta function from {0.4, 1, 2}. The Beta function is used to randomize the lamda weight, which is the weight for mixing two instances up. For DIR, we tune the causal ratio for selecting causal edges from {0.2, 0.4, 0.6, 0.8} and loss control from {1e1, 1, 1e-1, 1e-2}. For EERM, we tune the learning rate for reinforcement learning from {1e-2, 1e-3, 5e-3, 1e-4} and the beta value to trade off between mean and variance from {1, 2, 3}. For SRGNN, we tune the weight for shift-robust loss calculated by central moment discrepancy from {1e-4, 1e-5, 1e-6}. For DropNode, DropEdge and MaskFeature, we tune the drop/mask percentage rate from {0.05, 0.1, 0.15, 0.2, 0.3}. For FLAG, we set the number of ascending steps $M = 3$ and tune the ascent step size from the set {1e-2, 1e-3, 5e-3, 1e-4}. For LISA, we tune the parameters of the Beta function in the same way as Graph Mixup. For G-Mixup, we set the augmentation number to 10, tune the augmentation ratio from {0.1, 0.2, 0.3} and the lambda range from {[0.1,0.2], [0.2,0.3]}. For GIL, we tune IGA lambda value from {1e-2, 1e-3, 1e-4} and set top ratio of subgraphs and number of environments by its originally reported optimum. For CIGA, we tune the size ratio of the causal subgraphs from {0.4, 0.6, 0.8}, while contrastive loss and hinge loss weights from {0.5, 1, 2}.

For G-Splice, we tune the percentage of augmentation from {0.6, 0.8, 1.0}. The actual number of component graphs f is tuned from {2, 3, 4}, and the augmentation selection is tuned as a 3-digit binary code representing the 3 options, with at least one option applied. For the pre-training of the bridge generation, hyperparameters regularizing the bridge attribute and KL divergence α and β are tuned from {1.5, 1, 0.5, 0.1}. When the additional VREx-like regularization is applied, we tune the weight of loss variance penalty from {1, 1e1, 1e2}. For FeatX, we tune the the shape parameter a and scale parameter b of the gamma function $\Gamma(a, b)$ from {2, 3, 5, 7, 9} and {0.5, 1.0, 2.0}, respectively.

I Ablation Studies

I.1 Bridge Generation Studies for G-Splice

I.1.1 VAE as Bridge Generator

In this work, we adopt conditional VAE (Kingma and Welling, 2013; Kipf and Welling, 2016; Sohn et al., 2015) as the major bridge generator for G-Splice due to its adequate capability and high efficiency. We show empirically that VAEs are well suitable for our task.

We reconstruct the generation process with diffusion model (Ho et al., 2020), a generative model with high capability and favorable performances across multiple tasks. Diffusion models consist of a diffusion process which progressively distorts a data point to noise, and a generative denoising

process which approximates the reverse of the diffusion process. In our case, the diffusion process adds Gaussian noise independently on each node and edge features encoded into one-hot vectors at each time step. Then the denoising network is trained to predict the noises, and we minimize the error between the predicted noise and the true noise computed in closed-form. During sampling, we iteratively sample bridge indexes and attribute values, and then map them back to categorical values in order that we obtain a valid graph. We compare performances and computational efficiency of the two generative models. As a baseline for bridge generation, we also present the results of random bridges, where bridges of predicted number and corresponding attributes are randomly sampled from the group of component graphs. Note that we do not apply the regularization in these experiments.

Table 4: Comparison on bridge generation methods. G-Splice-Rand, G-Splice-VAE, and G-Splice-Diff show the performance of G-Splice on GOODHIV with random bridge, VAE generated bridge, diffusion model generated bridge, respectively. The train time ratio presents the entire training duration of a method, including module pre-training time, divided by the training duration of G-Splice-Rand in average.

Method	GOOD-HIV-size↑		GOOD-HIV-scaffold↑		Train time ratio
	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	
ERM	83.72±1.06	59.94±2.86	82.79±1.10	69.58±1.99	0.57
G-Splice-Rand	83.25±0.96	62.36±2.25	**84.33**±0.69	71.89±2.80	1
G-Splice-VAE	**84.75**±0.18	**64.46**±1.38	83.23±0.97	72.82±1.16	1.52
G-Splice-Diff	84.35±0.35	64.09±0.82	83.45±0.97	**72.95**±1.80	20.25

As can be observed in Table 4, OOD test results from the two generative models are comparable, both significantly improving over G-Splice-Rand. The diffusion model may be slightly limited in performance gain due to the discreteness approximations during sampling. The results implies the necessity of generative models in the splicing operation for overall structural extrapolation. Meanwhile, this shows that VAE is capable of the bridge generation task. In contract, the training duration of diffusion model is 13 times that of VAE due to the sampling processes through massive time steps. Overall, we obtain comparable performances from the two generative models, while VAEs are much less expensive computationally. Therefore, empirical results demonstrate that adopting VAE as our major bridge generator is well suitable.

I.1.2 Bridge Generation Design

As we have introduced in Sec 4.1, we generate bridges of predicted number along with corresponding edge attributes between given component graphs to splice graphs. We do not include new nodes as a part of the bridge, since we aim at preserving the local structures of the component graphs and extrapolating certain global features. More manually add-on graph structures provide no extrapolation significance, while their interpolation influence are not proven beneficial. We evidence the effectiveness of our design with experiments. We additionally build a module to generate nodes in the bridges. The number of nodes is predicted with a pre-trained predictive model and then a generative model generates the node features. Moreover, we evaluate the results with fixed instead of generated bridge attributes. The performances with our original bridge generator, node generation applied and edge attribute generation removed is summarized as follows. Note that we do not apply the regularization in these experiments.

Table 5: Comparison of bridge generation designs. G-Splice orig, G-Splice + node, and G-Splice - attr show the performance of G-Splice on GOODHIV with the original bridge generator, node generation applied and edge attribute generation removed, respectively.

Method	GOOD-HIV-size↑		GOOD-HIV-scaffold↑	
	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}
ERM	83.72±1.06	59.94±2.86	82.79±1.10	69.58±1.99
G-Splice orig	**84.75**±0.18	**64.46**±1.38	83.23±0.97	**72.82**±1.16
G-Splice + node	83.14±0.82	62.65±2.67	**84.67**±0.48	71.76±1.76
G-Splice - attr	84.50±0.44	64.13±0.62	83.41±1.10	72.07±1.52

As can be observed in Table 5, OOD test performances from the original bridge generator remains the highest. Without attribute generation, fixed bridge attributes degrades the overall performance due to the manual feature of the bridges, which may mislead the model with spurious information. When we include nodes as a part of the bridge, similarly the manually add-on graph structure may inject spurious information to the model and perturb the preservation of local structures, leading to limited improvements. This evidences the effectiveness of our design for bridge generation.

I.2 Comparison of Extrapolation Procedures for G-Splice

We evidence that certain extrapolation procedures specifically benefit size or base/scaffold shifts, as our theoretical analysis in Sec. 4 and 5. For size and base/scaffold shifts on GOODHIV and GOODMotif, we extrapolation with each of the three augmentation options, G_{inv}, $G_{\text{inv}} + f \cdot G_{\text{env}}$ and $f \cdot G$, individually and together, and compare the OOD performances. Note that we apply the VREx-like regularization in these experiments.

Table 6: Comparison of extrapolation procedures for G-Splice. Performances of G-Splice on GOOD-HIV and GOODMotif with augmentation options single causal subgraph, causal and environmental subgraphs spliced, whole graphs spliced, and all three options applied. Optimal show the performances with options selected after hyperparameter tuning.

G-Splice	GOOD-HIV-size↑		GOOD-HIV-scaffold↑		GOOD-Motif-size↑		GOOD-Motif-base↑	
	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}
G_{inv}	83.90±0.40	63.04±2.40	83.19±0.69	72.04±0.96	91.10±0.10	76.95±4.52	92.12±0.14	69.59±3.67
$G_{\text{inv}} + f \cdot G_{\text{env}}$	**85.40**±0.82	62.65±2.16	83.97±0.57	72.83±1.86	91.08±0.63	72.10±5.43	92.01±0.23	73.92±5.44
$f \cdot G$	84.75±0.18	63.94±1.46	**85.10**±0.67	73.14±1.05	91.93±0.21	**85.07**±4.50	92.12±0.15	76.19±10.99
All	85.09±0.61	63.16±1.38	83.47±0.45	72.39±0.52	**92.00**±0.30	82.86±2.53	92.01±0.16	80.09±12.10
Optimal	84.85±0.19	**65.56**±0.34	83.36±0.40	**73.28**±0.16	91.93±0.21	**85.07**±4.50	**92.14**±0.29	**83.96**±7.38

Let the three augmentation options, G_{inv}, $G_{\text{inv}} + f \cdot G_{\text{env}}$ and $f \cdot G$ be numbered 1, 2, and 3. The optimal augmentation options for GOOD-HIV-size, GOOD-HIV-scaffold, GOOD-Motif-size, and GOOD-Motif-base after hyperparameter tuning are 1+3, 2+3, 3, and 2+3, respectively. As can be observed from Table 6, G_{inv} and $f \cdot G$ have advantages in size shifts, while $G_{\text{inv}} + f \cdot G_{\text{env}}$ and $f \cdot G$ are better for base/scaffold shifts. This matches our theoretical analysis of augmentation procedures. For size distribution shifts, G_{inv} and $f \cdot G$ environments enable size extrapolation by creating smaller and larger graphs outside the training distribution, respectively. For base/scaffold distribution shifts, the two new environments respectively construct graphs without base/scaffold, and graphs with f base/scaffolds, achieving base extrapolation with new base/scaffolds introduced. Splicing whole graphs has the advantage of extrapolating to larger graphs, simplicity in operation, and little loss in local structural information. Extracting subgraphs allows better flexibility for G-Splice, making graphs smaller than the training size accessible. In addition, the performance gain from $f \cdot G$ shows the effectiveness of the simple splicing strategy by itself.

I.3 Ablation Studies on FeatX

FeatX enables extrapolation w.r.t. the selected variant features. By generating causally valid samples with OOD node features, FeatX essentially expands the training distribution range. Theoretical analysis evidences that our extrapolation spans the feature space outside $P^{\text{train}}(\boldsymbol{X})$ for $\boldsymbol{x}_{\text{var}}$, thereby transforming OOD areas to ID. We further show with experiments that extrapolation substantially benefits feature shifts in OOD tasks compared with interpolation, which can also improve generalization by boosting learning processes. In addition, we show that our invariance mask and variance score vectors succeed in selecting non-causal features by comparisons between perturbation on selected features and all features.

As can be observed from Table 7, whether selecting non-causal features and the choice between interpolation and extrapolation both show significant influence on generalization performances. In all three datasets, extrapolation performances exceed corresponding interpolation performances with a clear gap, demonstrating the benefits of extrapolation by generating samples in OOD area that interpolation cannot reach. In GOODWebKB, perturbing selected non-causal features achieve significant improvements over perturbing all features, regardless of interpolation and extrapolation. This

Table 7: Performances w/o feature selection and extrapolation for FeatX. We show the performance comparisons of interpolating or extrapolating the selected or all features on GOODCMNIST, GOOD-WebKB and GOODCBAS.

Feature	Perturbation	GOOD-CMNIST-color↑		GOOD-WebKB-university↑		GOOD-CBAS-color↑	
		ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}
ERM		77.96±0.34	28.60±2.01	38.25±0.68	14.29±3.24	89.29±3.16	76.00±3.00
All	Interpolation	76.62±0.46	29.61±2.54	37.56±3.67	15.59±2.48	92.00±0.15	81.76±1.75
All	Extrapolation	75.14±0.38	54.13±5.08	38.26±5.18	17.10±3.45	**93.25**±0.43	84.28±3.67
Selected	Interpolation	**78.15**±0.30	31.65±5.02	43.15±1.42	26.16±2.50	90.10±2.14	80.37±1.35
Selected	Extrapolation	69.54±1.51	**62.49**±2.12	**50.82**±0.00	**32.54**±8.98	92.86±1.17	**87.62**±2.43

evidences the effectiveness of non-causal feature selection using variance score vectors, empirically supporting our design. In GOODCMNIST and GOODCBAS, since the features are manually added colors, the effect of feature selection is not as obvious as in GOODWebKB, the real world dataset. Experimental results evidence the effectiveness of the strategies designed in FeatX.

J Metric Score and Loss Curves

We report the metric score curves and loss curves for part of the datasets in Figure 2-5. As can be observed from each pair of curves, our proposed methods, G-Splice and FeatX, consistently achieve better metric scores and lower loss compared with other baselines during the learning process. This evidences the substantial improvements achieved by structure and feature extrapolation, which benefits OOD generalization in essence.

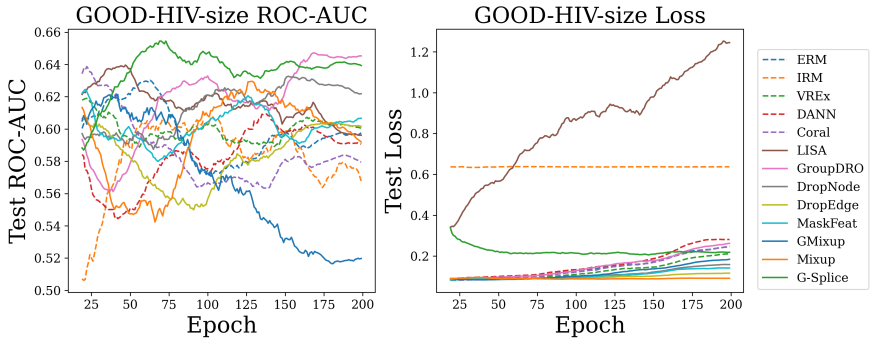

Figure 2: ROC-AUC score curve and loss curve for GOODHIV-size.

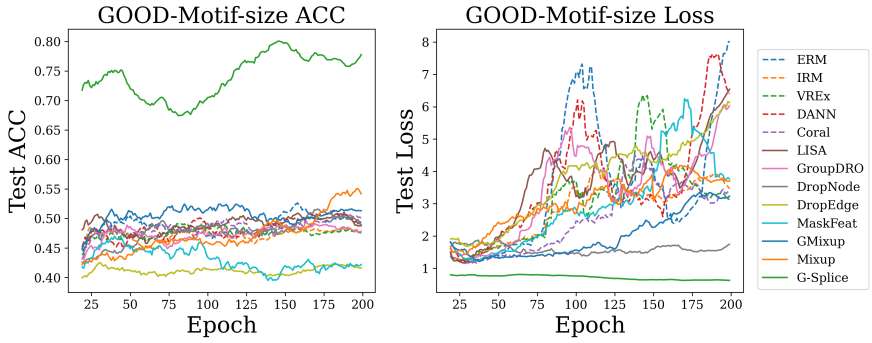

Figure 3: Accuracy score curve and loss curve for GOODMotif-size.

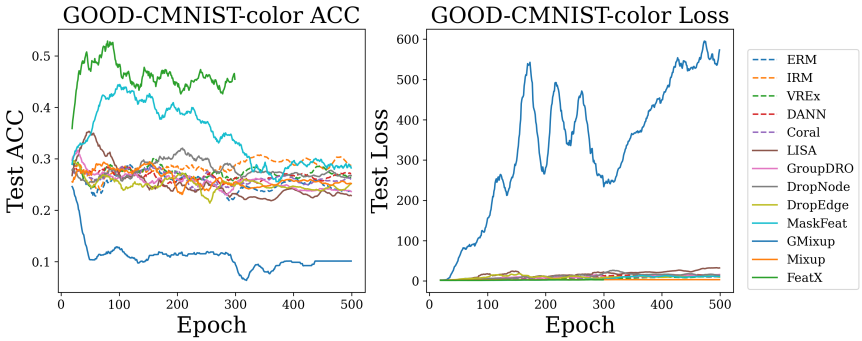

Figure 4: Accuracy score curve and loss curve for GOODCMNIST-color.

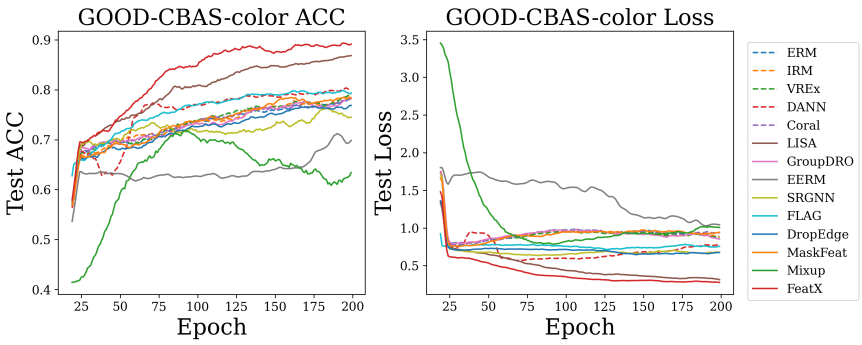

Figure 5: Accuracy score curve and loss curve for GOODCBAS-color.

REFERENCES

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.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. In *International conference on machine learning*, pages 511–520. PMLR, 2018.

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29, 2016.

Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. Size-invariant graph representations for graph classification extrapolations. In *International Conference on Machine Learning*, pages 837–851. PMLR, 2021.

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

Yimeng Chen, Ruibin Xiong, Zhi-Ming Ma, and Yanyan Lan. When does group invariant learning survive spurious correlations? *Advances in Neural Information Processing Systems*, 35:7038–7051, 2022a.

Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, Kaili Ma, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Learning causally invariant representations for out-of-distribution generalization on graphs. In *Advances in Neural Information Processing Systems*, 2022b.

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*, 2022c.

Aniket Anand Deshmukh, Yunwen Lei, Srinagesh Sharma, Urun Dogan, James W Cutler, and Clayton Scott. A generalization error bound for multi-class domain generalization. *arXiv preprint arXiv:1905.10392*, 2019.

John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang. Debiasing graph neural networks via learning disentangled causal substructure. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=ex60CCi5GS.

Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural networks for semi-supervised learning on graphs. *Advances in neural information processing systems*, 33:22092–22103, 2020.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14004–14013, 2020.

Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. GOOD: A graph out-of-distribution benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022a.

Shurui Gui, Hao Yuan, Jie Wang, Qicheng Lao, Kang Li, and Shuiwang Ji. Flowx: Towards explainable graph neural networks via message flows. *arXiv preprint arXiv:2206.12987*, 2022b.

Shurui Gui, Meng Liu, Xiner Li, Youzhi Luo, and Shuiwang Ji. Joint learning of label and environment causal independence for graph out-of-distribution generalization. *arXiv preprint arXiv:2306.01103*, 2023.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Hongyu Guo and Yongyi Mao. Intrusion-free graph mixup. *ArXiv*, abs/2110.09344, 2021.

Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-Mixup: Graph data augmentation for graph classification. *arXiv preprint arXiv:2202.07179*, 2022.

David Heckerman. *A tutorial on learning with Bayesian networks*. Springer, 1998.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

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.

Yinan Huang, Xingang Peng, Jianzhu Ma, and Muhan Zhang. 3dlinker: An e (3) equivariant variational autoencoder for molecular linker design. *arXiv preprint arXiv:2205.07309*, 2022.

Ilia Igashov, Hannes Stärk, Clément Vignac, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael Bronstein, and Bruno Correia. Equivariant 3d-conditional diffusion models for molecular linker design. *arXiv preprint arXiv:2210.05274*, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Boris Knyazev, Graham W Taylor, and Mohamed Amer. Understanding attention and generalization in graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Robust optimization as data augmentation for large-scale graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 60–69, 2022.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (REx). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

Kisoo Kwon, Kuhwan Jeong, Sanghyun Park, Sangha Park, Hoshik Lee, Seung-Yeon Kwak, Sungmin Kim, and Kyunghyun Cho. Extramix: Extrapolatable data augmentation for regression using generative models. 2022.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning invariant graph representations for out-of-distribution generalization. In *Advances in Neural Information Processing Systems*, 2022.

Yong Lin, Shengyu Zhu, Lu Tan, and Peng Cui. Zin: When and how to learn invariance without environment partition? *Advances in Neural Information Processing Systems*, 35:24529–24542, 2022.

Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph rationalization with environment-based augmentations. *arXiv preprint arXiv:2206.02886*, 2022.

Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. Heterogeneous risk minimization. In *International Conference on Machine Learning*, pages 6804–6814. PMLR, 2021a.

Meng Liu, Youzhi Luo, Limei Wang, Yaochen Xie, Hao Yuan, Shurui Gui, Haiyang Yu, Zhao Xu, Jingtun Zhang, Yi Liu, et al. DIG: A turnkey library for diving into graph deep learning research. *The Journal of Machine Learning Research*, 22(1):10873–10881, 2021b.

Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2021.

Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*, pages 15524–15543. PMLR, 2022.

Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2011.06.019. URL https://www.sciencedirect.com/science/article/pii/S0031320311002901.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pages 10–18. PMLR, 2013.

Joonhyung Park, Hajin Shim, and Eunho Yang. Graph transplant: Node saliency-guided graph mixup with local structure preservation. In *Proceedings of the First MiniCon Conference*, 2022.

Judea Pearl. *Causality*. Cambridge university press, 2009.

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

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

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

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning*, pages 4470–4479. PMLR, 2018.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*, 2019.

Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kunag. Stable learning via sample reweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5692–5699, 2020.

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

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

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

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Yongduo Sui, Xiang Wang, Jiancan Wu, An Zhang, and Xiangnan He. Adversarial causal augmentation for graph covariate shift. *arXiv preprint arXiv:2211.02843*, 2022.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

Hao Tang, Zhiao Huang, Jiayuan Gu, Bao-Liang Lu, and Hao Su. Towards scale-invariant graph-related problem solving by iterative homogeneous gnns. *Advances in Neural Information Processing Systems*, 33:15811–15822, 2020.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514*, 2021.

Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ, accepted as poster.

Petar Veličković, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. Neural execution of graph algorithms. *arXiv preprint arXiv:1910.10593*, 2019.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153, 2018.

Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Mixup for node and graph classification. In *Proceedings of the Web Conference 2021*, pages 3663–3674, 2021.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

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

Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat seng Chua. Discovering invariant rationales for graph neural networks. In *ICLR*, 2022b.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=ryGs6iA5Km.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? *arXiv preprint arXiv:1905.13211*, 2019b.

Keyulu Xu, Mozhi Zhang, Jingling Li, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. *arXiv preprint arXiv:2009.11848*, 2020.

Chenxiao Yang, Qitian Wu, Jiahua Wang, and Junchi Yan. Graph neural networks are inherently good generalizers: Insights by bridging gnns and mlps. *arXiv preprint arXiv:2212.09034*, 2022a.

Nianzu Yang, Kaipeng Zeng, Qitian Wu, Xiaosong Jia, and Junchi Yan. Learning substructure invariance for out-of-distribution molecular representations. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=2nWUNTnFijm.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. *arXiv preprint arXiv:2201.00299*, 2022.

Gilad Yehudai, Ethan Fetaya, Eli Meirom, Gal Chechik, and Haggai Maron. From local structures to size generalization in graph neural networks. In *International Conference on Machine Learning*, pages 11975–11986. PMLR, 2021.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33: 5812–5823, 2020.

Junchi Yu, Jian Liang, and Ran He. Finding diverse and predictable subgraphs for graph domain generalization. *arXiv preprint arXiv:2206.09345*, 2022.

Junchi Yu, Jian Liang, and Ran He. Mind the label shift of augmentation-based graph ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11620–11630, 2023.

Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *arXiv preprint arXiv:2012.15445*, 2020.

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graph-SAINT: Graph sampling based inductive learning method. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJe8pkHFwS.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Xingxuan Zhang, Linjun Zhou, Renzhe Xu, Peng Cui, Zheyan Shen, and Haoxin Liu. NICO++: Towards better benchmarking for domain generalization. *arXiv preprint arXiv:2204.08040*, 2022.

Qi Zhu, Natalia Ponomareva, Jiawei Han, and Bryan Perozzi. Shift-robust GNNs: Overcoming the limitations of localized graph training data. *Advances in Neural Information Processing Systems*, 34, 2021.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.