

A Invariant Risk Minimization and Graph Out-of-Distribution Generalization

IRM [1] Invariant Risk Minimization (IRM) is a framework for learning predictors that remain robust under distribution shifts by enforcing that the same classifier is simultaneously optimal across multiple training environments. In its ideal form, IRM solves the bi-level problem

$$\min_{\phi, w} \sum_{e \in \mathcal{E}} R^e(w \circ \phi) \quad \text{s.t.} \quad w \in \arg \min_{\bar{w}} R^e(\bar{w} \circ \phi) \text{ for all } e \in \mathcal{E}, \quad (11)$$

where ϕ is a feature extractor, w a classifier, and R^e the risk in environment e . Many approaches for out-of-distribution generalization on graphs are based on the IRM framework. In practice, IRMv1 approximates this constraint by adding a penalty on the squared norm of the gradient of each environment’s risk with respect to w , encouraging the learned representation to capture only invariant (i.e., causal) features:

$$\min_{\phi, w} \sum_{e \in \mathcal{E}} R^e(w \circ \phi) + \lambda \sum_{e \in \mathcal{E}} \left\| \nabla_{w|w=1} R^e(w \circ \phi) \right\|_2^2, \quad (12)$$

Graph out-of-distribution methods typically build on the frameworks of IRM, which seek to extract the causal subgraph/features and learn an equipredictive classifier across environments in order to capture invariant features, which demands that the dataset be divided into well-defined environments. Depending on the environmental partitioning strategy, these environments fall into three main categories: Approaches such as LECI [8] and G-splice [20] depend on environment labels provided in the dataset, but these labels are not always available and incur high annotation costs. Other methods such as GIL [19] and OOD-GCL [18] use unsupervised clustering to infer environment labels, which may not always align well with real environment distribution. Other approaches such as DIR [33] explicitly create distinct environments by applying causal interventions to the dataset. However, designing causal interventions to generate training distributions should require domain expertise or incur additional overhead for different task, and unreasonable designed interventions may fail to remove all spurious features or even damage crucial information[25]. Such limitations relying on environment information hinder the deployment of these methods in real-world scenarios. Recent work has further shown that recovering real environment information is infeasible without external information[21]. In summary, the limitations of these invariant learning methods have prompted us to explore an alternative approach for uncovering causal subgraphs.

B Graph Data Generation Process

In graph data generation process presented in [2], C and S denote latent codes for the causal and spurious factors, respectively. The observed graph G is composed of two latent components: an causal subgraph G_c driven by the causal factor C , and a spurious subgraph G_s driven by the non-causal factor S , regardless of noise. The variable C causally influences the target Y , whereas S may vary across environments E . Depending on how S interacts with Y conditional on C , prior work typically distinguish two scenarios, i.e., (i) Fully Informative Invariant Features (FIIF) when $Y \perp S|C$ and (ii) Partial Informative Invariant Features (PIIF) when $Y \not\perp S|C$.

In case (i), the invariant factor C is fully informative (FIIF) to the target label Y , and the latent spurious factor S provide no further information. In case (ii), the invariant factor C is only partially informative (PIIF) about Y , spurious factor S can further provide additional information to aid the prediction of Y , however, as S is directly affected by E , it is not stable across different environments. The SCMs for the two scenarios are illustrated in Figure 7 and these two assumptions have been extensively discussed and empirically validated in prior work on out-of-distribution graph tasks [3, 4, 5, 8, 19, 24, 33, 36, 37] and is founded on Structural Causal Models [26].

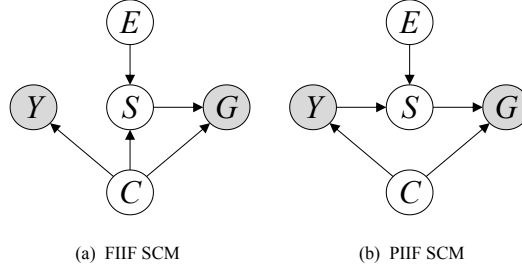


Figure 7: Structure causal models for graph data generation.

512 C Proofs for Theoretical Results

513 C.1 Proof of Lemma 1

514 *Proof.* By Assumption 1 $Y = f(G_c)$ for a fixed causal mechanism $f(\cdot)$ that does not vary with e .
 515 This means that for any value g_c of G_c , $P(Y | G_c = g_c)$ is defined entirely by $f(g_c)$ and is the
 516 same in every environment. More formally, for any measurable subset A of the range of Y and
 517 for any g_c , $P_e(Y \in A | G_c = g_c) = \mathbf{1}f(g_c) \in A$, which is evidently independent of e . Hence
 518 $P_e(Y | G_c) = P(Y | G_c)$ for all e . This captures the essence of invariant causal prediction, wherein
 519 the correct causal features G_c yield a predictor that holds across domains .

520 Now, if we remove any component of G_c , the remaining features would be an incomplete causal
 521 subgraph, insufficient to fully determine Y . In that case, the conditional $P_e(Y | \tilde{G}_c)$ (with $\tilde{G}_c \subsetneq G_c$)
 522 would generally depend on e because the relationship between \tilde{G}_c and Y could be confounded by
 523 the part of G_c that is missing. Similarly, if we include any non-causal features from G_s to form an
 524 augmented subgraph $G_c \cup G_s$, then $P_e(Y | G_c \cup G_s)$ may vary with e because G_s can carry spurious
 525 correlations with Y that differ by environment. By Assumption 2 the correlation between G_s and
 526 Y is not stable: there exists at least two environments e, e' for which $P_e(Y | G_s) \neq P_{e'}(Y | G_s)$
 527 (since G_s has no direct causal link to Y , any association is incidental and can change). Therefore,
 528 $P_e(Y | G_c, G_s)$ would generally differ from $P_{e'}(Y | G_c, G_s)$ because conditioning on G_s can
 529 introduce environment-specific information. We conclude that only the true causal subgraph G_c (or
 530 any superset that does not include spurious features) yields an invariant conditional for Y . \square

531 C.2 Proof of Lemma 2

532 *Proof.* This is a standard result from domain adaptation theory [2]. We treat each environment as
 533 a domain with distribution $P_e(Z, Y)$. The $\mathcal{H}\Delta\mathcal{H}$ -divergence between $P_e(Z)$ and $P_{e'}(Z)$ measures
 534 how well a classifier can distinguish between source and target representations; it can be seen as
 535 twice the supremum difference in probabilities assigned to sets by the two distributions (related to
 536 total variation distance restricted to hypothesis class \mathcal{H}). The cited bound (with $d(D_e, D_{e'})$ denoting
 537 this divergence) shows how much the source error can fail to transfer to target. For completeness:
 538 one derivation is

$$R_{e'}(h) - R_e(h) \leq \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(P_e(Z), P_{e'}(Z)) + \lambda^*, \quad (13)$$

539 and similarly $R_e(h) - R_{e'}(h)$ is bounded by the same quantity, yielding the two-sided inequality
 540 mentioned in different form . The term λ^* represents the best possible joint error; if the labeling rule
 541 is identical across domains, there exists a hypothesis (namely the Bayes-optimal classifier on that
 542 rule) that achieves low error on both, so λ^* would be small (zero in the ideal case where Bayes error
 543 is zero for that representation). Under our Assumption 1 the same causal labeling function $f(G_c)$
 544 applies in all environments, so for $Z = G_c$ one can achieve $\lambda^* = 0$ by choosing $h = f$. Thus, for the
 545 causal subgraph representation,

$$R_{e'}(h^*) \leq R_e(h^*) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(P_e(G_c), P_{e'}(G_c)), \quad (14)$$

with h^* being the invariant optimal classifier. This quantifies that any degradation in accuracy is due solely to the shift in G_c 's distribution across e and e' . This formal result confirms: an invariant representation (causal G_c) minimizes the transferable risk penalty to just the distribution divergence term, whereas a spurious representation incurs an additional irreducible error jump term. \square

550 C.3 Proof of Lemma 3

Proof. This statement reflects a basic requirement for generalization under distribution shift. If $\text{Supp}(P_{e'}(Z)) \subseteq \text{Supp}(P_e(Z))$, the classifier $h_\phi(\cdot)$ trained on P_e is at least receiving familiar inputs under $P_{e'}$. In the ideal case, if $h_\phi(\cdot)$ has learned the correct decision rule on $P_e(Z)$ (e.g. the true $f(\cdot)$ for G_c) and the rule remains the same (invariant labeling), it will apply equally to $P_{e'}(Z)$ as long as those inputs are not qualitatively new. Formally, for any z in the target support, since z is also in source support, $h_\phi(\cdot)$ had the opportunity to adjust its decision (or an equivalent z) during training; thus (\cdot) 's prediction at z can be expected to be as reliable as in training. If the supports overlap heavily but not completely, we can expect performance to degrade gracefully in proportion to how much probability mass falls in the unfamiliar regions.

On the other hand, if $\text{Supp}(P_{e'}(Z))$ extends to regions where $P_e(Z)$ has zero (or very low) density, then those z values are effectively never seen during training. A classifier cannot be expected to extrapolate correctly to arbitrarily novel inputs without additional knowledge; in the worst case, an adversarially chosen out-of-support input could be assigned an incorrect label by (\cdot) since (\cdot) has no basis to learn the correct behavior there. In domain adaptation terms, when support does not overlap, the $\mathcal{H}\Delta\mathcal{H}$ -divergence reaches its maximum (because a hypothesis can perfectly separate source and target supports), yielding a trivial bound $R_{e'}(h) \leq R_e(h) + 1/2 \cdot 2 + \lambda^* = R_e(h) + 1 + \lambda^*$, which means essentially no guarantee of generalization. In summary, overlapping support is a necessary condition for successful transfer; without it, the new domain may contain feature patterns fundamentally outside the model's experience, leading to unpredictable performance. \square

570 C.4 Proof of Theorem 1

Proof. We fix two arbitrary environments e (source) and e' (target) and compare the cross-environment divergence $\Delta(Z)$ for different choices of the subgraph Z . There are three typical cases to consider for an alternative subgraph G' that is not equal to G_c :

Case 1: G' includes non-causal parts (G_s). In this case, G' can be viewed as $G' = G_c \cup U$ where $U \subseteq G_s$ is some subset of spurious features (or possibly all of G_s , including the trivial case $G' = G$). Because G_s by definition contains the features that are not causally relevant to Y , any correlation between U and Y is spurious or environment-specific. By Assumption 2, environmental changes affect G_s significantly; thus the marginal distribution of U (and its correlation with G_c or Y) varies across environments. This implies that the joint distribution $P_e(G_c, U)$ differs from $P_{e'}(G_c, U)$ to a greater extent than $P_e(G_c)$ differs from $P_{e'}(G_c)$. Intuitively, since G_c is relatively stable but U is highly variable across e and e' , including U will amplify the cross-environment disparity. Formally, most divergence measures are monotonic under the introduction of additional differing variables; for example, if $P_e(G_c) = P_{e'}(G_c)$ but $P_e(U) \neq P_{e'}(U)$, then the joint divergence satisfies $d(P_e(G_c, U), P_{e'}(G_c, U)) \geq d(P_e(U), P_{e'}(U)) > 0$. Even if $P_e(G_c)$ changes slightly across environments, the changes in U (spurious part) are strictly larger (by Assumption 2), so $\Delta(G_c \cup U)$ will still exceed $\Delta(G_c)$.

In particular, consider the $\mathcal{H}\Delta\mathcal{H}$ -divergence as the measure d . If $Z = G_c \cup U$ contains environment-varying spurious components, one can construct a hypothesis in $\mathcal{H}\Delta\mathcal{H}$ distance that focuses on U to effectively distinguish which environment a sample came from. For instance, a classifier $h \in \mathcal{H}$ that predicts the environment identity from U will achieve better-than-chance accuracy due to U 's distribution shift, implying a large $d_{\mathcal{H}\Delta\mathcal{H}}(P_e(Z), P_{e'}(Z))$. In contrast, if $Z = G_c$ (with all G_s removed), then no classifier can reliably distinguish e vs e' because G_c by itself varies minimally – in the ideal case, $P_e(G_c) = P_{e'}(G_c)$ if the causal features are entirely invariant. Thus $d_{\mathcal{H}\Delta\mathcal{H}}(P_e(G_c), P_{e'}(G_c))$ will be small (in fact zero if G_c 's distribution is truly identical across e, e').

This reasoning formalizes that

$$\Delta(G_c \cup U) = d(P_e(G_c, U), P_{e'}(G_c, U)) > d(P_e(G_c), P_{e'}(G_c)) = \Delta(G_c).$$

Hence any inclusion of non-causal features U increases the divergence across environments.

Case 2: G' excludes part of the causal subgraph. In this scenario, G' is a strict subset of G_c (or possibly disjoint, but a disjoint subgraph would be pure G_s which is covered by Case 1). Let $G_c = G' \cup C_{\text{miss}}$, where C_{miss} is the portion of the true cause that is left out of G' . Because G' is missing some of the true causal features, it no longer fully determines the label Y . In fact, by Lemma 1 Y is not conditionally independent of the environment given G' – since G' omits part of G_c , the remaining features alone cannot guarantee the invariant relationship with Y . Equivalently, the effective labeling function on G' (i.e. the relationship between G' and Y) varies with the environment. In one environment, Y may depend on G' in one way, whereas in another environment the relationship shifts due to the influence of the missing causal factors C_{miss} . This means there is no single classifier on G' that perfectly fits $P(Y|G')$ across both e and e' – some environment-specific discrepancy in prediction is unavoidable.

Even if the marginal distributions of G' happen to be similar across environments (for instance, if $P_e(G_c)$ itself is invariant or if the environment does not directly alter the observed part G'), the fact that $Y|G'$ differs implies a significant distribution shift in the joint distribution of features and labels. To see this, consider the joint divergence (e.g. total variation or KL) between $P_e(G', Y)$ and $P_{e'}(G', Y)$. We can decompose it as differences in the conditional label distributions: if there exists any z' in the support of G' for which $P_e(Y|G' = z') \neq P_{e'}(Y|G' = z')$, the joint distributions will differ. In quantitative terms, one can lower-bound, for example, the total variation distance by the average conditional difference:

$$\text{TV}(P_e(G', Y), P_{e'}(G', Y)) \geq \frac{1}{2} \int_{z'} |P_e(Y | G' = z') - P_{e'}(Y | G' = z')| P_e(dz'). \quad (15)$$

$\text{TV}(P, Q)$ is the total-variation distance between two probability measures P and Q . By Lemma 1 such a difference is nonzero for G' that excludes part of G_c (there is at least some z' for which the label distributions diverge across environments). Therefore, the joint distribution shift is positive. In contrast, for $Z = G_c$, we have $P_e(Y|G_c) = P_{e'}(Y|G_c)$ exactly (labeling function is invariant), so no such difference occurs and the joint distributions $P_e(G_c, Y)$ and $P_{e'}(G_c, Y)$ align on the conditional label component (any remaining shift comes only from $P(G_c)$ differences, which are small by Assumption 2).

From a domain adaptation viewpoint, the omitted causal features lead to an intrinsic labeling mismatch across domains. In the bound of Lemma 2 this manifests as a nonzero λ^* term for $Z = G'$. In fact, λ^* in inequality 3 represents the minimum combined error on both environments; if no single classifier can simultaneously achieve low error on both e and e' because the label mappings differ, then λ^* is bounded away from 0. This contributes to an effective increase in distribution shift beyond what the feature divergence alone ($d_{\mathcal{H}\Delta\mathcal{H}}$) captures. Meanwhile, for $Z = G_c$, Lemma 1 guarantees the labeling function is identical in e and e' (so $\lambda^* = 0$), and we are left only with the feature divergence term. Thus, even if $d(P_e(G'), P_{e'}(G'))$ were as low as $d(P_e(G_c), P_{e'}(G_c))$ on the surface, the true shift relevant to classification is larger for G' due to the label-distribution change. In summary, excluding part of G_c makes the cross-environment difference strictly worse in terms of maintaining a stable predictor.

Case 3: G' both includes G_s and misses part of G_c . In this scenario G' contains some spurious components and is also missing some causal components. By the arguments above, such a G' will suffer from both a larger marginal distribution shift (due to the spurious parts varying across e, e') and a label conditional shift (due to incomplete causal information), each of which increases the divergence between $P_e(G')$ and $P_{e'}(G')$. Therefore, this case trivially yields $\Delta(G') > \Delta(G_c)$ as well.

Combining the cases, we conclude that any alternative subgraph G' that is not the full causal subgraph incurs a strictly greater distribution discrepancy between environments than G_c does. The causal subgraph G_c uniquely achieves the minimal cross-environment divergence by exactly capturing the invariant factors and nothing extra.

Moreover, by focusing on G_c , the learning algorithm sees an input distribution that is as invariant as possible across environments (Assumption 2 ensures minimal shift in G_c), and the label-generating

mechanism is completely stable (Assumption 1 ensures Y depends only on G_c in all environments). Consequently, both the feature distribution shift and the label conditional shift are minimized. Any deviation from G_c either introduces additional feature shift (by including G_s) or label shift (by losing part of G_c), hence increasing the overall divergence. In formal terms, for any divergence measure d , $d(P_e(G_c), P_{e'}(G_c)) < d(P_e(G'), P_{e'}(G'))$ for all $G' \neq G_c$. This proves the claim $\Delta(G_c) < \Delta(G')$.

Finally, note that by Lemma 3 using G_c also ensures the support of the target distribution is covered by the source distribution (no out-of-support surprise in new environments), which means the classifier can confidently generalize without encountering completely novel feature combinations. In contrast, a subgraph G' containing G_s might lead to out-of-support samples in a new environment (since G_s can take unprecedented values), which is another manifestation of increased distribution shift and would break the classifier as for Lemma 3.

In conclusion, extracting the true causal subgraph G_c yields the most invariant representation across environments, while minimizing reasonable measurements of distribution shift. Any other choice G' either violates the invariant label relationship or introduces extra environment-dependent variation, thereby increasing the cross-environment divergence. This completes the proof that G_c uniquely minimizes distributional disparity across environments of Theorem 1.

□

C.5 Proof of Theorem 2

Proof. Under Assumption 3 we require that for robust performance, the test inputs should not be completely novel relative to training. We argue that focusing on G_c satisfies this requirement across environments, whereas including G_s may violate it. Because G_c is tightly related to Y , **all environments that share the same task (same Y definition) are likely to exhibit G_c patterns that are necessary to produce Y .** Even if the marginal distribution $P_e(G_c)$ shifts a bit (e.g., some G_c patterns become more or less frequent), the set of possible G_c values remains linked to the support of Y . Unless the new environment introduces an entirely new causal factor (which would effectively change the task definition and violate Assumption 1), G_c in the new environment should fall within the realm of possibilities seen in training (perhaps with different probabilities). For example, if G_c is a subgraph motif that causally triggers a certain label, any environment where that label can occur will contain that motif in those instances; it would not spontaneously create a completely different unseen motif to cause the same label, since Y still comes from $f(G_c)$. This intuitive argument is backed by the idea that the causal mechanism $f(\cdot)$ is invariant – one cannot get a new output Y without the appropriate G_c input, so new environments cannot generate different valid G_c 's for the same Y (they could only omit some or add irrelevant decoration via G_s). Therefore, we expect $\text{Supp}(P_{e'}(G_c)) \subseteq \text{Supp}(P_{\text{train}}(G_c))$ (or at least a strong overlap), for any environment e' that does not fundamentally alter the nature of the task. This fulfills the support overlap condition of Lemma 3 for $Z = G_c$. By that lemma, the classifier $h_\phi(\cdot)$ (which without loss of generality we take as the optimal invariant predictor $f(\cdot)$ or an approximation thereof) will perform equally well in environment e' as it did in training, because it is operating on familiar ground. The risk in e' can thus remain as low as the risk in training, i.e. performance is stable.

Conversely, if one uses a subgraph G' that includes spurious elements, the new environment might present combinations of G' that were never seen before. For instance, perhaps in training, a certain spurious pattern in G' always coincided with a certain label (making the classifier think it was a useful feature), but in a new environment that pattern might appear with a different label or in a new context. The classifier, having learned a correlation, will mispredict because this input lies outside the training support for the joint (G', Y) distribution (the model never learned the correct response to that scenario). In formal terms, $P_{e'}(G')$ may put mass on regions of the G' space where $P_{\text{train}}(G')$ had nearly zero mass (for example, motif with unseen basis before). Thus $\text{Supp}(P_{e'}(G')) \not\subseteq \text{Supp}(P_{\text{train}}(G'))$. The violation of support overlap triggers exactly the failure mode highlighted in Lemma 3: the classifier $h_\phi(\cdot)$ is asked to extrapolate. If $h_\phi(\cdot)$ is a complex model (e.g. deep network), it might still output something for those novel inputs, but there is no guarantee it aligns with the true label – in fact it often will not, as it relies on the wrong features. This leads to performance drops or even arbitrarily bad predictions in the new environment.

Thus, using the causal subgraph G_c ensures that the classifier is always seeing data within (or very near) the domain it was trained on (since what changes across e is mostly the frequency of G_c features,

not the support itself), guaranteeing stable performance. In contrast, using a non-causal subgraph means the classifier is likely to eventually step out of distribution, suffering from the classic OOD generalization failure. We conclude the proof for Theorem 2. \square

D Additional Discussion of Empirical Examples on Distribution Shift and Representation Norms

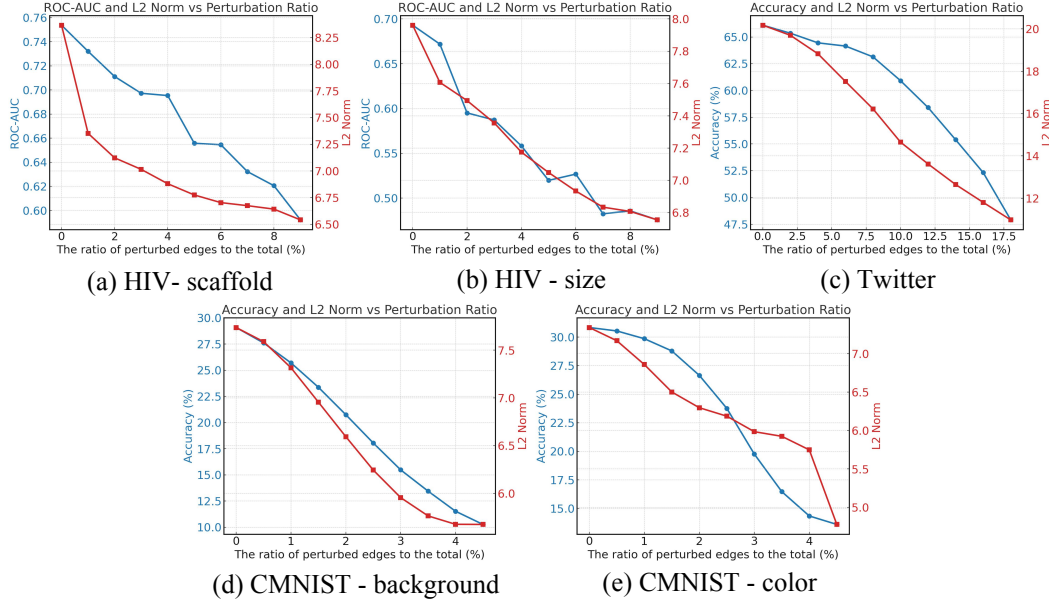


Figure 8: Structure causal models for graph data generation.

To illustrate that the representation norm decreases with increasing input distribution shift, we plot in Figure 2(b) how the norm varies when we perturb the model’s inputs. Figure 8 shows additional examples on other datasets. Specifically, following the experimental design described in the paper [14], we first train a high-accuracy GNN on the single-environment (no shift) dataset and freeze its parameters once converged. We then perturb the input graphs to simulate distribution shifts. Concretely, to model varying degrees of structural shift, we randomly insert a given proportion of edges into each input graph while simultaneously removing the same number of edges (thus preserving the total counts of nodes and edges so as to minimize any effects on the GNN’s message-passing and aggregation). As the figures demonstrate, the GNN is highly sensitive to structural shifts: as the shift magnitude grows, the overlap between the perturbed inputs and the low-dimensional weight subspace diminishes, causing the representation norm to fall and the model’s prediction accuracy to decline. These results show that the norm can be a robust indicator of distribution-shift severity.

E Low-Rankness in Graph Neural Networks

In neural networks, *low-rank* usually refers to the case where a layer’s weight matrix can be approximated by a matrix with lower rank. This property is widely used in model compression, fast inference, and generalization analysis. A common method to evaluate low-rank is singular value decomposition (SVD). If most singular values are close to zero and only a few are large, the matrix is considered approximately low-rank.

We examine the *low-rank* of graph neural networks using two common models: Graph Convolutional Network (GCN) and Graph Isomorphism Network (GIN). Experiments are conducted on the synthetic dataset GOODMotif and the real-world dataset GOODSST2. Both models use three layers, and each GIN layer includes a two-layer MLP for feature transformation. As shown in the Figure 3 and 9, the weight matrices in each convolutional layer show clear low-rank patterns for both GCN and GIN.

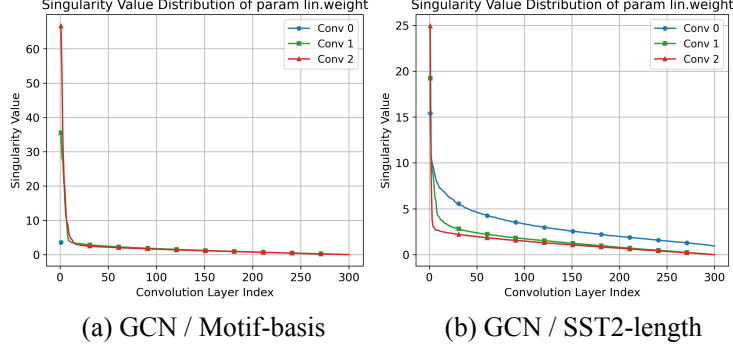


Figure 9: Singular value decomposition (SVD) results of GCN weights. Both models are trained with the empirical risk minimization (ERM) objective. Singular values are sorted in descending order. Clear low-rank patterns are observed across layers.

720 F A Toy Example of Low-Rank and Norm

721 Consider the weight matrix $W \in \mathbb{R}^{4 \times 3}$ given by:

$$W = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},$$

722 which has rank $\text{rank}(W) = 2$. Now take two input vectors of unit length in different directions:

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

723 We compute their images under W , regardless of the bias:

$$W \mathbf{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad W \mathbf{x}_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

724 Consequently, their Euclidean norms satisfy

$$\|W \mathbf{x}_1\|_2 = \sqrt{1^2 + 0^2 + 1^2 + 0^2} = \sqrt{2} > \|W \mathbf{x}_2\|_2 = 0.$$

725 This simple example illustrates that a low-rank weight matrix can produce substantially different
726 output norms for inputs aligned with its row-space versus those not aligned.

727 G Related Work

728 Out-of-distribution (OOD) generalization is a critical challenge in graph machine learning, as models
729 trained on a given data distribution often fail to perform well on unseen distributions. Invariant learn-
730 ing, grounded in causal theory [25], is a primary approach to this problem: it seeks to learn causally
731 relevant representations that remain stable across different environments. To acquire environment
732 information, some methods leverage dataset-provided environment labels, e.g., IRM [1] and LECI [8],
733 while others predict environment labels via unsupervised clustering, as in MoleOOD [35], GIL [19],
734 and OOD-GCL [18], which entails prior assumptions about the environment distribution. Approaches
735 such as DIR [33], GREa [22] and iMoLD [40] identify invariant features through structure- or feature-
736 level disentanglement and recombination; CIGA [4], EQuAD [36], and LIRS [38] use self-supervised
737 learning to separate invariant from spurious features.

738 Beyond invariant learning, alternative strategies have been developed to enhance generalization.
739 DANN [6] applies domain-generalization techniques to tackle OOD issues; GSAT [24] and
740 GOODGAT [32] exploit the graph information bottleneck to discover causal subgraphs; G-Splice [20]
741 uses linear extrapolation to broaden dataset distributions; DGAT [9] leverages GAT [31]’s attention
742 mechanism to strengthen GNN generalization; DIVE [29] makes predictions and summaries by
743 selecting different and non-overlapping subgraphs from a single input graph respectively.

H Datasets

We adopt two widely used benchmarks for graph OOD generalization—GraphOOD [7] and DrugOOD [13]—which together cover synthetic graphs, superpixel graphs, molecular graphs, and textual graphs:

- **GraphOOD**: a systematic benchmark tailored to graph OOD problems. We draw on four dataset groups of covariate shift in GraphOOD for graph classification: (1) **GOOD-Motif**: a synthetic dataset with two domain types—base-graph structure and graph size. (2) **GOOD-CMNIST**: a multi-class, semi-synthetic dataset obtained by converting Colored MNIST [1] into superpixel graphs, with different digit-color as domains. (3) **GOOD-HIV**: a real-world binary classification task predicting whether a molecule inhibits HIV replication, with scaffold and size as domains. (4) **GOOD-SST2** and **GOOD-Twitter**: sentiment-analysis tasks (binary and ternary, respectively) derived by encoding sentences as syntax trees, using sequence length as the domain.

- **DrugOOD**, an molecule OOD benchmark for drug discovery, defines three domain splits—assay, scaffold, and size—applied to two binding-affinity measurements (IC50 and EC50). This yields six binary-classification datasets, each predicting drug–target binding affinity.

As in prior work, we partition each dataset by its domain attribute to induce distribution shifts. For example, in the Motif basis-shift setting, the motif types in the test set are entirely disjoint from those in the training and validation sets, thus rigorously assessing model generalization.

We use the ROC-AUC metric for the binary classification dataset and Accuracy for the others. More details on the datasets can be found in the original papers [7, 13].

I Baselines Details

We adopt the following methods as baselines for comparison:

General methods:

- **ERM** minimizes the empirical loss on the training set.

- **IRM** [1] seeks to find data representations across all environments by penalizing feature distributions that have different optimal classifiers.

- **Coral** [28] encourages feature distributions consistent by penalizing differences in the means and covariances of feature distributions for each domain.

- **VREx** [17] reduces the risk variances of training environments to achieve both covariate robustness and invariant prediction.

Graph-specific OOD methods:

- **DIR** [33] discovers the subset of a graph as invariant rationale by conducting interventional data augmentation to create multiple distributions.

- **GIL** [19] employs unsupervised clustering to infer environmental labels and leverages the invariant principle to identify causal subgraphs.

- **GSAT** [24] proposes to build an interpretable graph learning method through the attention mechanism and inject stochasticity into the attention to select label-relevant subgraphs.

- **CIGA** [4] proposes an information-theoretic objective to extract the desired invariant subgraphs from the lens of causality.

- **LECI** [8] assume the availability of environment labels, and study environment exploitation strategies for graph OOD generalization.

- **iMoLD** [40] employ environment augmentation techniques to facilitate the learning of invariant graph-level representations.

- **EQuAD** [36] adopts self-supervised learning to learn spurious features first, followed by learning invariant features by unlearning spurious features.

789 • **LIRS** [38] takes an indirect approach by first learning the spurious features and then removing them
790 from the ERM-learned features.

791 Our selected baselines encompass a diverse array of approaches for tackling graph out-of-distribution
792 (OOD) problems, including state-of-the-art and recently proposed methods. Some approaches such
793 as OOD-GCL [18], GOODGAT [32], G-Splice [20] DGAT [9], et al. are omitted owing to a lack
794 of comparable performance results or available implementation details. Moreover, the baselines we
795 selected already encompass the main research directions of most state-of-the-art graph OOD methods.

796 J Visualized Cases

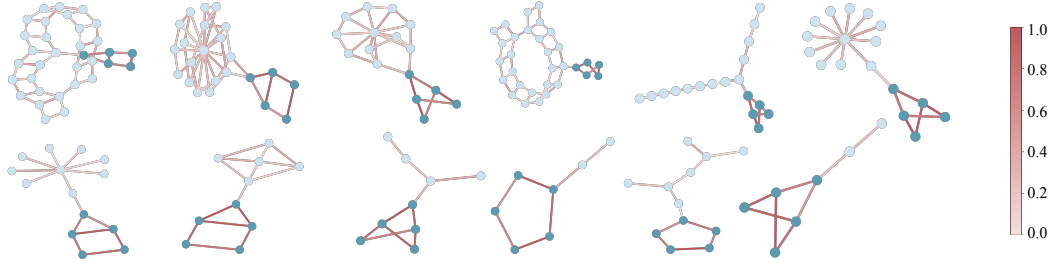


Figure 10: Visualized cases of the Motif-OOD dataset with size and basis domain shift. Nodes with dark blue and light blue colors represent the motif nodes and base graph nodes, respectively. The shading of the edges indicates the importance score of each edge generated by the subgraph extractor.

797 We present several visualized cases with size and basis domain shift in Figure 10. These examples
798 demonstrate that our method can effectively extract the causal subgraph (Motif) from the input graph
799 rather than selecting spurious factors (basis graphs). This also highlights the inherent interpretability
800 of our approach.

801 K Discussion on the Effect of Top Ratio Hyperparameter

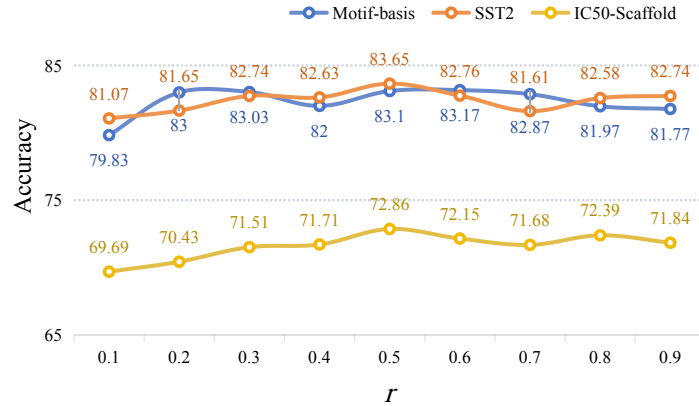


Figure 11: Performance on three datasets with different r .

802 In our approach, the hyperparameter r controls the fraction of edges designated as the causal subgraph.
803 However, the real edge ratio of subgraph in real-world datasets is typically unknown. As shown in
804 Figure 11 we evaluate model accuracy across three datasets for different r values and find that r
805 has no significant impact on generalization performance. From these results, we draw two empirical
806 conclusions:

Avoid overly small r : If r is set too low, the selected subgraph fails to fully cover the causal structure, degrading performance.

Robustness for a moderate r : When r exceeds a minimal threshold, its exact value has little effect. We find that this robustness stems from the entropy-regularization term \mathcal{L}_{comp} in [10] which automatically enforces an appropriate level of mask sparsity: some selected edges acquire near-zero mask weights and are thus effectively omitted during prediction.

Accordingly, we recommend $r = 0.5$ as a reasonable default for training.

L Efficiency Study

Experiments in this paper are conducted on NVIDIA RTX3090 GPUs. Our method is concise and streamlined: norm computation introduces virtually no additional overhead, and the two-stage parameter updates have a negligible impact on efficiency. Moreover, the Table 4 and 5 reports training and inference times of IDG and baselines, underscoring the high efficiency of our approach.

Dataset	Training Batch Size	Testing Batch Size	Training Time (s)	Inference Time (s)
ERM	64	256	1032	1.3
LECI	64	256	3404	1.6
DIR	64	256	3213	1.7
IDG	64	256	1603	1.6

Table 4: Efficiency study of our method on Motif-basis.

Dataset	Training Batch Size	Testing Batch Size	Training Time (s)	Inference Time (s)
ERM	64	256	1148	0.51
LECI	64	256	3973	1.43
DIR	64	256	3472	1.62
IDG	64	256	1855	1.53

Table 5: Efficiency study of our method on SST2.

M Broader Impact

Our work aims to enhance the generalization of graph neural networks (GNNs) in out-of-distribution (OOD) scenarios, which is crucial for real-world applications. By focusing on causal subgraph extraction, we provide a method that can potentially improve the robustness and reliability of GNNs in various domains, including drug discovery, social network analysis, and biological data interpretation. However, it is important to acknowledge that our approach may not be universally applicable to all graph-based tasks or datasets. The evaluations of our approach are mainly across limited graph domains, which may not represent all possible real-world scenarios. The approach can be evaluated on various environmental domains to be validated in a more realistic setting.

N Limitations

As with other graph generalization methods, although our approach improves the model’s out-of-distribution performance to some extent, its transferability to other domains remains uncertain. Moreover, for datasets without any ground truth, leveraging the extracted subgraphs to further enhance generalization is a direction that has yet to be fully explored.