Causal-aware Graph Neural Architecture Search under Distribution Shifts

Peiwen Li*
SIGS, Tsinghua University
Shenzhen, China
lpw22@mails.tsinghua.edu.cn

Xin Wang†
DCST, BNRist, Tsinghua University
Beijing, China
xin_wang@tsinghua.edu.cn

Zeyang Zhang
DCST, Tsinghua University
Beijing, China
zy-zhang20@mails.tsinghua.edu.cn

Ziwei Zhang
DCST, Tsinghua University
Beijing, China
zw-zhang16@tsinghua.org.cn

Fang Shen
Alibaba Cloud
Hangzhou, China
ziru.sf@alibaba-inc.com

Jialong Wang
Alibaba Cloud
Hangzhou, China
quming.wjl@alibaba-inc.com

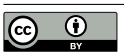
Yang Li
SIGS, Tsinghua University
Shenzhen, China
yangli@sz.tsinghua.edu.cn

Wenwu Zhu†
DCST, BNRist, Tsinghua University
Beijing, China
wwzhu@tsinghua.edu.cn

Abstract

Graph neural architecture search (NAS) has emerged as a promising approach for autonomously designing graph neural network architectures by leveraging correlations between graphs and architectures. However, existing methods merely rely on correlations, which may be *spurious* and vary across distributions. This reliance, without considering causal graph-architecture relationships, limits their ability to generalize under distribution shifts that are ubiquitous in real-world graph scenarios. In this paper, we propose to handle the distribution shifts in NAS process by exploiting the *causal* graph-architecture relationship to search for optimal architectures that can generalize under distribution shifts. Key challenges remain unexplored: discovering causal graph-architecture relationships with stable cross-distribution predictive abilities, and leveraging them to handle distribution shifts. To address these challenges, we propose a novel approach, Causal-aware Graph Neural Architecture Search (**CARNAS**), which is capable of capturing causal graph-architecture relationship during NAS process and discovering optimal graph architecture under distribution shifts. We propose Disentangled Causal Subgraph Identification to extract causal subgraphs with stable predictive power across distributions, followed by Graph Embedding Intervention to intervene on these subgraphs in latent space by preserving essential features while filtering out non-causal elements, and Invariant Architecture Customization to enhance their causal invariance for optimizing graph architectures. Extensive experiments on synthetic and real-world datasets show that CARNAS enhances out-of-distribution generalization by uncovering causal graph-architecture relationships during NAS.

*The work was done during author's internship at Alibaba Cloud.
†Corresponding authors.



KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3736873

CCS Concepts

• **Computing methodologies** → **Neural networks**.

Keywords

Graph Neural Architecture Search, Out-of-Distribution Generalization, Causal Learning

ACM Reference Format:
Peiwen Li, Xin Wang, Zeyang Zhang, Ziwei Zhang, Fang Shen, Jialong Wang, Yang Li, and Wenwu Zhu. 2025. Causal-aware Graph Neural Architecture Search under Distribution Shifts. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3736873

KDD Availability Link:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.15575595.

1 Introduction

Graph neural architecture search (Graph NAS), aiming at automating the designs of GNN architectures for different graphs, has shown great success by exploiting the correlations between graphs and architectures. Present approaches [8, 18, 28] leverage a rich search space filled with GNN operations and employ strategies like reinforcement learning and continuous optimization algorithms to pinpoint an optimal architecture for specific datasets, aiming to decode the natural correlations between graph data and their ideal architectures. Based on the independently and identically distributed (I.I.D) assumption on training and testing data, existing methods assume the graph-architecture correlations are stable across graph distributions.

Nevertheless, distribution shifts are ubiquitous and inevitable in real-world graph scenarios, particularly evident in applications existing with numerous unforeseen and uncontrollable hidden factors like drug discovery, in which the availability of training data is limited, and the complex chemical properties of different molecules lead to varied interaction mechanisms [15]. Consequently, GNN

models developed for such purposes must be generalizable enough to handle unavoidable variations in data distribution between training and testing sets, underlining the critical need for models that can adapt to and perform reliably under such varying conditions.

Existing Graph NAS methods primarily rely on correlations between graphs and architectures during the search process. These correlations, while effective in the training distribution, may be *spurious* and tend to vary with distribution shifts. Without specifically considering the intrinsic causal relationships between graph structures and architectures, the search process unintendedly captures these spurious patterns that are specific to training data. Consequently, although achieving good performance on the training distribution, the identified patterns fail to generalize when the underlying data distribution changes in the test set, resulting in significant performance degradation under distribution shifts.

In this paper, we study the problem of graph neural architecture search under distribution shifts by capturing the *causal* relationship between graphs and architectures to search for the optimal graph architectures that can generalize under distribution shifts. The problem is highly non-trivial with the following challenges:

- How to discover the causal graph-architecture relationship that has stable predictive abilities across distributions?
- How to handle distribution shifts with the discovered causal graph-architecture relationship to search the generalized graph architectures?

To address these challenges, we propose the Causal-aware Graph NAS (**CARNAS**), which is able to capture the causal relationship, stable to distribution shifts, between graphs and architectures, and thus handle the distribution shifts in the graph architecture search process. Specifically, we design a *Disentangled Causal Subgraph Identification* module, which employs disentangled GNN layers to obtain node and edge representations, then further derive causal subgraphs based on the importance of each edge. This module enhances the generalization by deeply exploring graph features as well as latent information with disentangled GNNs, thereby enabling a more precise extraction of causal subgraphs, carriers of causally relevant information, for each graph instance. Following this, our *Graph Embedding Intervention* module employs another shared GNN to encode the derived causal subgraphs and non-causal subgraphs in the same latent space, where we perform interventions on causal subgraphs with non-causal subgraphs. Additionally, we ensure the causal subgraphs involve principal features by engaging the supervised classification loss of causal subgraphs into the training objective. We further introduce the *Invariant Architecture Customization* module, which addresses distribution shifts not only by constructing architectures for each graph with their causal subgraph but also by integrating a regularizer on simulated architectures corresponding to those intervention graphs, aiming to reinforce the causal invariant nature of causal subgraphs derived in module 1. We remark that the classification loss for causal subgraphs in module 2 and the regularizer on architectures for intervention graphs in module 3 help with ensuring the causality between causal subgraphs and the customized architecture for a graph instance. Moreover, by incorporating them into the training and search process, we make the Graph NAS model intrinsically interpretable to some degree. Empirical validation across both synthetic and real-world datasets underscores the remarkable out-of-distribution generalization capabilities of CARNAS over existing baselines. Detailed ablation studies further verify our designs. The contributions of this paper are summarized as follows:

- We are the first to study graph neural architecture search under distribution shifts from the causal perspective, by proposing the causal-aware graph neural architecture search (CARNAS), that integrates causal inference into graph neural architecture search, to the best of our knowledge.
- We propose three modules: disentangled causal subgraph identification, graph embedding intervention, and invariant architecture customization, offering a nuanced strategy for extracting and utilizing causal graph-architecture relationships, which is stable under distribution shifts, thereby enhancing model's capability of out-of-distribution generalization.
- Extensive experiments on both synthetic and real-world datasets confirm that CARNAS significantly outperforms existing baselines, showcasing its efficacy in improving graph classification accuracy across diverse datasets, and validating the superior out-of-distribution generalization capabilities of our methods.

2 Preliminary

2.1 Graph NAS under distribution shifts

Denote $\mathbb{G}$ and $\mathbb{Y}$ as the graph and label space. We consider a training graph dataset $\mathcal{G}_{tr} = \{(G_i, Y_i)\}_{i=1}^{N_{tr}}$ and a testing graph dataset $\mathcal{G}_{te} = \{(G_i, Y_i)\}_{i=1}^{N_{te}}$, where $G_i \in \mathbb{G}$, $Y_i \in \mathbb{Y}$, N_{tr} and N_{te} represent the number of graph instances in training set and testing set, respectively. The generalization of graph classification under distribution shifts can be formed as:

Problem 1. *We aim to find the optimal prediction model $F^*(\cdot) : \mathbb{G} \to \mathbb{Y}$ that performs well on $\mathcal{G}_{te}$ when there is a distribution shift between training and testing data, i.e. $P(\mathcal{G}_{tr}) \neq P(\mathcal{G}_{te})$:*

$$F^*(\cdot) = \arg\min_{F} \mathbb{E}_{(G,Y)\sim P(\mathcal{G}_{te})} \left[\ell(F(G), Y) \mid \mathcal{G}_{tr}\right], \tag{1}$$

where $\ell(\cdot,\cdot) : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}$ is a loss function.

Graph NAS methods search the optimal GNN architecture A^* from the search space $\mathcal{A}$, and form the complete model F together with the learnable parameters ω. Unlike most existing works using a fixed GNN architecture for all graphs, [37] is the first to customize a GNN architecture for each graph, supposing that the architecture only depends on the graph. We follow the idea and inspect deeper concerning the graph neural architecture search process.

2.2 Causal view of the Graph NAS process

Causal approaches are largely adopted when dealing with out-of-distribution (OOD) generalization by capturing the stable causal structures or patterns in input data that influence the results [21]. While in normal graph neural network cases, previous work that studies the problem from a causal perspective mainly considers the causality between graph data and labels [23, 46].

Causal analysis in Graph NAS. Based on the known that different GNN architectures suit different graphs [5, 51] and inspired by [47], we analyze the potential relationships between graph instance G,

causal subgraph G_c, non-causal subgraph G_s and optimal architecture A^* for G in the graph neural architecture search process:

- $G_c \rightarrow G \leftarrow G_s$ indicates that two disjoint parts, causal subgraph G_c and non-causal subgraph G_s, together form the input graph G.
- $G_c \rightarrow A^*$ represents our assumption that there exists the causal subgraph which solely determines the optimal architecture A^* for input graph G. Taking the Spurious-Motif dataset [53] as an example, [37] discovers that different shapes of graph elements prefer different architectures.
- $G_c \dashleftarrow\dashrightarrow G_s$ means that there are potential probabilistic dependencies between G_c and G_s [33, 34], which can make up spurious correlations between the non-causal subgraph G_s and the optimal architecture A^*.

Intervention. Inspired by the ideology of invariant learning [1, 3, 17], that forms different environments to abstract the invariant features, we do interventions on causal subgraph G_c by adding different spurious (non-causal) subgraphs to it, and therefore simulate different environments for a graph instance G.

2.3 Problem formalization

Based on the above analysis, we propose to search for a causal-aware GNN architecture for each input graph. To be specific, we target to guide the search for the optimal architecture A^* by identifying the causal subgraph G_c in the Graph NAS process. Therefore, Problem 1 is transformed into the following concrete task as in Problem 2.

Problem 2. *We systematize model $F : \mathbb{G} \rightarrow \mathbb{Y}$ into three modules, i.e. $F = f_C \circ f_A \circ f_Y$, in which $f_C(G) = G_c : \mathbb{G} \rightarrow \mathbb{G}_c$ abstracts the causal subgraph G_c from input graph G, where causal subgraph space $\mathbb{G}_c$ is a subset of $\mathbb{G}$, $f_A(G_c) = A : \mathbb{G}_c \rightarrow \mathcal{A}$ customizes the GNN architecture A for causal subgraph G_c, and $f_Y(G, A) = \hat{Y} : \mathbb{G} \times \mathcal{A} \rightarrow \mathbb{Y}$ outputs the prediction $\hat{Y}$. Further, we derive the following objective function:*

$$\min_{f_C, f_A, f_Y} \sigma \mathcal{L}_{pred} + (1 - \sigma)\mathcal{L}_{causal}, \tag{2}$$

$$\mathcal{L}_{pred} = \sum_{i=1}^{N_{tr}} \ell \left(F_{f_C(G_i), f_A(G_{ci}), f_Y(G_i, A_i)}\left(G_i\right), Y_i \right), \tag{3}$$

where $\mathcal{L}_{pred}$ guarantees the final prediction performance of the whole model, $\mathcal{L}_{causal}$ is a regularizer for causal constraints and σ is the hyper-parameter to adjust the optimization of those two parts.

3 Method

We present our proposed method in this section based on the above causal view. Firstly, we present the disentangled causal subgraph identification module to obtain the causal subgraph for searching optimal architecture in Section 3.1. Then, we propose the intervention module in Section 3.2, to help with finding the invariant subgraph that is causally correlated with the optimal architectures, making the NAS model intrinsically interpretable to some degree. In Section 3.3, we introduce the simulated customization module which aims to deal with distribution shift by customizing for each graph and simulating the situation when the causal subgraph is affected by different spurious parts. Finally, we show the total invariant learning and optimization procedure in Section 3.4. To more rigorously establish our method, we provide a theoretical analysis in Appendix A about the problem of identifying and leveraging causal graph-architecture relationship to find the optimal architecture.

3.1 Disentangled causal subgraph identification

This module utilizes disentangled GNN layers to capture different latent factors of the graph structure and further split the input graph instance G into two subgraphs: causal subgraph G_c and non-causal subgraph G_s. Specifically, considering an input graph $G = (\mathcal{V}, \mathcal{E})$, its adjacency matrix is $\mathbf{D} \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{V}|}$, where $\mathbf{D}_{i,j} = 1$ denotes that there exists an edge between node V_i and node V_j, while $\mathbf{D}_{i,j} = 0$ otherwise. Since optimizing a discrete binary matrix $\mathbf{M} \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{V}|}$ is unpractical due to the enormous number of subgraph candidates [53], and learning $\mathbf{M}$ separately for each input graph fails in generalizing to unseen test graphs [30], we adopt shared learnable disentangled GNN layers to comprehensively unveil the latent graph structural features and better abstract causal subgraphs. Firstly, we denote Q as the number of latent features taken into account, and learn Q-chunk node representations by Q GNNs:

$$\mathbf{Z}^{(l)} = \|_{q=1}^{Q} \mathrm{GNN}_0 \left(\mathbf{Z}_q^{(l-1)}, \mathbf{D} \right), \tag{4}$$

where $\mathbf{Z}_q^l$ is the q-th chunk of the node representation at l-th layer, $\mathbf{D}$ is the adjacency matrix, and $\|$ denotes concatenation. Then, we generate the edge importance scores $\mathcal{S}_\mathcal{E} \in \mathbb{R}^{|\mathcal{E}| \times 1}$ with an MLP:

$$\mathcal{S}_\mathcal{E} = \mathrm{MLP} \left(\mathbf{Z}_{row}^{(L)}, \mathbf{Z}_{col}^{(L)} \right), \tag{5}$$

where $\mathbf{Z}^{(L)} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the node representations after L layers of disentangled GNN, and $\mathbf{Z}_{row}^{(L)}, \mathbf{Z}_{col}^{(L)}$ are the subsets of $\mathbf{Z}^{(L)}$ containing the representations of row nodes and column nodes of edges $\mathcal{E}$ respectively. After that, we attain the causal and non-causal subgraphs by picking out the important edges through $\mathcal{S}_\mathcal{E}$:

$$\mathcal{E}_c = \mathrm{Top}_t(\mathcal{S}_\mathcal{E}),\ \mathcal{E}_s = \mathcal{E} - \mathcal{E}_c, \tag{6}$$

where $\mathcal{E}_c$ and $\mathcal{E}_s$ denotes the edge sets of G_c and G_s, respectively, and $\mathrm{Top}_t(\cdot)$ selects the top t-percentage of edges with the largest edge score values.

3.2 Graph embedding intervention

After obtaining the causal subgraph G_c and non-causal subgraph G_s of an input graph G, we use another shared GNN_1 to encode those subgraphs so as to do interventions in the same latent space:

$$\mathbf{Z_c} = \mathrm{GNN}_1\left(G_c\right),\ \mathbf{Z_s} = \mathrm{GNN}_1\left(G_s\right). \tag{7}$$

Moreover, a readout layer is placed to aggregate node-level representations into graph-level representations:

$$\mathbf{H_c} = \mathrm{READOUT}\left(\mathbf{Z_c}\right),\ \mathbf{H_s} = \mathrm{READOUT}\left(\mathbf{Z_s}\right). \tag{8}$$

Supervised classification for causal subgraphs. We claim that the causal subgraph G_c inferred in Section 3.1 for finding the optimal GNN architecture is supposed to contain the main characteristic of graph G's structure as well as capture the essential part for the final

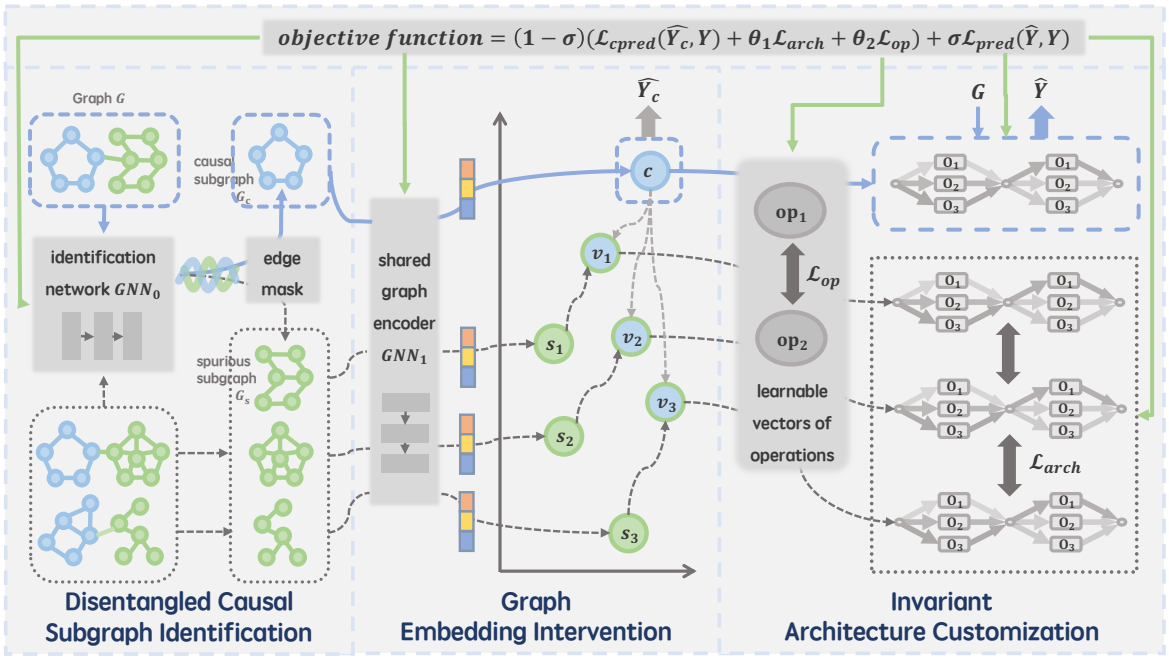


Figure 1: The framework of our proposed method CARNAS. As for an input graph G, the disentangled causal subgraph identification module abstracts its causal subgraph G_c with disentangled GNN layers. Then, in the graph embedding intervention module, we conduct several interventions on G_c with non-causal subgraphs in latent space and obtain $\mathcal{L}_{cpred}$ from the embedding of G_c in the meanwhile. After that, the invariant architecture customization module aims to deal with distribution shift by customizing architecture from G_c to attain $\hat{Y}$, $\mathcal{L}_{pred}$, and form $\mathcal{L}_{arch}$, $\mathcal{L}_{op}$ to further constrain the causal invariant property of G_c. Blue lines present the prediction approach and grey lines show other processes in the training stage. Additionally, green lines denote the updating process.

graph classification predicting task. Hence, we employ a classifier on $\mathbf{H_c}$ to construct a supervised classification loss:

$$\mathcal{L}_{cpred} = \sum_{i=1}^{N_{tr}} \ell\left(\hat{Y}_{c_i}, Y_i\right), \ \hat{Y}_{c_i} = \Phi\left(\mathbf{H_c}_i\right), \tag{9}$$

where Φ is a classifier, $\hat{Y}_{c_i}$ is the prediction of graph G_i's causal subgraph G_{c_i} and Y_i is the ground truth label of G_i.

Interventions by non-causal subgraphs. Based on subgraphs' embedding $\mathbf{H_c}$ and $\mathbf{H_s}$, we formulate the intervened embedding $\mathbf{H_v}$ in the latent space. Specifically, we collect all the representations of non-causal subgraphs $\{\mathbf{H_s}_i\}, i \in [1, N_{tr}]$, corresponding to each input graph $\{G_i\}, i \in [1, N_{tr}]$, in the current batch, and randomly sample N_s of them as the candidates $\{\mathbf{H_s}_j\}, j \in [1, N_s]$ to do intervention with. As for a causal subgraph G_c with representation $\mathbf{H_c}$, we define the representation under an intervention as:

$$do(S = G_{s\,j}) : \ \mathbf{H_v}_j = (1-\mu)\cdot\mathbf{H_c} + \mu\cdot\mathbf{H_s}_j, \ j \in [1, N_s], \tag{10}$$

in which $\mu \in (0, 1)$ is the hyper-parameter to control the intensity of an intervention.

3.3 Invariant architecture customization

After obtaining graph representations $\mathbf{H_c}$ and $\mathbf{H_v}_j, j \in [1, N_s]$, we introduce the method to construct a specific GNN architecture from a graph representation on the basis of differentiable NAS [28].

Architecture customization. To begin with, we denote the space of operator candidates as O and the number of architecture layers as K. Then, the ultimate architecture A can be represented as a super-network:

$$g^k(\mathbf{x}) = \sum_{u=1}^{|O|} \alpha_u^k o_u(\mathbf{x}), \ k \in [1, K], \tag{11}$$

where $\mathbf{x}$ is the input to layer k, $o_u(\cdot)$ is the operator from O, α_u^k is the mixture coefficient of operator $o_u(\cdot)$ in layer k, and $g^k(x)$ is the output of layer k. Thereat, an architecture A can be represented as a matrix $\mathbf{A} \in \mathbb{R}^{K\times|O|}$, in which $\mathbf{A}_{k,u} = \alpha_u^k$. We learn these coefficients from graph representation $\mathbf{H}$ via trainable prototype vectors $\mathbf{op}_u^k$ ($u \in [1, |O|], k \in [1, K]$), of operators:

$$\alpha_u^k = \frac{\exp\left(\mathbf{op}_u^{k\,T}\mathbf{H}\right)}{\sum_{u'=1}^{|O|}\exp\left(\mathbf{op}_{u'}^{k\,T}\mathbf{H}\right)}. \tag{12}$$

In addition, the regularizer for operator prototype vectors:

$$\mathcal{L}_{op} = \sum_k \sum_{u,u'\in[1,|O|],u\neq u'} \cos(\mathbf{op}_u^k, \mathbf{op}_{u'}^k), \tag{13}$$

where $\cos(\cdot,\cdot)$ is the cosine distance between two vectors, is engaged to avoid the mode collapse, following the exploration in [37].

Architectures from causal subgraph and intervention graphs. So far we form the mapping of $f_A : \mathbb{G} \to \mathcal{A}$ in Problem 2. As for an input graph G, we get its optimal architecture A_c with the matrix $\mathbf{A_c}$ based on its causal subgraph's representation $\mathbf{H_c}$ through equation (12), while for each intervention graph we have $\mathbf{A_v}_j$ based on $\mathbf{H_v}_j$, $j \in [1, N_s]$ similarly.

The customized architecture A_c is used to produce the ultimate prediction of input graph G by $f_Y : \mathbb{G} \times \mathcal{A} \to \mathbb{Y}$ in Problem 2, and we formulate the main classification loss as:

$$\mathcal{L}_{pred} = \sum_{i=1}^{N_{tr}} \ell(\hat{Y}_i, Y_i), \ \hat{Y}_i = f_Y(G_i, A_{ci}). \tag{14}$$

Furthermore, we regard each $\mathbf{A_v}_j$, $j \in [1, N_s]$ as an outcome when causal subgraph G_c is in a specific environment (treating the intervened part, i.e. non-causal subgraphs, as different environments). Therefore, the following variance regularizer is proposed as a causal constraint to compel the inferred causal subgraph G_c to have the steady ability to solely determine the optimal architecture for input graph instance G:

$$\mathcal{L}_{arch} = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \mathbf{1}^T \cdot \mathbf{Var}_i \cdot \mathbf{1}, \mathbf{Var}_i = \text{var}\left(\{\mathbf{A_v}_{ij}\}\right), \ j \in [1, N_s], \tag{15}$$

where $\text{var}(\cdot)$ calculates the variance of a set of matrix, $\mathbf{1}^T \cdot \mathbf{Var}_i \cdot \mathbf{1}$ represents the summation of elements in matrix $\mathbf{Var}_i$.

3.4 Optimization framework

Up to now, we have introduced $f_C : \mathbb{G} \to \mathbb{G}_c$ in section 3.1, $f_A : \mathbb{G}_c \to \mathcal{A}$ in section 3.2 and 3.3, $f_Y : \mathbb{G} \times \mathcal{A} \to \mathbb{Y}$ in section 3.3, and whereby deal with Problem 2. To be specific, the overall objective function in equation (2) is as below:

$$\begin{aligned} \mathcal{L}_{all} &= \sigma \mathcal{L}_{pred} + (1 - \sigma)\mathcal{L}_{causal}, \\ \mathcal{L}_{causal} &= \mathcal{L}_{cpred} + \theta_1 \mathcal{L}_{arch} + \theta_2 \mathcal{L}_{op}, \end{aligned} \tag{16}$$

where θ_1, θ_2 and σ are hyper-parameters. Additionally, we adopt a linearly growing σ_p corresponding to the epoch number p as:

$$\sigma_p = \sigma_{min} + (p - 1)\frac{\sigma_{max} - \sigma_{min}}{P}, \ p \in [1, P], \tag{17}$$

where P is the maximum number of epochs. In this way, we can dynamically adjust the training key point in each epoch by focusing more on the causal-aware part (i.e. identifying suitable causal subgraph and learning vectors of operators) in the early stages and focusing more on the performance of the customized super-network in the later stages. We show the dynamic training process and how σ_p improve the training and convergence efficiency in Section 5.3. The overall framework and optimization procedure of the proposed CARNAS are summarized in Figure 1 and Algorithm 1.

4 Experiments

In this section, we present the comprehensive results of our experiments on both synthetic and real-world datasets to validate the effectiveness of our approach. We also conduct a series of ablation studies to thoroughly examine the contribution of the components within our framework. At the end of this section, we analyze both time and parameter complexity.

Algorithm 1 The overall algorithm of **CARNAS**

Require: Training Dataset $\mathcal{G}_{tr}$,
Hyper-parameters t in Eq. (6), μ in Eq. (10), θ_1, θ_2 in Eq. (16)

```
Initialize all trainable parameters
for p = 1, ..., P do
    Set σ_p as Eq. (17)
    Derive causal and non-causal subgraphs as Eq. (4) (5) (6)
    Calculate graph representations of causal and non-causal subgraphs as Eq. (7) (8)
    Calculate L_cpred using Eq. (9)
    Sample N_s non-causal subgraphs as candidates
    for causal subgraph G_c of graph G in G_tr do
        Do interventions on G_c in latent space as Eq. (10)
        Calculate architecture matrix A_c and {A_v j} from causal subgraph and their intervention graphs as Eq. (12)
    end for
    Calculate L_op using Eq. (13)
    Calculate L_pred using Eq. (11) (14)
    Calculate L_arch using Eq. (15))
    Calculate the overall loss L_all using Eq. (16)
    Update parameters using gradient descends
end for
```

4.1 Experiment setting

4.1.1 Setting. To ensure the reliability and reproducibility, we execute each experiment ten times using distinct random seeds and present the average results along with their standard deviations.

4.1.2 Baselines. We compare our model with 12 baselines from the following two different categories:

- **Manually design GNNs.** We incorporate widely recognized architectures: GCN [16], GAT [42], GIN [50], SAGE [12], and GraphConv [32], into our search space as well as baseline methods. Apart from that, we include MLP and two recent advancements: ASAP [40] and DIR [47], which is specifically proposed for out-of-distribution generalization.
- **Graph Neural Architecture Search.** For **classic NAS**, we compare with DARTS [28], a differentiable architecture search method, and random search. For **graph NAS**, we explore a reinforcement learning-based method GNAS [8], and PAS [44] that is specially designed for graph classification tasks. Additionally, we compare two state-of-the-art graph NAS methods that are specially designed for non-i.i.d. graph datasets, including GRACES [37] and DCGAS [52].

4.1.3 Definition of search space. The number of layers in our model is predetermined before training, and the type of operator for each layer can be selected from our defined operator search space O. We incorporate widely recognized architectures GCN, GAT, GIN, SAGE, GraphConv, and MLP into our search space as candidate operators in our experiments. This allows for the combination of various sub-architectures within a single model, such as using GCN in the first layer and GAT in the second layer. Furthermore, we consistently use standard global mean pooling at the end of the GNN architecture to generate a global embedding.

4.2 On synthetic datasets

4.2.1 Datasets. The synthetic dataset, **Spurious-Motif** [37, 47, 53], encompasses 18,000 graphs, each uniquely formed by combining a base shape (denoted as *Tree*, *Ladder*, or *Wheel* with $S = 0, 1, 2$) with a motif shape (represented as *Cycle*, *House*, or *Crane* with $C = 0, 1, 2$). Notably, the classification of each graph relies solely on its motif shape, despite the base shape typically being larger. This dataset is particularly designed to study the effect of distribution shifts, with a distinct bias introduced solely on the training set through the probability distribution $P(S) = b \times \mathbb{I}(S = C) + \frac{1-b}{2} \times \mathbb{I}(S \neq C)$, where b modulates the correlation between base and motif shapes, thereby inducing a deliberate shift between the training set and testing set, where all base and motif shapes are independent with equal probabilities. We choose $b = 0.7/0.8/0.9$, enabling us to explore our model's performance under various significant distributional variations. The effectiveness of our approach is measured using **accuracy** as the evaluation metric on this dataset.

4.2.2 Results. Table 1 presents the experimental results on three synthetic datasets, revealing that our model significantly outperforms all baseline models across different scenarios.

Specifically, we observe that the performance of all GNN models is particularly poor, suggesting their sensitivity to spurious correlations and their inability to adapt to distribution shifts. However, DIR [47], designed specifically for non-I.I.D. datasets and focusing on discovering invariant rationale to enhance generalizability, shows pretty well performance compared to most of the other GNN models. This reflects the feasibility of employing causal learning to tackle generalization issues.

Moreover, NAS methods generally yield slightly better outcomes than manually designed GNNs in most scenarios, emphasizing the significance of automating architecture by learning the correlations between input graph data and architecture to search for the optimal GNN architecture. Notably, methods specifically designed for non-I.I.D. datasets, such as GRACES [37], DCGAS [52], and our CARNAS, exhibit significantly less susceptibility to distribution shifts compared to NAS methods intended for I.I.D. data.

Among these, our approach consistently achieves the best performance across datasets with various degrees of shifts, demonstrating the effectiveness of our method in enhancing Graph NAS performance, especially in terms of out-of-distribution generalization, which is attained by effectively capturing causal invariant subgraphs to guide the architecture search process, and filtering out spurious correlations meanwhile.

4.3 On real-world datasets

4.3.1 Datasets. The real-world datasets **OGBG-Mol***, including Ogbg-molhiv, Ogbg-molbace, and Ogbg-molsider [13, 48], feature 41127, 1513, and 1427 molecule graphs, respectively, aimed at molecular property prediction. The division of the datasets is based on scaffold values, designed to segregate molecules according to their structural frameworks, thus introducing a significant challenge to the prediction of graph properties. The predictive performance of our approach across these diverse molecular structures and properties is measured using **ROC-AUC** as the evaluation metric.

Table 1: The test accuracy of all methods on synthetic dataset Spurious-Motif. Values after ± denote the standard deviations. The best results overall are in bold and the best results of baselines in each category are underlined separately.

Method	$b = 0.7$	$b = 0.8$	$b = 0.9$
GCN	$48.39_{\pm 1.69}$	$41.55_{\pm 3.88}$	$39.13_{\pm 1.76}$
GAT	$50.75_{\pm 4.89}$	$42.48_{\pm 2.46}$	$40.10_{\pm 5.19}$
GIN	$36.83_{\pm 5.49}$	$34.83_{\pm 3.10}$	$37.45_{\pm 3.59}$
SAGE	$46.66_{\pm 2.51}$	$44.50_{\pm 5.79}$	$44.79_{\pm 4.83}$
GraphConv	$47.29_{\pm 1.95}$	$44.67_{\pm 5.88}$	$44.82_{\pm 4.84}$
MLP	$48.27_{\pm 1.27}$	$46.73_{\pm 3.48}$	$\underline{46.41_{\pm 2.34}}$
ASAP	$\underline{54.07_{\pm 13.85}}$	$\underline{48.32_{\pm 12.72}}$	$43.52_{\pm 8.41}$
DIR	$50.08_{\pm 3.46}$	$48.22_{\pm 6.27}$	$43.11_{\pm 5.43}$
Random	$45.92_{\pm 4.29}$	$51.72_{\pm 5.38}$	$45.89_{\pm 5.09}$
DARTS	$50.63_{\pm 8.90}$	$45.41_{\pm 7.71}$	$44.44_{\pm 4.42}$
GNAS	$55.18_{\pm 18.62}$	$51.64_{\pm 19.22}$	$37.56_{\pm 5.43}$
PAS	$52.15_{\pm 4.35}$	$43.12_{\pm 5.95}$	$39.84_{\pm 1.67}$
GRACES	$65.72_{\pm 17.47}$	$59.57_{\pm 17.37}$	$50.94_{\pm 8.14}$
DCGAS	$\underline{87.68_{\pm 6.12}}$	$\underline{75.45_{\pm 17.40}}$	$\underline{61.42_{\pm 16.26}}$
CARNAS	$\mathbf{94.41}_{\pm 4.58}$	$\mathbf{88.04}_{\pm 13.77}$	$\mathbf{87.15}_{\pm 11.85}$

4.3.2 Results. Results from real-world datasets are detailed in Table 2, where our CARNAS model once again surpasses all baselines across three distinct datasets, showcasing its ability to handle complex distribution shifts under various conditions.

For manually designed GNNs, the optimal model varies across different datasets: GIN achieves the best performance on Ogbg-molhiv, GCN excels on Ogbg-molsider, and GraphConv leads on Ogbg-molbace. This diversity in performance confirms a crucial hypothesis in our work, that different GNN models are predisposed to perform well on graphs featuring distinct characteristics.

In the realm of NAS models, we observe that DARTS and PAS, proposed for I.I.D. datasets, perform comparably to manually crafted GNNs, whereas GRACES, DCGAS and our CARNAS , specifically designed for non-I.I.D. datasets outshine other baselines. Our approach reaches the top performance across all datasets, with a particularly remarkable breakthrough on Ogbg-molsider, highlighting our method's superior capability in adapting to and excelling within diverse data environments.

We provide reproducibility details in Appendix B, including dataset details, and hyperparameter settings.

5 Deeper Analysis

5.1 Ablation study

Setups. In this section, we conduct ablation studies to examine the effectiveness of each vital component in our framework. Specifically, we compare the following ablated variants of our model:

- 'CARNAS w/o $\mathcal{L}_{arch}$' removes $\mathcal{L}_{arch}$ from the overall loss in Eq. (16). In this way, the contribution of the *graph embedding intervention module* together with the *invariant architecture customization module* to improve generalization performance

Table 2: The test ROC-AUC of all methods on real-world datasets OGBG-Mol*. Values after ± denote the standard deviations. The best results overall are in bold and the best results of baselines in each category are underlined separately.

Method	HIV	SIDER	BACE
GCN	$75.99_{\pm 1.19}$	$\underline{59.84_{\pm 1.54}}$	$68.93_{\pm 6.95}$
GAT	$76.80_{\pm 0.58}$	$57.40_{\pm 2.01}$	$75.34_{\pm 2.36}$
GIN	$\underline{77.07_{\pm 1.49}}$	$57.57_{\pm 1.56}$	$73.46_{\pm 5.24}$
SAGE	$75.58_{\pm 1.40}$	$56.36_{\pm 1.32}$	$74.85_{\pm 2.74}$
GraphConv	$74.46_{\pm 0.86}$	$56.09_{\pm 1.06}$	$\underline{78.87_{\pm 1.74}}$
MLP	$70.88_{\pm 0.83}$	$58.16_{\pm 1.41}$	$71.60_{\pm 2.30}$
ASAP	$73.81_{\pm 1.17}$	$55.77_{\pm 1.18}$	$71.55_{\pm 2.74}$
DIR	$77.05_{\pm 0.57}$	$57.34_{\pm 0.36}$	$76.03_{\pm 2.20}$
DARTS	$74.04_{\pm 1.75}$	$60.64_{\pm 1.37}$	$76.71_{\pm 1.83}$
PAS	$71.19_{\pm 2.28}$	$59.31_{\pm 1.48}$	$76.59_{\pm 1.87}$
GRACES	$77.31_{\pm 1.00}$	$61.85_{\pm 2.58}$	$79.46_{\pm 3.04}$
DCGAS	$\underline{78.04_{\pm 0.71}}$	$\underline{63.46_{\pm 1.42}}$	$\underline{81.31_{\pm 1.94}}$
CARNAS	$\mathbf{78.33}_{\pm 0.64}$	$\mathbf{83.36}_{\pm 0.62}$	$\mathbf{81.73}_{\pm 2.92}$

by restricting the causally invariant nature for constructing architectures of the causal subgraph is removed.

- 'CARNAS w/o $\mathcal{L}_{cpred}$' removes $\mathcal{L}_{cpred}$, thereby relieving the supervised restriction on causal subgraphs for encapsulating sufficient graph features, which is contributed by *disentangled causal subgraph identification module* together with the *graph embedding intervention module* to enhance the learning of causal subgraphs.
- 'CARNAS w/o $\mathcal{L}_{arch}$ & $\mathcal{L}_{cpred}$' further removes both of them.

Besides, we also compare with the best performance in baselines.

Results. From Figure 2, we have the following observations. First of all, our proposed CARNAS outperforms all the variants as well as the best-performed baseline on all datasets, demonstrating the effectiveness of each component of our proposed method. Secondly, the performance of 'CARNAS w/o $\mathcal{L}_{arch}$', 'CARNAS w/o $\mathcal{L}_{cpred}$' and 'CARNAS w/o $\mathcal{L}_{arch}$ & $\mathcal{L}_{cpred}$' dropped obviously on all datasets comparing with the full CARNAS , which validates that our proposed modules help the model to identify stable causal components from comprehensive graph feature and further guide the Graph NAS process to enhance its performance significantly especially under distribution shifts. What's more, though 'CARNAS w/o $\mathcal{L}_{arch}$' decreases, its performance still surpasses the best results in baselines across all datasets, indicating that even if the invariance of the influence of the causal subgraph on the architecture is not strictly restricted by $\mathcal{L}_{arch}$, it is effective to use merely the causal subgraph guaranteed by $\mathcal{L}_{cpred}$ to contain the important information of the input graph and use it to guide the architecture search.

5.2 Case study

For graphs with different motif shapes (causal subparts), we present the learned operation probabilities for each layer (in expectation) in Figure 3. The values that are notably higher than others for each layer are highlighted in bold, and the most preferred operators for each layer are listed in the last row.

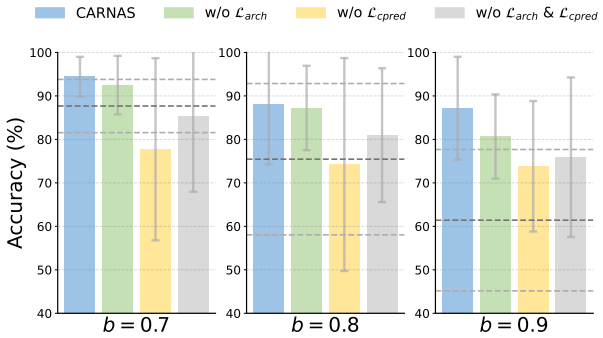


Figure 2: Results of ablation studies on synthetic datasets, where 'w/o $\mathcal{L}_{arch}$' removes $\mathcal{L}_{arch}$ from the overall loss in Eq. (16), 'w/o $\mathcal{L}_{cpred}$' removes $\mathcal{L}_{cpred}$, and 'w/o $\mathcal{L}_{arch}$ & $\mathcal{L}_{cpred}$' removes both of them. The error bars report the standard deviations. Besides, the average and standard deviations of the best-performed baseline on each dataset are denoted as the dark and light thick dash lines respectively.

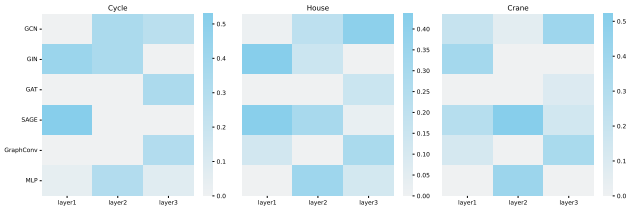
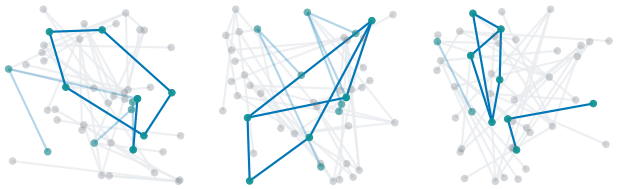


Figure 3: Comparison of operation probabilities for graphs with different motif shapes.

We observe that different motif shapes indeed prefer different architectures, e.g., graphs with cycle prefer GAT in the third layer, while this operator is seldom chosen in neither layer of the other two types of graphs; the operator distributions are similar for graphs with cycle and house in the first layer, but differ in other layers. To be specific, Motif-Cycle is characterized by a closed-loop structure where each node is connected to two neighbors, displaying both symmetry and periodicity. For graphs with this motif, CARNAS identifies SAGE-GCN-GAT as the most suitable architecture. Motif-House, on the other hand, features a combination of triangular and quadrilateral structures, introducing a certain level of hierarchy and asymmetry. For graphs with this shape, CARNAS determines that GIN-MLP-GCN is the optimal configuration. Lastly, Motif-Crane presents more complex cross-connections between nodes compared to the previous two motifs, and CARNAS optimally configures graphs with it with a GIN-SAGE-GCN architecture.

By effectively integrating various operations and customizing specific architectures for different causal subparts (motifs) with diverse features, our NAS-based CARNAS can further improve OOD generalization.



(a) Cycle (b) House (c) Crane

Figure 4: Visualization of edge importance for forming causal subgraphs in SP-Motif Dataset. Structures with deeper colors mean higher importance.

To better illustrate the learned graph-architecture relationship, we also visualize the causal subgraphs for each dataset in our case study in Figure 4.

5.3 Training process

Furthermore, we report both the training loss and validation loss for the two components ($\mathcal{L}_{causal}$, representing the causal-aware part, and $\mathcal{L}_{pred}$, representing the customized super-network optimization as defined in Equation (16)) in the following settings:

- 'with Dynamic σ' means we use the dynamic σ_p in Eq.(17) to adjust the training key point in each epoch.
- 'w/o Dynamic σ' means we fix the σ in Eq.(16) as a constant value $\frac{\sigma_{max}+\sigma_{min}}{2}$.

For the training loss, $\mathcal{L}_{pred}$ decreases more steadily and reaches a lower value with less fluctuation under the dynamic schedule. In terms of validation loss, $\mathcal{L}_{pred}$ with the dynamic schedule decreases significantly in later stages, whereas without it, $\mathcal{L}_{pred}$ struggles to converge. Additionally, $\mathcal{L}_{causal}$ without the dynamic schedule exhibits a slight initial increase before decreasing, whereas with the dynamic schedule, it decreases smoothly from the outset. These results indicate that the dynamic schedule effectively adjusts the training focus during each epoch. It emphasizes the *causal-aware part* (i.e., identifying suitable causal subgraphs and learning operator vectors) in the early stages and shifts focus to the *customized super-network* performance in later stages.

Additionally, according to Figure 6, *our method can converge rapidly in 10 epochs*. Figure 6 also obviously reflects that after 10 epochs the validation loss with dynamic σ keeps declining and its *accuracy continuously rising*. However, in the setting without dynamic σ, the validation loss may rise again, and accuracy cannot continue to improve. These results verify our aim to adopt this σ_p to *elevate the efficiency of model training* in the way of dynamically adjusting the training key point in each epoch.

We also illustrate the efficiency of CARNAS, we provide a direct comparison with the best-performed NAS baseline, DCGAS, based on the total runtime for 100 epochs. As shown in Table 3, CARNAS consistently requires less time across different datasets while achieving superior best performance, demonstrating its enhanced efficiency and effectiveness.

Table 3: Comparison of runtime

Method	SPMotif	HIV	BACE	SIDER
DCGAS	104 min	270 min	12 min	11 min
CARNAS	76 min	220 min	8 min	8 min

6 Related Work

6.1 Graph neural architecture search

In the rapidly evolving domain of automatic machine learning, Neural Architecture Search (NAS) represents a groundbreaking shift towards automating the discovery of optimal neural network architectures. This shift is significant, moving away from the traditional approach that heavily relies on manual expertise to craft models. NAS stands out by its capacity to autonomously identify architectures that are finely tuned for specific tasks, demonstrating superior performance over manually engineered counterparts. The exploration of NAS has led to the development of diverse strategies, including reinforcement learning (RL)-based approaches [14], evolutionary algorithms-based techniques [29], and methods that leverage gradient information [28]. Among these, graph neural architecture search has garnered considerable attention.

The pioneering work of GraphNAS [8] introduced the use of RL for navigating the search space of graph neural network (GNN) architectures, incorporating successful designs from the GNN literature such as GCN, GAT, etc. This initiative has sparked a wave of research [8–10, 35–39, 44, 49, 63], leading to the discovery of innovative and effective architectures. Recent years have seen a broadening of focus within Graph NAS towards tackling graph classification tasks, which are particularly relevant for datasets comprised of graphs, such as those found in protein molecule studies. This research area has been enriched by investigations into graph classification on datasets that are either independently identically distributed [44] or non-independently identically distributed, with [2, 37, 52] being notable examples of latter. Through these efforts, the field of NAS continues to expand its impact, offering tailored solutions across a wide range of applications and datasets.

6.2 Graph out-of-distribution generalization

In the realm of machine learning, a pervasive assumption posits the existence of identical distributions between training and testing data. However, real-world scenarios frequently challenge this assumption with inevitable shifts in distribution, presenting significant hurdles to model performance in out-of-distribution (OOD) scenarios [57, 58, 60]. The drastic deterioration in performance becomes evident when models lack robust OOD generalization capabilities, a concern particularly pertinent in the domain of Graph Neural Networks (GNNs), which have gained prominence within the graph community [21, 43, 59]. Several noteworthy studies [19, 22, 23, 46, 47] have tackled this challenge by focusing on identifying environment-invariant subgraphs to mitigate distribution shifts. These approaches typically rely on pre-defined or dynamically generated environment labels from various training scenarios to discern variant information and facilitate the learning of invariant subgraphs. [45, 55] have divided recent literature that

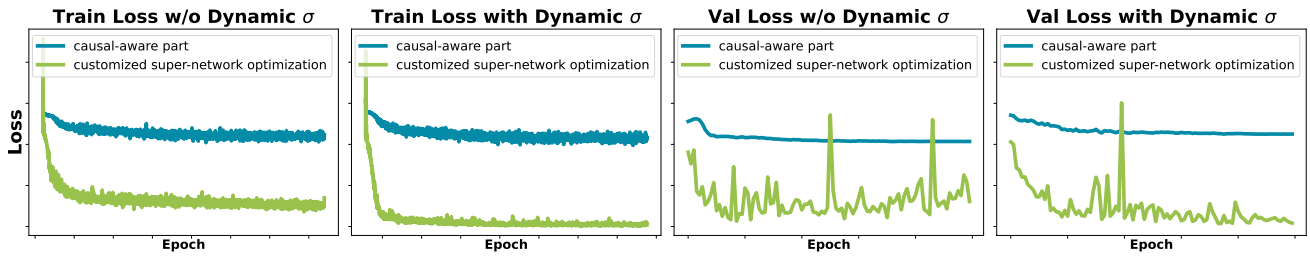


Figure 5: Changes of the two parts of loss.

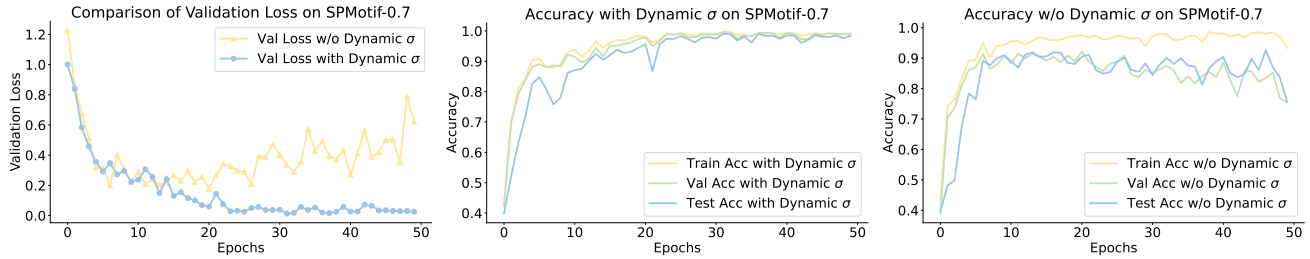


Figure 6: Training process of synthetic datasets.

solve the graph OOD generalization problem, into three categories: Graph augmentation methods [54] enhance OOD generalization by increasing the quantity and diversity of training data through systematic graph modifications. The second type of methods [20, 31] develop new graph models to learn OOD-generalized representations. The third type of methods [56] enhance OOD generalization through tailored training schemes with specific objectives and constraints. There are various datasets and benchmarks [11, 15] help for assessing generalizability and adaptability. Moreover, the existing methods usually adopt a fixed GNN encoder in the whole optimization process, neglecting the role of graph architectures in out-of-distribution generalization. In this paper, we focus on automating the design of generalized graph architectures by discovering causal relationships between graphs and architectures, and thus handle distribution shifts on graphs.

6.3 Causal learning on graphs

The field of causal learning investigates the intricate connections between variables [24–26, 34], offering profound insights that have significantly enhanced deep learning methodologies. Leveraging causal relationships, numerous techniques have made remarkable strides across diverse computer vision applications. Recent research has delved into the realm of graphs [61, 62]. For instance, [47] implements interventions on non-causal components to generate representations, facilitating the discovery of underlying graph rationales. [7] decomposes graphs into causal and bias subgraphs, mitigating dataset biases. [27] introduces invariance into self-supervised learning, preserving stable semantic information. [4] ensures out-of-distribution generalization by capturing graph invariance. However, these methods adopt a fixed GNN architecture in the optimization process, neglecting the role of architectures in causal learning on graphs. In contrast, in this paper, we focus on handling distribution shifts in the graph architecture search process from the causal perspective by discovering the causal relationship between graphs and architectures.

7 Conclusion

In this paper, we focus on tackling distribution shifts in graph neural architecture search (Graph NAS) from the causal perspective. While existing methods have shown promise in designing graph neural network architectures, they often struggle with distribution shifts between training and testing sets, since the correlations between graphs and architectures they exploit may be spurious and varying across distributions. To mitigate this issue, we introduce a novel approach, Causal-aware Graph Neural Architecture Search (**CARNAS**), which focuses on discerning stable causal structures and their relationship with architectures during the architecture search process. Specifically, we propose three key modules, including Disentangled Causal Subgraph Identification, Graph Embedding Intervention, and Invariant Architecture Customization, which are able to effectively identify and leverage the causal relationships between graph structures and architectures to search generalized graph neural architectures. Our extensive experiments on synthetic and real-world datasets demonstrate that CARNAS achieves superior out-of-distribution generalization ability, highlighting the importance of incorporating causal awareness into the graph neural architecture search process.

Acknowledgments

This work is supported by National Natural Science Foundation of China No.62222209, No.62472018, Beijing National Research Center for Information Science and Technology under Grant No.BNR2023TD03006, and Alibaba Innovative Research Program.

References

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

[2] Jie Cai, Xin Wang, Haoyang Li, Ziwei Zhang, and Wenwu Zhu. 2024. Multimodal graph neural architecture search under distribution shifts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8227–8235.

[3] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2020. Invariant rationalization. In *International Conference on Machine Learning*. PMLR, 1448–1458.

[4] Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, MA Kaili, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. 2022. Learning causally invariant representations for out-of-distribution generalization on graphs. *Advances in Neural Information Processing Systems* 35 (2022), 22131–22148.

[5] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. 2020. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems* 33 (2020), 13260–13271.

[6] Abbas El Gamal and Young-Han Kim. 2011. *Network information theory*. Cambridge university press.

[7] Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang. 2022. Debiasing graph neural networks via learning disentangled causal substructure. *Advances in Neural Information Processing Systems* (2022).

[8] Yang Gao, Hong Yang, Peng Zhang, Chuan Zhou, and Yue Hu. 2021. Graph neural architecture search. In *International joint conference on artificial intelligence*. International Joint Conference on Artificial Intelligence.

[9] Chaoyu Guan, Xin Wang, Hong Chen, Ziwei Zhang, and Wenwu Zhu. 2022. Large-scale graph neural architecture search. In *International Conference on Machine Learning*. PMLR, 7968–7981.

[10] Chaoyu Guan, Xin Wang, and Wenwu Zhu. 2021. Autoattend: Automated attention representation search. In *International conference on machine learning*. PMLR, 3864–3874.

[11] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. 2022. Good: A graph out-of-distribution benchmark. *Advances in Neural Information Processing Systems* 35 (2022), 2059–2073.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

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

[14] Yesmina Jaafra, Jean Luc Laurent, Aline Deruyver, and Mohamed Saber Naceur. 2019. Reinforcement learning for neural architecture search: A review. *Image and Vision Computing* 89 (2019), 57–66.

[15] Yuanfeng Ji, Lu Zhang, Jiaxiang Wu, Bingzhe Wu, Lanqing Li, Long-Kai Huang, Tingyang Xu, Yu Rong, Jie Ren, Ding Xue, et al. 2023. Drugood: Out-of-distribution dataset curator and benchmark for ai-aided drug discovery–a focus on affinity prediction problems with noise annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 8023–8031.

[16] Thomas N Kipf and Max Welling. 2022. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

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

[18] Guohao Li, Guocheng Qian, Itzel C Delgadillo, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2020. Sgas: Sequential greedy architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1620–1630.

[19] Haoyang Li, Xin Wang, Zeyang Zhang, Haibo Chen, Ziwei Zhang, and Wenwu Zhu. 2024. Disentangled Graph Self-supervised Learning for Out-of-Distribution Generalization. In *ICML*.

[20] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2022. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[21] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2022. Out-of-distribution generalization on graphs: A survey. *arXiv preprint arXiv:2202.07987* (2022).

[22] Haoyang Li, Xin Wang, Xueling Zhu, Weigao Wen, and Wenwu Zhu. 2025. Disentangling invariant subgraph via variance contrastive estimation under distribution shifts. In *International Conference on Machine Learning*.

[23] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. 2022. Learning invariant graph representations for out-of-distribution generalization. *Advances in Neural Information Processing Systems* 35 (2022), 11828–11841.

[24] Peiwen Li, Yuan Meng, Xin Wang, Fang Shen, Yue Li, Jialong Wang, and Wenwu Zhu. 2023. Causal discovery in temporal domain from interventional data. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4074–4078.

[25] Peiwen Li, Xin Wang, Zeyang Zhang, Yuan Meng, Fang Shen, Yue Li, Jialong Wang, Yang Li, and Wenwu Zhu. 2024. RealTCD: temporal causal discovery from interventional data with large language model. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 4669–4677.

[26] Peiwen Li and Menghua Wu. 2024. Learning to refine domain knowledge for biological network inference. *arXiv preprint arXiv:2410.14436* (2024).

[27] Sihang Li, Xiang Wang, An Zhang, Yingxin Wu, Xiangnan He, and Tat-Seng Chua. 2022. Let invariant rationale discovery inspire graph contrastive learning. In *ICML*. 13052–13065.

[28] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2018. DARTS: Differentiable Architecture Search. In *International Conference on Learning Representations*.

[29] Yuqiao Liu, Yanan Sun, Bing Xue, Mengjie Zhang, Gary G Yen, and Kay Chen Tan. 2021. A survey on evolutionary neural architecture search. *IEEE transactions on neural networks and learning systems* (2021).

[30] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. *Advances in neural information processing systems* 33 (2020), 19620–19631.

[31] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled graph convolutional networks. In *International conference on machine learning*. PMLR, 4212–4221.

[32] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 4602–4609.

[33] Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress* 19, 2 (2000), 3.

[34] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

[35] Yijian Qin, Xin Wang, Peng Cui, and Wenwu Zhu. 2021. Gqnas: Graph q network for neural architecture search. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1288–1293.

[36] Yijian Qin, Xin Wang, Ziwei Zhang, Hong Chen, and Wenwu Zhu. 2023. Multi-task graph neural architecture search with task-aware collaboration and curriculum. *Advances in neural information processing systems* 36 (2023), 24879–24891.

[37] Yijian Qin, Xin Wang, Ziwei Zhang, Pengtao Xie, and Wenwu Zhu. 2022. Graph neural architecture search under distribution shifts. In *International Conference on Machine Learning*. PMLR, 18083–18095.

[38] Yijian Qin, Xin Wang, Zeyang Zhang, and Wenwu Zhu. 2021. Graph differentiable architecture search with structure learning. *Advances in neural information processing systems* 34 (2021), 16860–16872.

[39] Yijian Qin, Ziwei Zhang, Xin Wang, Zeyang Zhang, and Wenwu Zhu. 2022. NAS-Bench-Graph: Benchmarking Graph Neural Architecture Search. In *Advances in Neural Information Processing Systems*.

[40] Ekagra Ranjan, Soumya Sanyal, and Partha Talukdar. 2020. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5470–5477.

[41] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. 2018. Invariant models for causal transfer learning. *Journal of Machine Learning Research* 19, 36 (2018), 1–34.

[42] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat* 1050, 20 (2017), 10–48550.

[43] Xin Wang, Haoyang Li, Zeyang Zhang, and Wenwu Zhu. 2025. Graph Machine Learning under Distribution Shifts: Adaptation, Generalization and Extension to LLM. In *Companion Proceedings of the ACM on Web Conference 2025* (Sydney NSW, Australia) *(WWW '25)*. Association for Computing Machinery, New York, NY, USA, 53–56. doi:10.1145/3701716.3715863

[44] Lanning Wei, Huan Zhao, Quanming Yao, and Zhiqiang He. 2021. Pooling architecture search for graph classification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2091–2100.

[45] Man Wu, Xin Zheng, Qin Zhang, Xiao Shen, Xiong Luo, Xingquan Zhu, and Shirui Pan. 2024. Graph Learning under Distribution Shifts: A Comprehensive Survey on Domain Adaptation, Out-of-distribution, and Continual Learning. *arXiv preprint arXiv:2402.16374* (2024).

[46] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. 2022. Handling Distribution Shifts on Graphs: An Invariance Perspective. *International Conference on Learning Representations* (2022).

[47] Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. 2022. Discovering Invariant Rationales for Graph Neural Networks. In *International Conference on Learning Representations*.

[48] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical science* 9, 2 (2018), 513–530.

[49] Beini Xie, Heng Chang, Ziwei Zhang, Zeyang Zhang, Simin Wu, Xin Wang, Yuan Meng, and Wenwu Zhu. 2024. Towards Lightweight Graph Neural Network Search with Curriculum Graph Sparsification. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3563–3573.

[50] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*.
[51] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon Shaolei Du, Ken-Ichi Kawarabayashi, and Stefanie Jegelka. 2020. How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks. In *International Conference on Learning Representations*.
[52] Yang Yao, Xin Wang, Yijian Qin, Ziwei Zhang, Wenwu Zhu, and Hong Mei. 2024. Data-augmented curriculum graph neural architecture search under distribution shifts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 16433–16441.
[53] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems* 32 (2019).
[54] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems* 33 (2020), 5812–5823.
[55] Kexin Zhang, Shuhan Liu, Song Wang, Weili Shi, Chen Chen, Pan Li, Sheng Li, Jundong Li, and Kaize Ding. 2024. A Survey of Deep Graph Learning under Distribution Shifts: from Graph Out-of-Distribution Generalization to Adaptation. *arXiv preprint arXiv:2410.19265* (2024).
[56] Shengyu Zhang, Kun Kuang, Jiezhong Qiu, Jin Yu, Zhou Zhao, Hongxia Yang, Zhongfei Zhang, and Fei Wu. 2021. Stable Prediction on Graphs with Agnostic Distribution Shift. *arXiv preprint arXiv:2110.03865* (2021).
[57] Zeyang Zhang, Xingwang Li, Fei Teng, Ning Lin, Xueling Zhu, Xin Wang, and Wenwu Zhu. 2023. Out-of-Distribution Generalized Dynamic Graph Neural Network for Human Albumin Prediction. In *IEEE International Conference on Medical Artificial Intelligence*.
[58] Zeyang Zhang, Ning Lin, Xingwang Li, Xueling Zhu, Fei Teng, Xin Wang, and Wenwu Zhu. 2023. Out-of-distribution generalized dynamic graph neural network for human albumin prediction. In *2023 IEEE International Conference on Medical Artificial Intelligence (MedAI)*. IEEE, 153–164.
[59] Zeyang Zhang, Xin Wang, Haibo Chen, Haoyang Li, and Wenwu Zhu. 2024. Disentangled Dynamic Graph Attention Network for Out-of-Distribution Sequential Recommendation. *ACM Trans. Inf. Syst.* 43, 1, Article 19 (Dec. 2024), 42 pages. doi:10.1145/3701988
[60] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. 2022. Dynamic graph neural networks under spatio-temporal distribution shift. In *Advances in Neural Information Processing Systems*.
[61] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, and Wenwu Zhu. 2023. Out-of-Distribution Generalized Dynamic Graph Neural Network with Disentangled Intervention and Invariance Promotion. *arXiv preprint arXiv:2311.14255* (2023).
[62] Zeyang Zhang, Xin Wang, Ziwei Zhang, Zhou Qin, Weigao Wen, Hui Xue, Haoyang Li, and Wenwu Zhu. 2023. Spectral Invariant Learning for Dynamic Graphs under Distribution Shifts. In *Advances in Neural Information Processing Systems*.
[63] Zeyang Zhang, Ziwei Zhang, Xin Wang, Yijian Qin, Zhou Qin, and Wenwu Zhu. 2023. Dynamic Heterogeneous Graph Attention Neural Architecture Search. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*.

A Theoretical Analysis

In this section, in order to more rigorously establish our method, we provide a theoretical analysis about the problem of identifying and leveraging causal graph-architecture relationship to find the optimal architecture.

To begin with, since causal relationships are, by definition, invariant across environments, we make the below assumption on our causal invariant subgraph generator $f_C(G) = G_c : \mathbb{G} \to \mathbb{G}_c$, following previous literature on invariant learning [23, 41].

Assumption 1. *There exists an optimal causal invariant subgraph generator $f_C(G)$ satisfying: (1) Invariance property: For all $e, e' \in supp(\mathcal{E})$, $P^e(A^*|f_C(G)) = P^{e'}(A^*|f_C(G))$. (2) Sufficiency property: $A^* = f_A(f_C(G)) + \epsilon$, where $f_A(\cdot)$ customizes the GNN architecture from a graph, $\epsilon \perp G$ (indicating statistical independence), and ϵ is random noise.*

Invariance assumption indicates that the subgraph generator $f_C(G)$ is capable of generating invariant subgraphs across different environments $e, e' \in \text{supp}(\mathcal{E})$, where $\mathcal{E}$ is a random variable of all environments. This ensures that the conditional distribution $P(A^*|f_C(G))$ remains consistent and unaffected by the environment. Sufficiency assumption demonstrates that the subgraph generated by $f_C(G)$ has sufficient expressive power to enable prediction of the optimal architecture A^*. This is achieved through $f_A(\cdot)$, customizing the GNN architecture from a graph, while the added random noise ϵ is independent of the graph G.

Then, how can we get the optimal causal invariant subgraph generator? Following previous work[23], we can prove that it can be obtain through maximizing $I(A^*; f_C(G))$, i.e. the mutual information between optimal architecture and the generated subgraph.

Theorem 1 (Optimal Generator of Causal Subgraphs). *A generator $f_C(G)$ is the optimal generator that satisfies Assumption 1 if and only if it is the maximal causal subgraph generator, i.e.,*

$$f_C^* = \arg \max_{f_C \in \mathcal{F}_\mathcal{E}} I(A^*; f_C(G)), \tag{18}$$

where $\mathcal{F}_\mathcal{E}$ is the subgraph generator set with related to the random vector of all environments, and $I(\cdot;\cdot)$ is the mutual information between the optimal architecture A^* and the generated causal subgraph.

Proof. Let $\hat{f}_C = \arg\max_{f_C \in \mathcal{F}_\mathcal{E}} I(A^*; f_C(G))$. From the invariance property in Assumption 1, it follows that $f_C^* \in \mathcal{F}_\mathcal{E}$. To prove the theorem, we show that: $I(A^*; \hat{f}_C(G)) \le I(A^*; f_C^*(G))$, which implies $\hat{f}_C = f_C^*$. Using the functional representation lemma [6], any random variable X_2 can be expressed as a function of another random variable X_1 and an independent random variable X_3. Applying this to $f_C^*(G)$ and $\hat{f}_C(G)$, there exists a $f_C'(G)$ such that $f_C'(G) \perp f_C^*(G)$ and $\hat{f}_C(G) = \gamma(f_C^*(G), f_C'(G))$, where $\gamma(\cdot)$ is a deterministic function. Then, the mutual information can be decomposed as follows:

$$\begin{aligned} I(A^*; \hat{f}_C(G)) &= I(A^*; \gamma(f_C^*(G), f_C'(G))) \le I(A^*; f_C^*(G), f_C'(G)) \\ &= I(f_A(f_C^*(G)); f_C^*(G), f_C'(G)) \\ &= I(f_A(f_C^*(G)); f_C^*(G)) = I(A^*; f_C^*(G)), \end{aligned} \tag{19}$$

which completes the proof.

Since maximizing $I(A^*; f_C(G))$ is difficult, we transform it into another way which will be introduced in later passage.

Next, we show the theorem that guide us to optimize the model, represented as Q, that can construct optimal architecture A^* for graph instance G under distribution shifts. The prediction is based on the causal subgraph G_c^*, which delineates the causal graph-architecture relationship. We denote the conditional distribution modeled by Q as $q(A^*|G_c^*)$.

Theorem 2. *Let f_C^* be the optimal causal invariant subgraph generator from Assumption 1, and let $G_c^* = f_C^*(G)$ and $G_s^* = G \setminus G_c^*$. Then, we can get the optimal model Q under distribution shifts by minimizing the following objective:*

$$\min \mathbb{E}\left[\log \frac{p(A^*|G_c^*)}{q(A^*|G_c^*)}\right] + I(G_s^*; A^*|G_c^*). \tag{20}$$

Here, $I(G_s^*; A^*|G_c^*)$ quantifies the spurious correlation between G_s^* and A^*, which the model need to ignore, and the first term ensures that $q(A^*|G_c^*)$ closely matches $p(A^*|G_c^*)$.

Proof. From the sufficiency assumption of f_C^* in Assumption 1, we know that: $A^* = f_A(f_C(G)) + \epsilon$, where $\epsilon \perp G$. This implies that A^* is conditionally independent of G_s^* (i.e., the non-causal subgraph) given G_c^*. Therefore, the full graph $G = (G_c^*, G_s^*)$ satisfies: $P(A^*|G) = P(A^*|G_c^*)$. Additionally, by the invariance property, for any $e, e' \in \text{supp}(\mathcal{E})$, the

conditional distribution of A^* given G_c^* remains invariant across environments: $P^e(A^*|G_c^*) = P^{e'}(A^*|G_c^*)$. This invariance guarantees that Q will generalize well under distribution shifts caused by changes in the environment, when $q(A^*|G_c^*)$ approximates the stable $p(A^*|G_c^*)$. To approximate $p(A^*|G_c^*)$ with $q(A^*|G_c^*)$, we minimize the negative conditional log-likelihood of the observed data: $-\ell = -\sum_{i=1}^{n} \log q(A_i^*|G_{c_i}^*)$. Expanding this objective using $G = (G_c^*, G_s^*)$, we rewrite it as:

$$-\ell = \sum_{i=1}^{n} \log \frac{p(A_i^*|G_{c_i}^*)}{q(A_i^*|G_{c_i}^*)} + \sum_{i=1}^{n} \log \frac{p(A_i^*|G_i)}{p(A_i^*|G_{c_i}^*)} - \sum_{i=1}^{n} \log p(A_i^*|G_i) \quad (21)$$

$$= \mathbb{E}\left[\log \frac{p(A^*|G_c^*)}{q(A^*|G_c^*)}\right] + \mathbb{E}\left[\log \frac{p(A^*|G)}{p(A^*|G_c^*)}\right] - \mathbb{E}\left[\log p(A^*|G)\right]. \quad (22)$$

The third term is irreducible constant inherent in the dataset, so we omit it when optimizing. Then, we decompose G into (G_c^*, G_s^*) and rewrite the second term as:

$$\mathbb{E}\left[\log \frac{p(A^*|G)}{p(A^*|G_c^*)}\right] = \mathbb{E}\left[\log \frac{p(A^*|G_c^*, G_s^*)}{p(A^*|G_c^*)}\right] \quad (23)$$

$$= \sum_{i=1}^{n} p(G_{c_i}^*, G_{s_i}^*, A_i^*) \log \frac{p(A_i^*|G_{c_i}^*, G_{s_i}^*)}{p(A_i^*|G_{c_i}^*)} \quad (24)$$

$$= \sum_{i=1}^{n} p(G_{c_i}^*, G_{s_i}^*, A_i^*) \log \frac{p(A_i^*|G_{c_i}^*, G_{s_i}^*) p(G_{s_i}^*|G_{c_i}^*)}{p(A_i^*|G_{c_i}^*) p(G_{s_i}^*|G_{c_i}^*)} \quad (25)$$

$$= \sum_{i=1}^{n} p(G_{c_i}^*, G_{s_i}^*, A_i^*) \log \frac{p(G_{s_i}^*, A_i^*|G_{c_i}^*)}{p(A_i^*|G_{c_i}^*) p(G_{s_i}^*|G_{c_i}^*)} = I(G_s^*; A^*|G_c^*). \quad (26)$$

Thus, the final objective to optimize $q(A^*|G_c^*)$ is: $\min \mathbb{E}\left[\log \frac{p(A^*|G_c^*)}{q(A^*|G_c^*)}\right] + I(G_s^*; A^*|G_c^*)$, where the second term, $I(G_s^*; A^*|G_c^*)$, measures the residual spurious correlation between G_s^* and A^* given G_c^*. This concludes the proof.

However, this objective is challenging to optimize directly in practice. To address this, we analyze each term intuitively and explain how our method is derived from the theorem.

The first term, $\mathbb{E}\left[\log \frac{p(A^*|G_c^*)}{q(A^*|G_c^*)}\right]$, ensures that the model accurately approximates the true conditional distribution $p(A^*|G_c^*)$ based on the causal subgraph G_c^*. Since the optimal architecture A^* is defined as the one achieving the best predictive performance on label Y, we indirectly optimize the first term $\mathbb{E}\left[\log \frac{p(A^*|G_c^*)}{q(A^*|G_c^*)}\right]$ by focusing on label's prediction performance. Specifically, we minimize $\mathcal{L}_{pred}$ (Equation 14), which measures the loss between the ground-truth label and the prediction from the learned optimal architecture A^*/A_c. This surrogate loss guides $q(A^*|G_c^*)$ to approximate $p(A^*|G_c^*)$, as A^* is inherently tied to the optimal predictive performance on final task.

The second term $I(G_s^*; A^*|G_c^*)$ represents the conditional mutual information between the optimal architecture A^* and the spurious subgraph G_s^*, given the causal subgraph G_c^*. Minimizing this term encourages the model to reduce its reliance on the spurious subgraph G_s^* when predicting the optimal architecture, given G_c^*. This motivates the use of $\mathcal{L}_{arch}$ in Equation 15, which measures the variance of simulated architectures corresponding to intervention graphs formed by combining the causal subgraph with different spurious subgraphs. By reducing this variance, the model is encouraged to rely solely on the causal subgraph G_c^* for determining the optimal architecture, ensuring that the causal subgraph has a stable and consistent predictive capability across varying spurious components in input graph G. Then, we prove that $I(G_s^*; A^*|G_c^*) = I(G; A^*) - I(G_c^*; A^*)$:

Proof. By the chain rule of mutual information, we have

$$I(G; A^*) = I(G_c^*, G_s^*; A^*) = I(G_c^*; A^*) + I(G_s^*; A^*|G_c^*), \quad (27)$$

where $G = (G_c^*, G_s^*)$. Rearranging the equation, we obtain

$$I(G_s^*; A^*|G_c^*) = I(G; A^*) - I(G_c^*; A^*). \quad (28)$$

Thus, minimizing $I(G_s^*; A^*|G_c^*)$ in turn encourages maximizing $I(A^*; f_C(G))$, which proved to lead to optimizing the causal subgraph generator in Theorem 1.

Therefore, we propose to **jointly optimize causal graph - architecture relationship and architecture search** by offering an **end-to-end training strategy** for extracting and utilizing causal relationships between graph data and architecture, which is stable under distribution shifts, during the architecture search process, thereby enhancing the model's capability of OOD generalization.

B Reproducibility details

B.1 Datasets details

We utilize synthetic SPMotif datasets, which are characterized by three distinct degrees of distribution shifts, and three different real-world datasets, each with varied components, following previous works [37, 47, 52]. Based on the statistics of each dataset as shown in Table 4, we conducted a comprehensive comparison across *various scales and graph sizes*. The real-world datasets are 3 molecular property prediction datasets in OGB [13], and are adopted from the MoleculeNet [48]. Each graph represents a molecule, where nodes are atoms, and edges are chemical bonds. The division of the datasets is based on scaffold values, designed to segregate molecules according to their structural frameworks, thus introducing a significant challenge to the prediction of graph properties.

Table 4: Statistics for different datasets.

	Graphs	Avg. Nodes	Avg. Edges
ogbg-molhiv	41127	25.5	27.5
ogbg-molsider	1427	33.6	35.4
ogbg-molbace	1513	34.1	36.9
SPMotif-0.7/0.8/0.9	18000	26.1	36.3

B.2 Detailed hyper-parameter settings

We fix the number of latent features $Q = 4$ in Eq. (4), number of intervention candidates N_s as batch size in Eq. (10), $\sigma_{min} = 0.1$, $\sigma_{max} = 0.7$, $P = 100$ in Eq. (17), and the tuned hyper-parameters for each dataset are as in Table 5.

Table 5: Hyper-parameter settings

Dataset	t in Eq. (6)	μ in Eq. (10)	θ_1 in Eq. (16)	θ_2 in Eq. (16)
SPMotif-0.7/0.8/0.9	0.85	0.26	0.36	0.010
ogbg-molhiv	0.46	0.68	0.94	0.007
ogbg-molsider	0.40	0.60	0.85	0.005
ogbg-molbace	0.49	0.54	0.80	0.003