Enhancing Out-of-distribution Generalization on Graphs via Causal Attention Learning

YONGDUO SUI, University of Science and Technology of China, Hefei, China
WENYU MAO, University of Science and Technology of China, Hefei, China
SHUYAO WANG, University of Science and Technology of China, Hefei, China
XIANG WANG, MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China, Hefei, China
JIANCAN WU, University of Science and Technology of China, Hefei, China
XIANGNAN HE, MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China, Hefei, China
TAT-SENG CHUA, National University of Singapore, Singapore, Singapore

In graph classification, attention- and pooling-based graph neural networks (GNNs) predominate to extract salient features from the input graph and support the prediction. They mostly follow the paradigm of "learning to attend," which maximizes the mutual information between the attended graph and the ground-truth label. However, this paradigm causes GNN classifiers to indiscriminately absorb all statistical correlations between input features and labels in the training data without distinguishing the causal and noncausal effects of features. Rather than emphasizing causal features, the attended graphs tend to rely on noncausal features as shortcuts to predictions. These shortcut features may easily change outside the training distribution, thereby leading to poor generalization for GNN classifiers. In this article, we take a causal view on GNN modeling. Under our causal assumption, the shortcut feature serves as a confounder between the causal feature and prediction. It misleads the classifier into learning spurious correlations that facilitate prediction in in-distribution (ID) test evaluation while causing significant performance drop in out-of-distribution (OOD) test data. To address this issue, we employ the backdoor adjustment from causal theory—combining each causal feature with various shortcut features, to identify causal patterns and mitigate the confounding effect. Specifically, we employ attention modules to estimate the causal and shortcut features of the input graph. Then, a memory bank collects the estimated shortcut features, enhancing the diversity of shortcut features for combination. Simultaneously, we apply the prototype strategy to improve the consistency of intra-class causal features. We term our method as CAL+, which can promote stable relationships between causal estimation and prediction, regardless of distribution changes. Extensive experiments on synthetic and real-world

Xiang Wang is also affiliated with Institute of Artificial Intelligence, Institute of Dataspace, Hefei Comprehensive National Science Center.

This research is supported by the National Key Research and Development Program of China (2021ZD0111802) and the National Natural Science Foundation of China (92270114).

Authors' addresses: Y. Sui, W. Mao, S. Wang, and J. Wu, University of Science and Technology of China, Hefei, China; e-mails: syd2019@mail.ustc.edu.cn, wenyum12345@gmail.com, shuyaowang@mail.ustc.edu.cn, wujcan@gmail.com; X. Wang and X. He, MoE Key Laboratory of Braininspired Intelligent Perception and Cognition, University of Science and Technology of China, Hefei, China; e-mails: xiangwang1223@gmail.com, xiangnanhe@gmail.com; T.-S. Chua, National University of Singapore, Singapore, Singapore; e-mail: dcscts@nus.edu.sg.

ACM 1556-4681/2024/03-ART127
https://doi.org/10.1145/3644392

OOD benchmarks demonstrate our method's effectiveness in improving OOD generalization. Our codes are released at https://github.com/shuyao-wang/CAL-plus.

CCS Concepts: • **Mathematics of computing → Graph algorithms**; • **Computing methodologies → Neural networks;**

Additional Key Words and Phrases: Graph learning, attention mechanism, out-of-distribution generalization

ACM Reference Format:
Yongduo Sui, Wenyu Mao, Shuyao Wang, Xiang Wang, Jiancan Wu, Xiangnan He, and Tat-Seng Chua. 2024. Enhancing Out-of-distribution Generalization on Graphs via Causal Attention Learning. *ACM Trans. Knowl. Discov. Data.* 18, 5, Article 127 (March 2024), 24 pages. https://doi.org/10.1145/3644392

1 INTRODUCTION

Graph neural networks (GNNs) [12, 33] have demonstrated remarkable performance in graph classification across various domains, including chemical molecules, social networks, and transaction graphs. This success is primarily attributed to the powerful representation learning of GNNs, which incorporates graph structure and encodes it into representations through an end-to-end process. Consequently, it is essential to highlight the critical aspects of the input graph while filtering out trivial components [45, 70, 71, 75]. For instance, when classifying the mutagenic properties of a molecular graph [50], GNNs should focus on functional groups (i.e., nitrogen dioxide (NO_2)) rather than irrelevant patterns (i.e., carbon rings) [9, 15, 16, 86]. Similarly, when detecting fraud in a transaction network, malicious behaviors or user coalitions are more informative than benign features [19–21].

To identify critical parts in graphs, subsequent studies [18, 32, 62, 68, 84] adopt the "learning to attend" paradigm [67, 78], which maximizes the mutual information between the attended graph and the ground-truth label to find the attended graph that optimizes the predictive performance. This paradigm consists of two research lines:

— Attention-based methods [4, 32, 44, 66, 68]. These methods often employ attention modules for nodes or edges to locate the attended graphs. These attention modules function as soft masks, identifying the importance of each edge and node for the final representations and predictions.
— Pooling-based methods [18, 37, 84, 89]. These methods mainly use hard masks to select a subset of nodes or edges as the attended graphs for information propagation.

These attended graphs aim to capture features that contribute to minimizing training loss, rather than differentiating between causal and non-causal effects.

Unfortunately, recent studies [8, 17, 22] have revealed that current attention and pooling learning methods are susceptible to exploiting shortcut features for predictions. These shortcuts often arise from data selection biases, noisy features, or trivial patterns in graphs, which, although non-causal, are discriminative in training data. The presence of these shortcuts allows models to capture them and complete classification tasks without learning causal features. For instance, instead of investigating the causal effect of functional groups, attended graphs may rely on "carbon rings" as cues for the "mutagenic" class, because most training "mutagenic" molecules occur in the context of "carbon rings." While these correlations represent statistical relationships inherent in the training data and benefit **in-distribution (ID)** test evaluations, they inevitably result in significant performance declines in **out-of-distribution (OOD)** test data that differ from the training distribution. Using molecule classification as an example, when most test "non-mutagenic" molecules appear in the context of "carbon rings," the attended graphs mislead GNNs into predicting "mutagenic." Since it is often unrealistic to assume that test data conform to the training distribution

in real-world scenarios, the poor generalization of these methods obstructs their application in critical situations.

To address this issue, we first examine the decision-making process of GNNs for graph classification from a causal perspective, which outlines the relationships among causal features, shortcut features, and predictions. Under our causal assumption illustrated in Figure 1, the shortcut feature acts as a confounder [54], creating a backdoor path [53] that leads to a spurious correlation between the causal feature and predictive label, such as misclassifying "non-mutagenic" molecules with "carbon rings" as "mutagenic" molecules. Therefore, mitigating the confounding effect holds promise for leveraging causal features while filtering out shortcut patterns, ultimately improving the generalization.

Towards this end, we propose CAL+ framework, an improved version of our existing **Causal Attention Learning (CAL)** method [63]. Inspired by backdoor adjustments in causal theory [53, 54], CAL+ aims to maximize the causal effect of the attended graph on predicting labels while minimizing the confounding effect of shortcut features. First, we apply attention modules to generate estimations of causal and shortcut features from input graphs. Then, we combine each causal estimation with various shortcut estimations, ensuring these combinations maintain stable predictions. To guarantee the diversity of shortcut features in the combined data, we employ a memory bank to collect them. Simultaneously, we use a prototype strategy to preserve the consistency of intra-class causal features, constraining the representations of estimated causal features. Consequently, causal estimations are encouraged to approach the causal features in the graph (e.g., nitrogen dioxide), while their complements target the shortcut features (e.g., carbon rings). CAL+ promotes invariant relationships between causal patterns and predictions, regardless of changes in shortcut parts or distribution shifts. In summary, this article offers the following key contributions:

— We highlight the poor generalization issue in current attention- and pooling-based GNNs for graph classification. From the causal perspective, we ascribe such an issue to the confounding effect of the shortcut features.
— We introduce a novel learning framework, CAL+, for graph classification. Inspired by backdoor adjustments in causal theory, CAL+ enables GNNs to utilize causal features for predictions while minimizing the confounding effect of shortcut features.
— We perform comprehensive experiments on OOD benchmarks and compare our approach with various baselines. The results validate the effectiveness of CAL+. Additionally, detailed visualizations and analyses demonstrate the interpretability and rationality of our method.

Building upon previous work, CAL [63], we detail the main differences in the following three aspects:

— **Enhanced Methodology.** The CAL+ model offers two key improvements over the base CAL model. (1) Memory Bank Module. Backdoor adjustment requires the combination of each causal feature with a stratification of the confounder, which involves traversing various types of shortcut features in the dataset. In CAL, the types of shortcut features are constrained to each mini-batch, substantially limiting their diversity and making the backdoor adjustment less effective. Increasing the batch size only offers limited diversity while significantly raising training memory consumption. To address this issue, we utilize a memory bank to store and sample shortcut features, effectively increasing their diversity for adequate combinations. (2) Prototype Module. Causal features of the same class should exhibit similar representations [8, 48, 90]. However, CAL does not account for intra-class representation relationships, resulting in inaccurate or unstable causal feature estimations. To resolve this, we define a prototype for each class and encourage causal representations to be close to

their respective class-wise prototypes. This strategy enhances the consistency of intra-class causal features and ensures more accurate and stable estimations.

— **Fuller Explanations and Discussions.** Inspired by recent studies [6, 7, 46], we delve into the foundational assumptions regarding the identifiability of causal features. These assumptions are critical, as they delineate the conditions essential for the effective application of our method. Furthermore, we present the **Structural Causal Models (SCMs)** established in related studies [8, 13, 22, 75] and compare them with our approach, highlighting both differences and connections. Finally, we also include the time complexity analysis of our method.

— **Comprehensive Evaluations.** We have made significant revisions to the experimental section compared to previous work. (1) Additional baselines. The baselines in CAL are insufficient, as it only compares mainstream GNNs for graph classification, such as attention- or pooling-based GNNs. In contrast, we include 12 recent baselines in this work, comprising general generalization methods, graph generalization methods, and graph data augmentation methods. (2) More OOD datasets. The datasets in CAL are insufficient. Besides synthetic datasets, CAL only employs commonly used datasets such as NCI1, COLLAB [50], and so on. However, these datasets exhibit minimal distribution shifts. Consequently, we remove these experiments and incorporate new OOD datasets. These datasets are generated based on specific graph characteristics, presenting evident distribution shift issues. In addition, we have added more detailed and comprehensive analyses, including ablation studies, hyperparameter sensitivity analysis, running-time statistics, and visualization experiments.

2 PRELIMINARIES

2.1 Notations

We denote a graph by $g = \{\mathbf{A}, \mathbf{X}\}$ with the node set $\mathcal{V}$ and edge set $\mathcal{E}$. Let $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times \mathrm{F}}$ be the node feature matrix, where $\mathbf{x}_i = \mathbf{X}[i,:]$ is the F-dimensional attribute vector of node $v_i \in \mathcal{V}$. We use the adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ to delineate the whole graph structure, where $\mathbf{A}[i,j] = 1$ if edge $(v_i, v_j) \in \mathcal{E}$, otherwise $\mathbf{A}[i,j] = 0$. We define $\mathrm{GConv}(\cdot)$ as a GNN layer module and denote the node representation matrix by $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times d}$, whose ith row $\mathbf{h}_i = \mathbf{H}[i,:]$ denotes the representation of node v_i.

2.2 Attention Mechanism in GNNs

In GNNs, attention can be defined over edges or nodes. For edge-level attentions [4, 32, 39, 66, 68], they utilize weighted message passing and aggregation to update node representations $\mathbf{H}'$:

$$\mathbf{H}' = \mathrm{GConv}\left(\mathbf{A} \odot \mathbf{M}_a, \mathbf{H}\right), \tag{1}$$

where $\mathbf{M}_a \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ denotes the attention matrix that is often derived from trainable parameters and node representations. For node-level attention, several studies [34, 37, 44] define the self-attention mask to select the most attentive node representations:

$$\mathbf{H}' = \mathrm{GConv}\left(\mathbf{A}, \mathbf{H} \odot \mathbf{M}_x\right), \tag{2}$$

where $\mathbf{M}_x \in \mathbb{R}^{|\mathcal{V}| \times 1}$ represents the node-level attentions, which can be generated by a network (e.g., GNNs or MLPs); $\odot$ is the broadcasted element-wise product. Hereafter, we can make further pooling operation [37] for the output node representations $\mathbf{H}^{out}$ and summarize the graph representation $\mathbf{h}_g$ for graph g via the readout function $f_{\mathrm{readout}}(\cdot)$. Then, we use a classifier Φ to project the graph representation into a probability distribution $\mathbf{z}_g$:

$$\mathbf{h}_g = f_{\mathrm{readout}}\left(\{\mathbf{h}_i^{out} | i \in \mathcal{V}\}\right), \quad z_g = \Phi(\mathbf{h}_g). \tag{3}$$

These methods follow the paradigm of "learning to attend" by minimizing the following empirical risk:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{|\mathcal{D}_{tr}|} \sum_{g \in \mathcal{D}_{tr}} \mathbf{y}_g^\top \log(\mathbf{z}_g), \tag{4}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss over the training dataset $\mathcal{D}_{tr}$, and $\mathbf{y}_g$ is the ground-truth label vector of g. This learning strategy, known as **Empirical Risk Minimization (ERM)**, heavily relies on the statistical correlations between input graphs and labels. Consequently, these methods inevitably capture noncausal shortcut features for making predictions.

2.3 OOD Issue in Graph Classification

In graph classification, we typically train a GNN model using a training dataset $\mathcal{D}_{tr}$ and predict labels in a test dataset $\mathcal{D}_{te}$. $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$ are independently sampled from the training distribution P_{tr} and test distribution P_{te}, respectively. When $P_{tr} = P_{te}$, ERM-based models usually perform well, even though they may rely on shortcuts for predictions. However, the test environment is often unknown and unstable in real-world scenarios, resulting in $P_{tr} \neq P_{te}$, which leads to distribution shift or **out-of-distribution (OOD)** issues. For instance, in molecular property prediction tasks, we generally use past molecules as training data, hoping that the model can predict properties of molecules in new environments in the future. In such OOD evaluation scenarios, shortcuts may not often exist, leading to a significant performance degradation.

2.4 *do*-Calculus in Causal Inference

Causal inference [26, 27, 52–55] enables researchers to understand and predict the effects of interventions in various systems. A fundamental concept within this domain is "*do*-calculus" [53–55]. It provides a formal language for expressing and resolving queries about causal relationships. This operator is used to denote a hypothetical intervention where one or more variables in a system are set to specific values, irrespective of their original causal relationships. The calculus consists of a set of rules that allow for the manipulation of probability expressions involving the *do*-operator. These rules enable the transformation of causal questions into statistical estimations, a process pivotal in discerning causal effects from observational data. A key aspect of applying *do*-calculus is the identification of appropriate variables to adjust for confounding. This leads to the concept of the "backdoor criterion." The criterion provides a method to identify a set of variables, known as a backdoor adjustment set, which, when controlled for, can eliminate the bias due to confounding. The backdoor criterion essentially ensures that the adjusted causal effect is not influenced by backdoor paths (indirect paths) between the intervention and the outcome. The process of backdoor adjustment involves conditioning on the selected set of variables. This is typically done through statistical techniques such as stratification or regression. By controlling for these backdoor paths, the causal effect of the intervention on the outcome is isolated, allowing for more accurate estimation of the causal relationship. In machine learning, *do*-calculus and backdoor adjustments play a crucial role in the design and interpretation of models, particularly in observational studies where randomized control trials are not feasible. We provide more discussion of various applications of causal inference in Section 5.3.

3 METHODOLOGY

In this section, we first analyze the GNN learning from the perspective of causality. From our causal assumption, we identify the shortcut feature as a confounder. Then, we propose the causal attention learning strategy to alleviate the confounding effect.

3.1 A Causal View on GNNs

We take a causal look at the GNN modeling and construct a **Structural Causal Model (SCM)**[54] in Figure 1. It presents the causalities among five variables: graph data G, causal feature C, shortcut feature S, graph representation R, and prediction or ground-truth label Y. Note that our SCM describes the model's forward process, so Y can represent both the prediction and ground-truth label, as they are optimized to be the same through the training process. The link from one variable to another indicates the cause-effect relationship: cause → effect. We list the following explanations for SCM:

- $C \leftarrow G \rightarrow S$. The variable C denotes the causal feature that truly reflects the intrinsic property of the graph data G. While S represents the shortcut feature that is usually caused by the data biases or trivial patterns. Since C and S naturally coexist in graph data G, these causal effects are established.
- $C \rightarrow R \leftarrow S$. The variable R is the representation of the given graph data G. To generate R, the conventional learning strategy takes the shortcut feature S and causal feature C as input to distill discriminative information.
- $R \rightarrow Y$. The ultimate goal of graph representation learning is to predict the properties of the input graphs. The classifier will make prediction Y based on the graph representation R.

Scrutinizing this SCM, we recognize a backdoor path between C and Y, i.e., $C \leftarrow G \rightarrow S \rightarrow R \rightarrow Y$, wherein the shortcut feature S plays a confounder role between C and Y. Even if C has no direct link to Y, the backdoor path will cause C to establish a spurious correlation with Y, e.g., making wrong predictions based on shortcut feature S instead of causal feature C. Hence, it is crucial to cut off the backdoor path and make the GNN exploit causal features.

3.2 Backdoor Adjustment

We have realized that shielding the GNNs from the confounder S is the key to exploiting causal features. Instead of modeling the confounded $P(Y|C)$ in Figure 1, we should achieve the graph representation learning by eliminating the backdoor path. But how to achieve this? Fortunately, causal theory [53, 54] provides us with a feasible solution: We can exploit the "*do*-calculus" on the variable C to remove the backdoor path by estimating $P_m(Y|C) = P(Y|do(C))$. It needs to stratify the confounder S between C and Y. Therefore, we can obtain the following three essential conclusions:

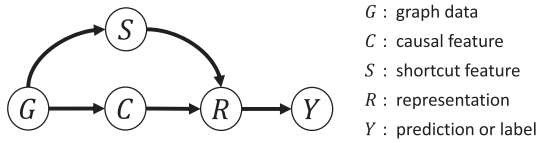


Fig. 1. Structural causal model for graph classification. We use Y for prediction or ground-truth label, as they are optimized to be the same.

- The marginal probability $P(S = s)$ is invariant under the intervention, because the shortcut feature will not be affected by cutting off the backdoor path. Thus, we have $P(s) = P_m(s)$.
- The conditional probability $P(Y|C, s)$ is invariant, because YâĂŹs response to C and S has nothing to do with the causal effect between C and S. Then, we can get: $P_m(Y|C, s) = P(Y|C, s)$.
- Obviously, the variables C and S are independent under the intervention, which we have: $P_m(s|C) = P_m(s)$.

Based on the above conclusions, we have:

$$\begin{aligned} P(Y|do(C)) &= P_m(Y|C) \\ &= \sum_{s\in\mathcal{T}} P_m(Y|C,s)P_m(s|C) \quad (BayesRule) \\ &= \sum_{s\in\mathcal{T}} P_m(Y|C,s)P_m(s) \quad (Independency) \\ &= \sum_{s\in\mathcal{T}} P(Y|C,s)P(s), \end{aligned} \tag{5}$$

where $\mathcal{T}$ denotes the confounder set; $P(Y|C,s)$ is the conditional probability given the causal feature C and confounder s; $P(s)$ is the prior probability of the confounder. Equation (5) is usually called backdoor adjustment [53], which is a powerful tool to eliminate the confounding effect. However, there exist two challenges for implementing Equation (5): (i) The confounder set $\mathcal{T}$ is commonly unobservable and hard to obtain. (ii) Due to the discrete nature of graph data, it seems difficult to directly manipulate the graph data, conditioning on domain-specific constraints (e.g., valency rules in molecule graphs). In Section 3.4.3, we will introduce a simple yet effective solution to overcome these issues.

3.3 Causal and Trivial Attended-graph

Given a graph $g = \{\mathbf{A}, \mathbf{X}\}$, we formulate the soft masks on the graph structure and node feature as $\mathbf{M}_a \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ and $\mathbf{M}_x \in \mathbb{R}^{|\mathcal{V}|\times 1}$, respectively. Wherein, each element of the masks indicates the attention score relevant to the task of interest, which often falls into the range of $(0, 1)$. Given an arbitrary mask $\mathbf{M}$, we define its complementary mask as $\overline{\mathbf{M}} = \mathbf{1} - \mathbf{M}$, where $\mathbf{1}$ is the all-one matrix. Therefore, we can divide the full graph g into two attended-graphs: $g_1 = \{\mathbf{A} \odot \mathbf{M}_a, \mathbf{X} \odot \mathbf{M}_x\}$ and $g_2 = \{\mathbf{A} \odot \overline{\mathbf{M}}_a, \mathbf{X} \odot \overline{\mathbf{M}}_x\}$.

With the inspection on the data-generating process, recent studies [34, 45, 75, 83] argue that the label of a graph is usually determined by its causal part. Considering a molecular graph, its mutagenic property relies on the existence of relevant functional groups [70]; taking the digit image in the form of superpixel graph as another example, the coalition of digit-relevant nodes determines its label. Formally, given a graph g, we define the attended graph collecting all causal features as the causal attended-graph g_c, while the counterpart forms the trivial attended-graph g_t. However, the ground-truth attended-graph is usually unavailable in real-world applications. Hence, we aim to capture the causal and trivial attended-graph from the full graph by learning the masks: $g_c = \{\mathbf{A}\odot\mathbf{M}_a, \mathbf{X}\odot\mathbf{M}_x\}$ and $g_t = \{\mathbf{A}\odot\overline{\mathbf{M}}_a, \mathbf{X}\odot\overline{\mathbf{M}}_x\}$. Learning to identify causal attended-graphs not only guides the representation learning of GNNs, but also answers "What knowledge does the GNN use to make predictions?," which is crucial to the applications on explainability, privacy, and fairness.

3.4 Causal Attention Learning

To implement the backdoor adjustment, we propose the **Causal Attention Learning Plus (CAL+)** framework.

3.4.1 Attention Score Generation. Towards effective causal intervention, it is necessary to separate the causal and shortcut features from the full graphs. To this end, we hire attention modules, which yield two branches for the causal and trivial proposals. Given a GNN-based encoder $f(\cdot)$ and a graph $g = \{\mathbf{A}, \mathbf{X}\}$, we can obtain the node representations:

$$\mathbf{H} = f(\mathbf{A}, \mathbf{X}). \tag{6}$$

Then, we adopt two MLPs, $\text{MLP}_{\text{node}}(\cdot)$ and $\text{MLP}_{\text{edge}}(\cdot)$, to generate the attention scores (i.e., soft-masks) from two orthogonal perspectives: node-level and edge-level. For node v_i and edge (v_i, v_j), we can obtain:

$$\alpha_{c_i}, \alpha_{t_i} = \sigma\big(\text{MLP}_{\text{node}}(\mathbf{h}_i)\big), \tag{7}$$

$$\beta_{c_{ij}}, \beta_{t_{ij}} = \sigma\big(\text{MLP}_{\text{edge}}(\mathbf{h}_i || \mathbf{h}_j)\big), \tag{8}$$

where $\sigma(\cdot)$ is softmax function, $||$ denotes concatenation operation; $\alpha_{c_i}, \beta_{c_{ij}}$ represent the node-level attention score for node v_i and edge-level attention score for edge (v_i, v_j) in causal attended-graph; analogously, $\alpha_{t_i}, \beta_{t_{ij}}$ are for trivial attended-graph. Note that $\alpha_{c_i} + \alpha_{t_i} = 1$, and $\beta_{c_{ij}} + \beta_{t_{ij}} = 1$. These attention scores indicate how much the model pays attention to each node or edge in the corresponding attended-graph. Now, we can construct the soft masks $\mathbf{M}_x$, $\overline{\mathbf{M}}_x$, $\mathbf{M}_a$, and $\overline{\mathbf{M}}_a$ based on the attention scores α_{c_i}, α_{t_i}, $\beta_{c_{ij}}$, and $\beta_{t_{ij}}$, respectively. Finally, we can decompose the graph g into the initial causal and trivial attended-graphs: $g_c = \{\mathbf{A} \odot \mathbf{M}_a, \mathbf{X} \odot \mathbf{M}_x\}$ and $g_t = \{\mathbf{A} \odot \overline{\mathbf{M}}_a, \mathbf{X} \odot \overline{\mathbf{M}}_x\}$.

3.4.2 Causal Disentanglement. Until now, we have distributed the attention scores at the granularity of nodes and edges to create the initial attended-graphs. Now, we need to make the causal and trivial attended-graphs to capture the causal and shortcut features from the input graphs, respectively. Specifically, we adopt two GNN layers to obtain the representations of attended-graphs and make predictions via readout function and classifiers:

$$\mathbf{h}_{g_c} = f_{\text{readout}}\big(\text{GConv}_c\big(\mathbf{A} \odot \mathbf{M}_a, \mathbf{X} \odot \mathbf{M}_x\big)\big), \quad \mathbf{z}_{g_c} = \Phi_c(\mathbf{h}_{g_c}), \tag{9}$$

$$\mathbf{h}_{g_t} = f_{\text{readout}}\big(\text{GConv}_t\big(\mathbf{A} \odot \overline{\mathbf{M}}_a, \mathbf{X} \odot \overline{\mathbf{M}}_x\big)\big), \quad \mathbf{z}_{g_t} = \Phi_t(\mathbf{h}_{g_t}). \tag{10}$$

To encourage causal attended-graphs to approximate causal features, we need to further constrain their representations. By examining the properties of causal features, we can observe that causal features of the same class should originate from the same prototypical features. For instance, water-soluble molecules tend to have "-OH" functional groups, while acidic molecules usually exhibit "-COOH"-based patterns. A prevalent approach is to learn a class-specific context that provides a global overview of each class, which is referred to as the prototype vector. Specifically, for graphs in class i, we register the prototype vector $\mathbf{p}_i^{(t-1)}$ at training step $t-1$. Then, we update the prototype $\mathbf{p}_i^{(t)}$ in the next step by calculating the similarity between the each causal representation $\mathbf{h}_{g_c,k}^{(t)}$ and $\mathbf{p}_i^{(t-1)}$. Formally, given K causal features in each class at training step t, we have:

$$\begin{aligned} s_k^{(t)} = \text{cosine}\Big(\mathbf{h}_{g_c,k}^{(t)}, \mathbf{p}_i^{(t-1)}\Big), \quad w_k^{(t)} = \frac{\exp(s_k^{(t)}/\tau)}{\sum_{k=1}^{K} \exp(s_k^{(t)}/\tau)}, \\ \mathbf{p}_i^{(t)} = \sum_{k=1}^{K} w_k^{(t)} \cdot \mathbf{h}_{g_c,k}^{(t)}, \end{aligned} \tag{11}$$

where τ denotes the temperature parameter; cosine refers to the cosine similarity. In particular, we initialize $\mathbf{p}_i^{(0)}$ to be the average of the vectors at the initial training step $t = 0$. Then, we define the following loss to ensure the consistency of causal representations within class i:

$$\ell_{\text{pro}}(g) = -\log \frac{\exp(\mathbf{h}_{g_c} \cdot \mathbf{p}_i)}{\sum_{i=1}^{C} \exp(\mathbf{h}_{g_c} \cdot \mathbf{p}_i)}, \tag{12}$$

where C is the class number. On the one hand, we also need to utilize supervisory signals to ensure the correctness of predictions based on causal estimations. So, we should classify their representations to ground-truth labels. On the other hand, the trivial attended-graph aims to approach the trivial patterns that are unnecessary for classification. We push its prediction evenly to all

categories, which means that the distribution of prediction is required to conform to the prior distribution of uniform. Hence, we define the following losses for causal and trivial estimations:

$$\ell_{\text{sup}}(g) = -\mathbf{y}_g^\top \log(\mathbf{z}_{g_c}),\ \ \ell_{\text{unif}}(g) = \text{KL}(\mathbf{y}_{\text{unif}}, \mathbf{z}_{g_t}), \tag{13}$$

where KL denotes the KL-Divergence, $\mathbf{y}_{\text{unif}}$ represents the uniform prior distribution. Then, we can define our objective for causal disentanglement as:

$$\mathcal{L}_{\text{dis}} = \mathbb{E}_{(g,\mathbf{y}_g)\sim\mathcal{D}_{\text{tr}}}[\ell_{\text{sup}}(g) + \rho_1 \ell_{\text{pro}}(g) + \rho_2 \ell_{\text{unif}}(g)], \tag{14}$$

where ρ_1 and ρ_2 are hyperparameters that determine the strength of disentanglement for causal features and trivial features, respectively. By optimizing Equation (14), we can approximately disentangle causal and trivial parts from the data, and through subsequent causal intervention, we will be able to distinguish them more accurately. Note that prior efforts [34, 45, 70, 71, 83] have shown that the mutual information between the causal part and label is greater than that between the full graph and label, due to the widespread trivial patterns or noise. Hence, the proposed disentanglement objective will not make the captured causal attended-graph converge to the full graph (noiseless full graph is a special case), which is not an optimal solution. See Section 4.5 for more supporting evidence and analyses.

3.4.3 Causal Intervention. As shown in Equation (5), one promising solution to alleviating the confounding effect is the backdoor adjustment—that is, stratifying the confounder and pairing the target causal attended-graph with every stratification of trivial attended-graph to compose the "intervened graphs." However, due to the irregular graph data, it is impossible to make the intervention on data-level, e.g., changing a graph's trivial part to generate a counterfactual graph data. Towards this end, we make the implicit intervention on representation-level and propose the following loss guided by the backdoor adjustment:

$$\mathbf{z}_{g'} = \Phi(\mathbf{h}_{g_c} + \mathbf{h}_{g_{t'}}), \tag{15}$$

$$\mathcal{L}_{\text{cau}} = -\frac{1}{|\mathcal{D}_{tr}| \cdot |\hat{\mathcal{T}}|} \sum_{g\in\mathcal{D}_{tr}} \sum_{t'\in\hat{\mathcal{T}}} \mathbf{y}_g^\top \log(\mathbf{z}_{g'}), \tag{16}$$

where $\mathbf{z}_{g'}$ is the prediction from a classifier Φ on "implicit intervened graph" g'; $\mathbf{h}_{g_c}$ is the representation of causal attended-graph g_c derived from Equation (9); while $\mathbf{h}_{g_{t'}}$ is the representation of stratification $g_{t'}$ obtained via Equation (10); $\hat{\mathcal{T}}$ is the estimated stratification set of the trivial attended-graph, which need to collect the appearing trivial features from training data. An ideal backdoor adjustment requires traversing various types of shortcut features that appear in the dataset. However, during training process, model parameters are usually updated through mini-batch. The types of shortcut features are constrained to each mini-batch, substantially limiting their diversity and making the backdoor adjustment less effective. Increasing the batch size only offers limited diversity while significantly raising training memory consumption.

To address this issue, we utilize memory bank strategy [24, 77] to store shortcut features. Specifically, we define $\mathcal{V} = \{\mathbf{v}_i\}$ as a memory bank. In each training step, we can feed more data (i.e., N times the batch size) into the model and collect the representations of shortcut features $\{\mathbf{h}_{g_t}\}$. Then, we update and store these representations into $\mathcal{V} = \{\mathbf{h}_{g_t} \rightarrow \mathbf{v}_i\}$. In the intervention stage, we can implement Equation (16) by combining the causal features with representations of various shortcut features in $\mathcal{V}$. We can find that this storage strategy can greatly improve the diversity of shortcut features in the process of causal intervention, thus making backdoor adjustment more effective. Finally, the objective of CAL+ can be defined as:

$$\mathcal{L} = \mathcal{L}_{\text{dis}} + \lambda \mathcal{L}_{\text{cau}}, \tag{17}$$

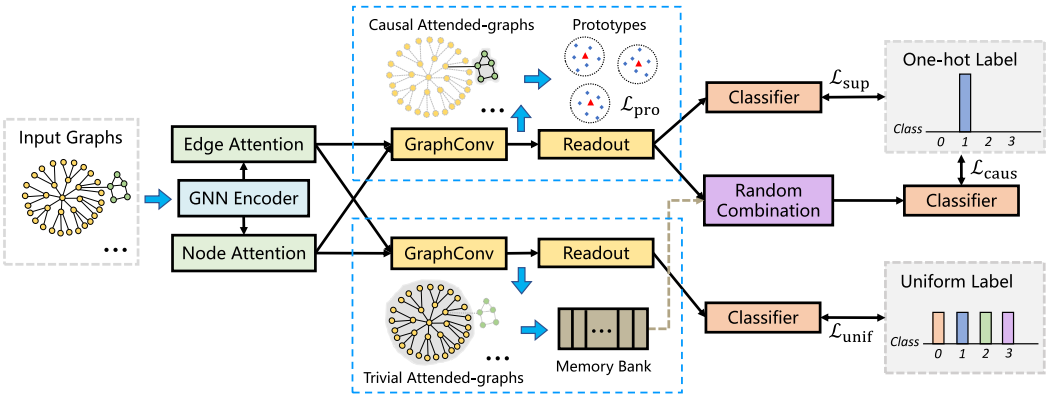


Fig. 2. The overview of the proposed CAL+ framework.

where λ is the hyperparameter that determines the strength of causal intervention. The overview of the proposed CAL+ is depicted in Figure 2.

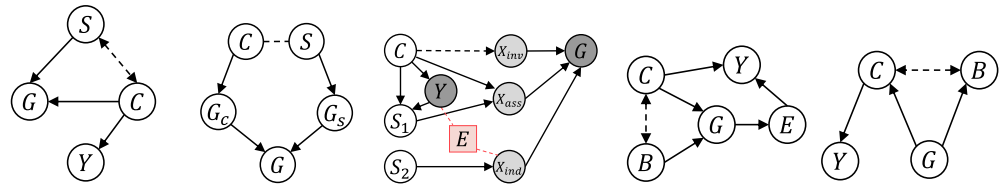
3.5 Assumptions and More Discussions

In this section, we delve into the assumptions underpinning our approach, contrast our structural causal model with existing literature, and examine the time complexity of our proposed algorithm.

3.5.1 Potential Assumptions. Recent studies [6, 7, 46] highlight that prevailing methods in invariant learning or causal learning are predicated on certain critical assumptions. The absence of these assumptions undermines the efficacy of existing invariance learning techniques. Notably, Assumption 3.3 from Reference [6] asserts that invariant subgraphs must maintain invariance across diverse environments $P^{e_1}(Y|G_c) = P^{e_2}(Y|G_c)$. However, this invariance falters in specific environments where spurious subgraphs fail to satisfy invariance $P^{e_1}(Y|G_s) \neq P^{e_2}(Y|G_s)$. This assumption is crucial for the model to effectively differentiate between causal subgraphs and those influenced by environmental factors. Furthermore, Assumption 3.5 from Reference [6] posits that $H(C|Y) \neq H(S|Y)$ in all environments, underscoring a consistency requirement in the correlation strengths between invariant and spurious subgraphs in relation to their labels. In our work, we align with the assumption similar to that in CIGA [8], which is $H(G_c|Y) < H(G_s|Y)$. This implies a stronger correlation between invariant features and labels as compared to the correlation between spurious features and labels. Inspired from References [6, 8, 13], we now summarize and refine our assumptions as follows:

Assumption 1 (Identifiability of Causal Feature). *Consider $\mathcal{E}_{tr}$ as the set of training environments, and let G_s and G_c represent the shortcut and causal feature variables of graph G, respectively. For any given shortcut feature G_s, causal feature G_c, and label Y in graph G, they adhere to the following conditions: (1) Variation sufficiency: There exist $e_1, e_2 \in \mathcal{E}_{tr}$ for which $P^{e_1}(Y|G_s) \neq P^{e_2}(Y|G_s)$, and for all $e_1, e_2 \in \mathcal{E}_{tr}$, it holds that $P^{e_1}(Y|G_c) = P^{e_2}(Y|G_c)$. (2) Correlation strength: Across all environments in $\mathcal{E}_{tr}$, $H(G_c|Y) < H(G_s|Y)$.*

3.5.2 Structural Causal Model. We delve into the intricacies of various **structural causal models (SCMs)** as delineated in related studies, focusing on their distinct characteristics and interconnections. As shown in Figure 3, we showcase the SCMs from DIR [75], CIGA [8], GOOD [22], DisC [13]. In DIR's context, S and C symbolize shortcut and causal features, respectively, playing a pivotal role in shaping the graph data G. Notably, the label Y is exclusively influenced by the causal feature C. In contrast, CIGA interprets S and C as latent variables for shortcut and causal features,



(a) SCM in DIR [75] (b) SCM in CIGA [8] (c) SCM in GOOD [22] (d) SCM in DisC [13] (e) SCM in DisC [13]

Fig. 3. A comparison of various types of SCMs in related studies.

respectively, with G_c and G_s depicting their manifestations in the graph space. GOOD discusses a more specific and complex data generation process and further divides S to S_1 and S_2. In the DisC (Figure 3(d)), C and B denote causal and biased features, while E symbolizes graph representation. The SCM, as illustrated in Figure 3(e), aims to segregate C and B within the data G, striving to sever their correlation and leverage C for predictive analysis. Notably, SCMs in Figures 3(a), (b), and (c) primarily describe data generation, whereas 3(d) encompasses both data generation and model prediction processes, and 3(e) focuses solely on model prediction. Our proposed SCM aligns with this framework, concentrating on the model's prediction mechanism. It acknowledges the presence of both causal and shortcut features within the data, which the model simultaneously assimilates to derive a representation of the graph data. This representation then forms the basis for the model's ultimate prediction. In summary, for graph data generation, our framework resonates with the SCMs exemplified in 3(a) and (b). Regarding the model prediction process, we align more closely with the paradigms set forth in 3(d) and (e).

3.5.3 Time Complexity Analysis. We analyze the time complexity of the proposed model. First, we define the average numbers of nodes and edges per graph as $|\mathcal{V}|$ and $|\mathcal{E}|$, respectively. Let B denote the batch size for each training iteration, l_f, l_a, l_g denote the numbers of layers in the GNN encoder, attention modules and GNN modules $\text{GConv}_c(\cdot)$, $\text{GConv}_t(\cdot)$, respectively. Let d_f, d_a, and d_g denote the dimensions of the hidden layers in the GNN encoder, attention modules and GNN modules, respectively. For the GNN encoder, the time complexity is $O(B\times(l_f|\mathcal{E}|d_f))$. For the mask generation process, the time complexity is $O(B \times (l_a(|\mathcal{V}| + |\mathcal{E}|)d_g))$. For the graph representation generation process, the time complexity is $O(2B \times (l_g|\mathcal{E}|d_g))$. For simplicity, we assume $l = l_f = l_a = l_g$ and $d = d_f = d_a = d_g$. Hence, the time complexity of a forward propagation is $O(Bld(4|\mathcal{E}| + |\mathcal{V}|))$.

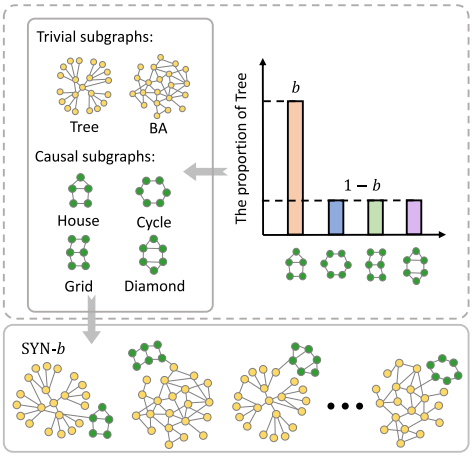


Fig. 4. Illustration of the synthetic dataset SYN-b.

4 EXPERIMENTS

To verify the superiority and effectiveness of the proposed CAL+, we conduct experiments to answer the following research questions:

— **RQ1:** How effective is the proposed CAL+ framework in alleviating the **out-of-distribution (OOD)** issue?
— **RQ2:** How does CAL+ perform compared to state-of-the-art methods?

— **RQ3:** For the different components in CAL+, what are their roles and impacts on performance?
— **RQ4:** Does CAL+ capture the causal attended-graphs with significant patterns and insightful interpretations?

4.1 Experimental Settings

4.1.1 Datasets. We conduct experiments on both synthetic datasets and real-world datasets.

Synthetic datasets: Following Reference [83], we create the synthetic dataset for graph classification, which contains a total of 8,000 samples with 4 classes and keeps balance (2,000 samples) for each class. As shown in Figure 4, each sample consists of two parts: causal subgraph and trivial subgraph. The task is to predict the type of the causal part in the whole graph. For simplicity, we choose the "House" class to define the bias-level:

$$b = \frac{\#\text{Tree-House}}{\#\text{House}}, \tag{18}$$

where #Tree-House denotes the number of "House" causal subgraphs with the "Tree" trivial subgraphs, and #House presents the number of graphs in the "House" class, which is 2,000. We set the proportion of "Tree" in the other three classes to $1-b$. Obviously, for the unbiased dataset, $b = 0.5$. We abbreviate the synthetic dataset with bias-level b as SYN-b. We keep the same bias-level on the training/validation set and keep the testing set unbiased. Additionally, we conduct experiments on the Motif dataset [22], a synthetic dataset inspired by Spurious-Motif [83]. Like SYN-b, Motif consists of a causal subgraph (i.e., motif) and a trivial subgraph (i.e., base graph). Each graph is generated by connecting a base graph (wheel, tree, ladder, star, or path) and a motif (house, cycle, or crane), with the label determined solely by its motif. In accordance with Reference [22], we use the base graph and size (i.e., node number) to create the concept shift. Although our work focuses on graph classification, we also use the idea of ego-graph [74] to extend our method to node classification task. We use the node classification datasets in Reference [43], including Citeseer and Amazon-Photo, to conduct experiments.

Real-world datasets: We carry out experiments on three real-world datasets: Molhiv, Molbbbp, and CMNIST, sourced from Graph OOD datasets [22] and OGB datasets. Following Reference [22], we select four different types of shortcut features (i.e., color, size, and scaffold) to create the concept shift. Specifically, Molhiv and Molbbbp are real-world molecular datasets adapted from MoleculeNet [76], with nodes representing atoms and edges representing chemical bonds. CMNIST contains graphs transformed from handwritten digits using superpixel techniques [1], and we utilize color to create the concept shift.

4.1.2 Baselines. To verify the superiority of CAL+, we adopt the following prevalent GNNs and diverse generalization solutions as baselines:

— **Attention-based GNNs:** GAT [68], GATv2 [4], SuperGAT [32], GlobalAttention [44], AGNN [66].
— **Pooling-based GNNs:** SortPool [89], DiffPool [84], Top-k Pool [18], SAGPool [37].
— **Other GNNs:** GCN [33], GIN [79].
— **General Generalization Algorithms:** ERM, IRM [2], GroupDRO [60], VREx [36].
— **Graph Generalization Algorithms:** DIR[75], OOD-GNN [40], StableGNN [14], CIGA [8], Disc [13].
— **Graph Data Augmentation:** DropEdge [58], FLAG [35], M-Mixup [73], G-Mixup [23], GREA [47].

Table 1. Test Accuracy (%) of Graph Classification on Synthetic Datasets with Diverse Biases

Method	SYN-0.1	SYN-0.3	Unbiased	SYN-0.7	SYN-0.9
GATv2	87.25 (↓ 7.37%)	92.19 (↓ 2.12%)	94.19	93.31 (↓ 0.93%)	90.62 (↓ 3.79%)
SuperGAT	83.81 (↓ 12.75%)	91.94 (↓ 4.29%)	96.06	88.50 (↓ 7.89%)	82.81 (↓ 13.79%)
GlobalAtt	87.19 (↓ 10.40%)	93.75 (↓ 3.66%)	97.31	94.62 (↓ 2.76%)	91.50 (↓ 5.97%)
AGNN	84.56 (↓ 11.69%)	93.06 (↓ 2.81%)	95.75	94.81 (↓ 0.98%)	88.12 (↓ 7.97%)
DiffPool	82.28 (↓ 8.69%)	88.02 (↓ 2.32%)	90.11	88.83 (↓ 1.42%)	84.50 (↓ 6.23%)
SortPool	80.70 (↓ 14.24%)	92.33 (↓ 1.88%)	94.10	92.14 (↓ 2.08%)	90.35 (↓ 3.99%)
Top-*k* Pool	84.31 (↓ 11.81%)	93.53 (↓ 2.17%)	95.60	94.44 (↓ 1.21%)	88.02 (↓ 7.93%)
SAGPool	88.08 (↓ 7.82%)	90.86 (↓ 4.91%)	95.55	92.22 (↓ 3.49%)	83.99 (↓ 12.10%)
GCN	84.94 (↓ 6.60%)	89.38 (↓ 1.72%)	90.94	90.25 (↓ 0.76%)	86.00 (↓ 5.43%)
GCN + CAL	89.38 (↓ 6.03%)	93.50 (↓ 1.70%)	95.12	95.06 (↓ 0.06%)	93.31 (↓ 1.90%)
GCN + CAL+	90.19 (↓ 5.96%)	95.12 (↓ 0.82%)	95.91	95.88 (↓ 0.03%)	93.06 (↓ 2.97%)
GIN	87.50 (↓ 9.55%)	93.94 (↓ 2.89%)	96.74	94.88 (↓ 1.92%)	89.62 (↓ 7.36%)
GIN + CAL	93.19 (↓ 3.87%)	96.31 (↓ 0.65%)	96.94	96.56 (↓ 0.39%)	95.25 (↓ 1.74%)
GIN + CAL+	94.31 (↓ 3.17%)	97.12 (↓ 0.28%)	97.40	97.19 (↓ 0.22%)	96.19(↓ 1.24%)
GAT	84.62 (↓ 8.71%)	89.50 (↓ 3.44%)	92.69	92.31 (↓ 0.41%)	87.62 (↓ 5.47%)
GAT + CAL	92.44 (↓ 4.37%)	96.25 (↓ 0.42%)	96.66	96.12 (↓ 0.56%)	92.56 (↓ 4.24%)
GAT + CAL+	92.56 (↓ 4.33%)	96.42 (↓ 0.34%)	96.75	96.69 (↓ 0.06%)	94.44(↓ 2.39%)

The number in brackets represents the performance degradation compared with the unbiased dataset. Our methods are highlighted with a gray background.

4.1.3 Hyperparameters. To help readers reproduce our results, we present the detailed settings of model, training, and hyperparameters for our method and baseline methods.

Our Settings. For dataset SYN-*b*, we train the models for 100 epochs using a batch size of 128, employing GNN encoders with three layers and 128 hidden units. For Graph OOD datasets [22] and OGB datasets [29], we use the GIN architecture as the encoder, select embedding dimensions from $\{32, 64, 128, 300\}$, choose batch sizes from $\{32, 64, 128, 256\}$, and set N in the range of $[2, 10]$ as the batch number for memory bank capacity. Moreover, we search for ρ_1, ρ_2, and λ within $(0, 1.0)$ with a step size of 0.1. We apply the default learning rate choices (e.g., 0.001, 0.002) for all experiments and optimize all models using the Adam optimizer across all datasets. All experiments are conducted using an NVIDIA 3090 Ti (24 GB GPU).

Baseline Settings. For all GNN models, such as attention-based GNNs, pooling-based GNNs, GCN, and GIN, we maintain the same settings as the CAL [63] and use the codes provided by these original papers to conduct experiments. Specifically, we use Adam optimizer and train the GNN models for 100 epochs with batch size of 128. For SYN-*b* dataset, we adopt the GNN encoders with 3 layers and 128 hidden units. For ERM, IRM [2], GroupDRO [60], VREx [36], and M-Mixup [73], we report the results from the GOOD [22] by default and reproduce the missing results on Molbbbp dataset. For DIR [75], CAL [63], DropEdge [58], GREA [47], FLAG [35], G-Mixup [23], CIGA [8], and DisC [13], they provide source codes for the implementations. We adopt their source codes to conduct experiments. For OOD-GNN [40] and StableGNN [14], their source codes are not publicly available. We reproduce them based on the codes of StableNet [91]. For all baseline methods, we use the same hyperparameter search range as ours.

4.2 Performance on Synthetic Graphs (RQ1)

To explore whether our proposed framework can alleviate the OOD issue, we first conduct experiments on SYN-*b* with different biases: $b \in \{0.1, 0.2, \ldots, 0.9\}$. The experimental results are summarized in Table 1 and Figure 5. We have the following **Obs**ervations:

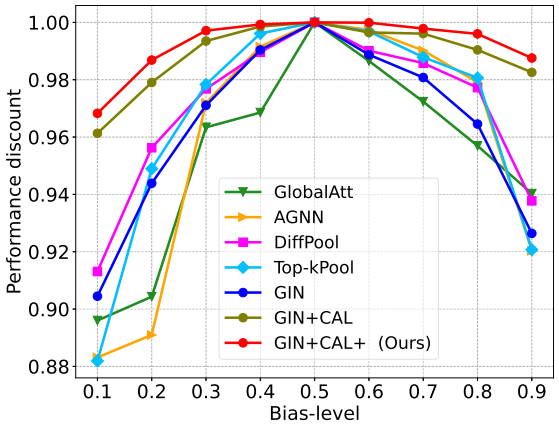


Fig. 5. The performance discount on synthetic datasets with different bias-levels.

Obs 1: Refining discriminative features without considering the causality leads to poor OOD generalization. For the unbiased dataset, most attention- and pooling-based baselines, such as GlobalAtt, SuperGAT, SortPool, Top-k Pool, outperform GCN. It indicates the effectiveness of extracting discriminative features in the ID setting. However, as the bias-level goes to extremes, the performance dramatically deteriorates. For instance, the performance drop of attention-based methods ranges from 7.37% $\sim$ 12.75% on SYN-0.1, and 3.79% $\sim$ 13.79% on SYN-0.9; Pooling-based methods drop from 7.82% $\sim$ 14.24% and 3.99% $\sim$ 12.10% for SYN-0.1 and SYN-0.9. These indicate that simply extracting discriminative features by attention or pooling module is prone to capture the data biases. These are also beneficial for reducing the training loss but lead to poor OOD generalization. Taking SYN-0.9 as an example, most "House" co-occur with "Tree" in the training data, so the model will mistakenly learn shortcut features from the "Tree"-type trivial subgraphs to make predictions, instead of probing the "House"-type causal subgraphs. This will mislead the model to adopt the "Tree" pattern to make decisions in the inference stage.

Obs 2: GNNs with better ID performance tend to have worse OOD generalization. For the unbiased dataset, GIN achieves the best performance (96.74%), while GAT (92.69%) outperforms the GCN (90.94%). This indicates that the **in-distribution (ID)** performance of these models exhibits such an order: GIN > GAT > GCN. However, when the bias is changed to 0.1 and 0.9, the performance of GIN drops by 9.55% and 7.36%, GAT drops by 8.71% and 5.47%, and GCN drops by 6.60% and 5.43%, respectively. It shows that the rankings of models' robustness against OOD issues are in the opposite order: GCN > GAT > GIN. This indicates that GNNs with better ID performance are prone to learn more shortcut features. Similar trends also occur in other baselines. After adopting CAL or CAL+, this phenomenon is significantly alleviated, which verifies the effectiveness of CAL and CAL+ in overcoming the OOD issue.

Obs 3: Mitigating the confounding effect can achieve more stable performance on OOD issues. We first define the performance discount on SYN-b as the accuracy on SYN-b normalized by the accuracy on unbiased SYN-0.5. It indicates the degree of the performance degradation on biased synthetic datasets without considering the model's ID generalization. We plot the performance discount curves on SYN-b with $b \in \{0.1, 0.2, \ldots, 0.9\}$. As depicted in Figure 5, we observe that pooling-based methods outperform GIN in a small range of bias-levels (0.2 $\sim$ 0.8), while the performance drops sharply when $b = 0.1$ or 0.9. For example, the performance discount of Top-k Pool drops from 0.95 to 0.88 as b reduces from 0.2 to 0.1. Attention-based methods perform worse than GIN when $b < 0.5$. For $b > 0.5$, AGNN achieves better performance than GIN, while

Table 2. Classification Performance of Different Methods on Synthetic and Real-world Datasets

Method	Motif		Molhiv		Molbbbp		CMNIST
	size	base	size	scaffold	size	scaffold	color
ERM	70.75±0.56	81.44±0.45	63.26±2.47	72.33±1.04	78.29±3.76	68.10±1.68	42.87±0.72
IRM	69.77±0.88	80.71±0.46	59.90±3.15	72.59±0.45	77.56±2.48	67.22±1.15	42.80±0.38
GroupDRO	69.98±0.86	81.43±0.70	61.37±2.79	**73.64±0.86**	79.27±2.43	66.47±2.39	43.32±0.75
VREx	70.24±0.72	81.56±0.35	60.23±1.70	72.60±0.82	78.76±2.37	68.74±1.03	43.31±0.78
DropEdge	55.27±5.93	70.84±6.81	54.92±1.73	66.78±2.68	78.32±3.44	66.49±1.55	38.43±1.94
FLAG	56.26±3.98	72.29±1.31	66.44±2.32	70.45±1.55	79.26±2.26	67.69±2.36	43.41±1.94
M-Mixup	67.81±1.13	77.63±0.57	64.87±1.77	72.03±0.53	78.92±2.43	68.75±1.03	40.96±0.81
G-Mixup	59.92±2.10	74.66±1.89	70.53±2.02	71.69±1.74	78.55±4.16	67.44±1.62	38.23±0.76
GREA	73.31±1.85	80.60±2.49	66.48±4.13	70.96±3.16	77.34±3.52	69.72±1.66	40.32±0.71
OOD-GNN	68.62±2.98	74.62±2.66	57.49±1.08	70.45±2.02	79.48±4.19	66.72±1.23	39.03±0.72
StableGNN	59.83±3.40	73.04±2.78	58.33±4.69	68.23±2.44	77.47±4.69	66.74±1.30	40.32±0.71
DIR	54.96±9.32	82.96±4.47	74.39±1.45	71.40±1.48	76.40±4.43	66.86±2.25	28.71±4.66
CIGA	70.65±4.81	77.48±2.54	73.62±1.33	71.65±1.33	76.08±1.21	66.43±1.99	39.39±3.30
DisC	53.34±13.71	76.70±0.47	56.59±10.09	67.12±2.11	75.68±3.16	60.72±0.89	34.18±1.88
CAL	66.64±2.74	68.54±2.14	62.36±1.42	72.61±1.84	79.50±4.81	68.06±2.60	42.48±0.48
CAL+ (ours)	**86.24±1.69**	**85.35±2.10**	**83.33±2.84**	73.05±1.86	**81.57 ±1.97**	**70.17±1.65**	**46.55±0.40**
Improvement	↑ 12.93%	↑ 1.64%	↑ 10.72%	↓ 0.59%	↑ 2.07%	↑ 0.45%	↑ 3.14%

The **bold** numbers indicate the best performance, while the underlined numbers indicate the second-best performance.

GlobalAttention often performs worse. These results reflect that attention- or pooling-based methods all have their own weaknesses, such that they cannot consistently overcome the diverse distribution shifts. Equipped with CAL and CAL+, GIN consistently outperforms all the baselines on all ranges of bias-levels and obviously keeps a large gap, which further demonstrates the significance of mitigating the confounding effect. Furthermore, the performance of CAL+ is better than that of CAL, indicating that incorporating the memory bank and prototype strategy can further enhance the effectiveness of backdoor adjustment.

4.3 Comparison with Existing Studies (RQ2)

In this section, we compare our method with more baseline methods, including general generalization algorithms, graph generalization algorithms, and graph augmentation methods. We conduct experiments on four graph classification datasets [22], including Motif, Molhiv, Molbbbp, and CMNIST, and two node classification datasets, including Citeseer and Amazon-Photo. For metrics, we use classification accuracy on the Motif, CMNIST, Citeseer, Amazon-Photo, and ROC-AUC on Molhiv and Molbbbp. We conduct 10 random runs and report the mean and standard deviation. The experimental results are shown in Tables 2 and 3. We make the following **Obs**ervations:

Obs 4: Existing efforts still exist limitations to address OOD issues. First, the direct application of general generalization algorithms to the graph domain does not yield significant improvements in performance. The average performance of these methods ranges from 67.22 to 67.94, which is merely on par with the ERM (i.e., 68.14). Second, current graph generalization algorithms do not consistently surpass ERM in performance. In particular, OOD-GNN and StableGNN fail to outperform ERM in the majority of cases. In contrast, invariant learning approaches such as DIR, CAL, and DisC exhibit superior performance to ERM in certain instances. For instance, DIR demonstrates a 9.35% improvement over ERM on Molhiv (size), while GSAT exhibits a 2.27% relative

Table 3. Performance of CAL+ in Node Classification Tasks

Method	Citeseer			Amazon-Photo		
	$r = 1/3$	$r = 0.5$	$r = 0.7$	$r = 1/3$	$r = 0.5$	$r = 0.7$
ERM	47.09±3.44	45.36±5.54	40.09±2.12	48.26±2.26	47.91±3.24	39.23±5.27
IRM	48.84±2.75	45.39±2.07	42.89±2.38	53.75±1.31	50.98±3.09	42.23±2.75
GroupDRO	49.32±6.47	46.30±5.44	40.68±2.83	49.62±6.45	47.65±8.34	41.15±5.50
VREx	47.53±3.65	43.11±4.06	43.03±4.29	47.13±8.01	48.53±8.37	37.49±5.39
CAL	56.37±2.48	47.59±2.14	46.89±2.32	53.68±1.70	51.25±2.83	42.38±3.44
CAL+ (ours)	**59.43**±1.46	**55.31**±1.86	**49.35**±2.46	**54.12**±1.66	**52.97**±2.25	**45.42**±2.11

improvement on Motif (base). Nevertheless, these methods also underperform in some scenarios. Specifically, DIR experiences a 14.47% decrease in performance compared to ERM in Motif (size), and GSAT records a 2.66% drop in Molbbbp (size). Third, data augmentation techniques display improved performance in specific settings. For example, GREA demonstrates 2.38% and 3.62% relative improvements over ERM on Molbbbp (scaffold) and Motif (size), respectively. G-mixup also achieves a 1.25% enhancement on the CMNIST dataset. However, in terms of average performance, these methods only maintain a comparable performance level (i.e., 61.58∼68.39) to ERM (i.e., 68.14). These findings suggest that existing generalization approaches continue to exhibit limitations in addressing **out-of-distribution (OOD)** issues.

Obs 5: Compared with the existing baselines, our method can effectively improve the generalization. Our proposed method, CAL+, exhibits substantial performance improvements and surpasses the majority of benchmark algorithms. Specifically, for Motif dataset, CAL+ registers accuracy improvements of 15.49% and 3.91% compared to ERM across two domains. In comparison to the optimal baseline algorithms, GREA and GSAT, CAL+ achieves improvements of 12.93% and 1.64%, respectively. For the CMNIST dataset, CAL+ records improvements of 3.68% and 3.14% in relation to ERM and the leading baseline, FLAG, respectively. For Molhiv and Molbbbp datasets, CAL+ consistently exhibits superior performance in the majority of instances. It surpasses ERM by an average of 11.68% and 1.40% across two distinct domains (i.e., size and scaffold). In terms of average performance, CAL+ registers improvements ranging from 6.78% to 14.56% in comparison to invariant learning-based approaches, such as DIR, CAL, GREA, and DisC. These findings underscore the effectiveness of the proposed CAL+ framework. Although our method is specifically designed to solve the graph classification problem, we follow the idea of ego-graph [74] to extend our method to the node classification task. The experimental results in Table 3 show that our method can defeat existing generalization methods and achieve significant performance improvements.

4.4 Further Analysis (RQ3)

In this section, we first explore the impact of different components on the final performance. We then explore the sensitivity of model performance to hyperparameters. Finally, we count the running time of the proposed method.

4.4.1 Ablation Study. We examine the impact of different components in CAL+ on the final performance, including node/edge attention mechanism, random combination, memory bank, and prototype strategies.

Node Attention & **Edge Attention. Node Attention (NA)** and **Edge Attention (EA)** refine the features from two different views: node-level and edge-level. Here, we want to examine the effect of adopting NA or EA alone. We adopt GCN as the encoder to conduct experiments on four biased synthetic datasets and two real-world datasets. GCN+CAL+ w/o NA or EA represents the

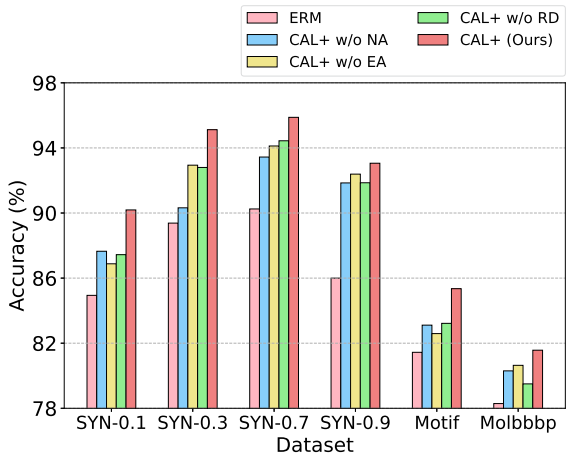


Fig. 6. The comparison of different components in CAL+.

node/edge attention scores in Equation (7)/(8) are evenly set as 0.5. The experimental results are shown in Figure 6. We can find that: (1) Comparing NA with EA, the performance of CAL+ without NA is significantly worse than that without EA, which indicates that the node feature contains more significant information compared with graph structure. (2) Just adopting NA or EA alone still achieves better performance than baselines, which demonstrates that only applying NA or EA can also disentangle the causal/trivial attended-graph and achieve causal intervention to some extent.

Random Combination. We need to stratify the confounder distribution for causal intervention. With the random combination, each causal feature will combine with different types of trivial patterns. To verify its importance, we change the "Random Combination" module in Figure 2 to "Combination," which just adopts the addition operation orderly in original graph, and we rename it as "GCN+CAL+ w/o RD." The experimental results are shown in Figure 6. We can find that: (1) The performance drops severely compared with GCN+CAL+, which demonstrates the importance of the causal intervention. (2) GCN+CAL+ w/o RD can also outperform the GCN baselines. We conjecture that just implementing disentanglement makes GNN pay more attention to the causal features, which will slightly ignore the data biases or trivial patterns. These results also reflect that disentanglement and causal intervention will help each other to improve their own effectiveness.

Memory Bank & **Prototype.** On the one hand, to mitigate the confounding effect on generalization, we leverage backdoor adjustments by combining each causal feature with the stratification of confounding factors. However, the mini-batch combination adopted by CAL limits the variety of shortcut features for each combination, reducing the effectiveness of backdoor adjustment. Therefore, we propose the memory bank to address this issue. On the other hand, CAL does not consider the consistency of intra-class causal estimation, which may lead to inaccurate and unstable causal feature estimations. Consequently, we adopt class-wise prototypes to enhance the stability of causal feature estimation. We independently verify the impact of these two modules on the final performance, and the experimental results are shown in Table 4. "w/ Mem" and "w/ Pro" represent the application of memory bank and prototype modules to the original CAL model, respectively. From the results, we make the following observations: (1) The performance of CAL with memory bank consistently outperforms CAL across all datasets. Specifically, it achieves 18.38% and 20.68% improvements over CAL on Motif (size) and Molhiv (size). Furthermore, its average performance exceeds CAL by 8.51% across four datasets. (2) For "CAL w/ Pro," a similar performance improvement trend is observed. For instance, on the Molhiv (size) and CMNIST datasets, it

Table 4. The Impact of Different Components in CAL+

Method	Motif		Molhiv		Molbbbp		CMNIST
	size	scaffold	size	base	size	scaffold	color
CAL	66.64±2.74	68.54±2.14	62.36±1.42	72.61±1.84	79.50±4.81	68.06±2.60	42.48±0.48
CAL w/ Mem	85.02±1.18	81.65±13.11	83.04±2.17	72.72±2.35	81.19±2.52	69.73±1.56	46.42±0.71
CAL w/ Pro	83.54±2.16	84.08±1.72	**85.17±2.11**	72.62±1.95	81.13±1.67	70.11±2.02	46.16±0.31
CAL+ (ours)	**86.24±1.69**	**85.35±2.10**	83.33±2.84	**73.05±1.86**	**81.57±2.07**	**70.17±1.65**	**46.55±0.40**

The **bold** numbers indicate the best performance.

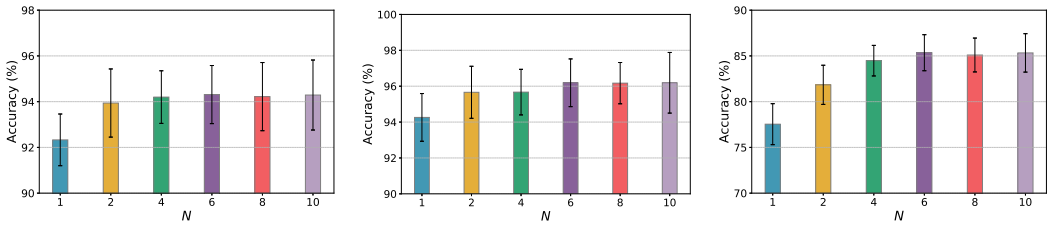
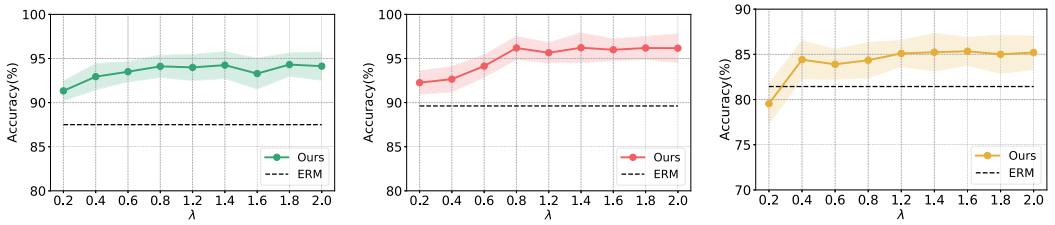


Fig. 7. Performance of CAL+ method with memory banks of different capacities.

enhances performance by 22.81% and 3.68% compared to CAL, respectively. This demonstrates that prototypes can enable CAL to capture more stable causal features and achieve generalization improvement. (3) Finally, CAL+ can further improve performance. In terms of average performance, CAL+ improves by 9.41% compared to CAL and by 0.90% and 0.47% compared to CAL w/ Mem and CAL w/ Pro, respectively. These results indicate that a more stable estimation of causal features and higher-quality backdoor adjustment strategies can complement each other, further eliminate confounding effects, and thereby enhance the OOD generalization.

4.4.2 Hyperparameter Sensitivity Analysis. In our CAL+ model, two critical hyperparameters are pivotal: the memory bank capacity N and the causal intervention coefficient λ. The capacity N in the memory bank denotes the count of shortcut feature representations employed by the model for backdoor adjustment. An increase in N allows for a broader array of representations, thereby enriching the diversity of shortcut features used for causal intervention. The coefficient λ, however, quantifies the intensity of causal intervention by the model. To assess the influence of these hyperparameters on CAL+'s performance, we conduct a comprehensive series of experiments. The values for N are varied across the set $\{1, 2, 4, 6, 8, 10\}$, while λ is tested over the range $[0.2, 2.0]$, incrementing in steps of 0.2. The outcomes of these experiments are presented in Figures 7 and 8. From the results, it is evident that increasing N enhances performance up to a certain point; specifically, beyond $N = 6$ the performance plateaued. Regarding the coefficient λ, an incremental improvement in performance is observed between 0.2 and 0.6, stabilizing thereafter up to 2.0. These findings underscore the significance of both the memory bank capacity and causal intervention in enhancing model performance. Moreover, they indicate that within an optimal range, CAL+'s performance exhibits a degree of insensitivity to variations in these hyperparameters.

4.4.3 Running Time and Model Size. Contrasting with the traditional GNN encoder, our enhanced CAL+ model integrates node and edge attention mechanisms, along with dual GNN layer modules. This integration introduces additional parameters, contributing to increased time and space complexity due to the inclusion of a memory bank and prototype. However, these increases



(a) SYN-0.1 (b) SYN-0.9 (c) Motif

Fig. 8. Hyperparameter sensitivity analysis on the coefficient of λ.

Table 5. Running Time, Model Size, and Performance Improvement

Dataset	ERM		CAL			CAL+		
	Running Time	Model Size	Running Time	Model Size	Performance Improvement	Model Size	Model Size	Performance Improvement
Motif	00h 51m 19s	1.515M	01h 37m 15s	2.213M	↓ 11.18%	01h 44m 46s	2.261M	↑ 12.73%
Molhiv	00h 27m 19s	1.515M	00h 46m 14s	2.213M	↓ 0.46%	00h 50m 37s	2.261M	↑ 15.34%
Molbbbp	00h 11m 58s	1.515M	00h 18m 22s	2.213M	↑ 0.81%	00h 24m 20s	2.261M	↑ 3.66%
CMNIST	01h 56m 32s	1.517M	02h 28m 49s	2.244M	↓ 0.91%	02h 41m 12s	2.397M	↑ 8.58%

remain within acceptable limits. Comprehensive comparisons regarding running time, model size, and performance improvements between CAL+, CAL, and the original model are conducted across various datasets, with the results detailed in Table 5. These experiments reveal that the running time of CAL+ is approximately 1.5 to 2 times longer than that of the base model, and the model size is about 1.5 times larger. Notably, CAL+ demonstrates comparable running time and model size to CAL. This comparison underscores a more favorable performance-complexity balance in CAL+, especially considering its significant performance enhancements. Therefore, we contend that the tradeoffs in complexity are justified and manageable for practical applications.

4.5 Visualization and Analysis (RQ4)

In this section, we plot node and edge attention areas of the causal attended-graphs based on the attention scores in CAL+. We adopt a GCN-based encoder and apply CAL and CAL+ on SYN-b. The visualizations are shown in Figure 9. Nodes with darker colors and edges with wider lines indicate higher attention scores. The results obtained by CAL and CAL+ demonstrate that most of the darker nodes and wider edges are distributed within the causal subgraphs in the graph data. This indicates that both methods can effectively make predictions based on causal features in the data to some degree. Furthermore, compared to CAL, CAL+ focuses less on trivial subgraphs. This outcome reveals that our proposed memory bank and prototype strategies can further refine the model's focus on causal features while minimizing the influence of shortcut features.

5 RELATED WORK

In this section, we briefly review some related studies, including attention mechanism, out-of-distribution generalization, and causal inference.

5.1 Attention Mechanism

Attention Mechanism selects the informative features from data, which has obtained great success in computer vision [11, 28, 61, 69, 81] and natural language processing tasks [10, 67]. In recent years, attention mechanism has gradually become prevalent in the GNN field. The attention modules for

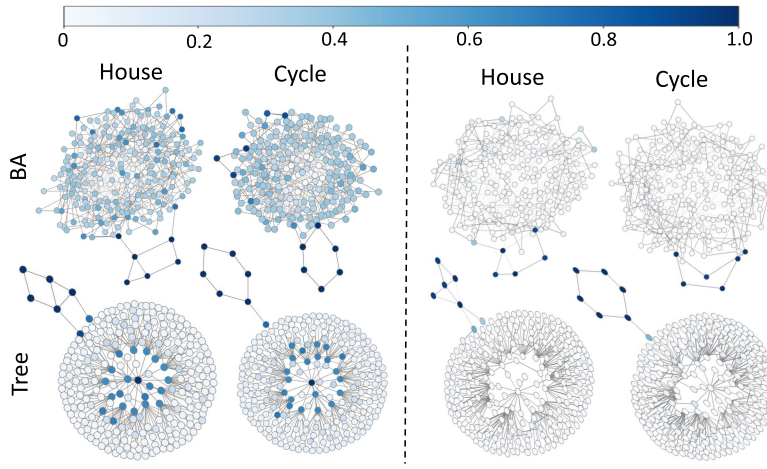


Fig. 9. Visualizations of causal attended-graphs. (Left): Attention scores of CAL, (Right): Attention scores of CAL+.

GNNs can be defined over edges [4, 32, 39, 66, 68] or over nodes [37, 38, 44]. However, most attention-based approaches still focus on how to better fit statistical correlations between data and labels. Hence, the learned attentions are inherently biased in OOD settings. Recent studies [69, 81] propose the causal attention modules to alleviate the bias. CaaM [69] adopts the adversarial training to generate the data partition in each iteration to achieve the causal intervention. CATT [81] proposes in-sample and cross-sample attentions based on front-door adjustment. However, they are both tailored for computer vision tasks, while they cannot transfer to graph learning tasks, due to the irregular and challenging graph-structure data. Distinct from them, we utilize the disentanglement and causal intervention strategies to strengthen the attention modules for GNNs.

5.2 Out-of-distribution Generalization

OOD generalization [2, 25, 59, 60] has been extensively explored in recent years. IRM [2] minimizes the empirical risk under different environments. Group-DRO [56, 57, 60] adversarially explores the group with the worst risk and achieves generalization by minimizing the empirical risk of the worst group. Recent studies [22, 41] have demonstrated that OOD problem is also prevalent in graph domain, including graph classification [3, 5, 72, 75, 82] and node classification tasks [74, 92]. Consequently, research focusing on graph generalization has been gaining traction, encompassing areas such as data augmentation [23, 31, 73, 85], stable learning [14, 40], and invariant learning [8, 13, 42, 49, 64, 80]. Among them, invariant learning has progressively evolved into a dominant paradigm for addressing OOD issue in graph-related tasks. It typically operates on the premise of graph data generation, which asserts that causal features exist within the data, that these features exhibit a causal relationship with the label, and that this relationship remains consistent across diverse environments. To capture these causal features, DIR intervenes in environmental features, promoting model predictions that maintain invariance. GREA [47], CAL [63], and DisC [13] also employ similar ideas, advocating for models that generate predictions based on causal features.

5.3 Causal Inference in Machine Learning

Causal Inference [53, 54] endows the model with the ability to pursue real causality. Thus, the model can avoid the interference from confounding factors. A growing number of studies [30, 51, 65, 88] have shown that causal inference is beneficial to diverse computer vision tasks.

CONTA [88] uses backdoor adjustment to eliminate the confounder in weakly supervised semantic segmentation tasks. DDE [30] proposes to distill the colliding effect between the old and the new data to improve class-incremental learning. Unlike computer vision, the application of causal intervention in the GNN community is still in its infancy. CGI [17] explores how to select trustworthy neighbors for GNN in the inference stage and shows its effectiveness in node classification. Reference [87] studies the connection between GNNs and SCM from a theoretical perspective. Different from them, we introduce a causal attention learning strategy to mitigate the confounding effect for GNNs. It encourages GNNs to pay more attention to causal features, which will enhance the robustness against the distribution shift.

6 CONCLUSION

In this work, we revisit the GNN modeling for graph classification from a causal view. We find that current GNN learning strategies are prone to exploit the shortcut features to support their predictions. However, the shortcut feature actually plays a confounder role. It establishes a backdoor path between the causal feature and the prediction, which misleads the GNNs to learn spurious correlations. To mitigate the confounding effect, we propose the CAL+, which is guided by the backdoor adjustment from the causal theory. It encourages the GNNs to exploit causal features while ignoring the shortcut parts. Extensive experimental results and analyses verify its effectiveness. In future work, We will also make efforts to apply CAL+ to other graph learning tasks, such as node classification or link prediction.

REFERENCES

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 11 (2012), 2274–2282.

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

[3] Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. 2021. Size-invariant graph representations for graph classification extrapolations. In *ICML*. PMLR, 837–851.

[4] Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks? In *ICLR*.

[5] Davide Buffelli, Pietro Lio, and Fabio Vandin. 2022. SizeShiftReg: A regularization method for improving size-generalization in graph neural networks. In *NeurIPS*.

[6] Yongqiang Chen, Yatao Bian, Kaiwen Zhou, Binghui Xie, Bo Han, and James Cheng. 2023. Does invariant graph learning via environment augmentation learn invariance? In *NeurIPS*. Retrieved from https://openreview.net/forum?id=EqpR9Vtt13

[7] Yongqiang Chen, Yatao Bian, Kaiwen Zhou, Binghui Xie, Bo Han, and James Cheng. 2023. Rethinking invariant graph representation learning without environment partitions In *ICML DG Workshop*.

[8] Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, M. A. Kaili, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. 2022. Learning causally invariant representations for out-of-distribution generalization on graphs. *Neural Inf. Process.* 35 (2022), 22131–22148.

[9] Asim Kumar Debnath, Rosa L. Lopez de Compadre, Gargi Debnath, Alan J. Shusterman, and Corwin Hansch. 1991. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *J. Medicin. Chem.* 34, 2 (1991).

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*. 4171–4186.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

[12] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2020. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982* (2020).

[13] Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang. 2022. Debiasing graph neural networks via learning disentangled causal substructure. In *NeurIPS*.

[14] Shaohua Fan, Xiao Wang, Chuan Shi, Peng Cui, and Bai Wang. 2021. Generalizing graph neural networks on out-of-distribution graphs. *arXiv preprint arXiv:2111.10657* (2021).

[15] Junfeng Fang, Wei Liu, Yuan Gao, Zemin Liu, An Zhang, Xiang Wang, and Xiangnan He. 2023. Evaluating post-hoc explanations for graph neural networks via robustness analysis. In *NeurIPS*.
[16] Junfeng Fang, Xiang Wang, An Zhang, Zemin Liu, Xiangnan He, and Tat-Seng Chua. 2023. Cooperative explanations of graph neural networks. In *WSDM*. ACM, 616–624.
[17] Fuli Feng, Weiran Huang, Xiangnan He, Xin Xin, Qifan Wang, and Tat-Seng Chua. 2021. Should graph convolution trust neighbors? A simple causal inference method. In *SIGIR*. 1208–1218.
[18] Hongyang Gao and Shuiwang Ji. 2019. Graph U-Nets. In *ICML*. 2083–2092.
[19] Yuan Gao, Xiang Wang, Xiangnan He, Huamin Feng, and Yong-Dong Zhang. 2023. Rumor detection with self-supervised learning on texts and social graph. *Front. Comput. Sci.* 17, 4 (2023), 174611.
[20] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. 2023. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *WWW*. ACM, 1528–1538.
[21] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. 2023. Alleviating structural distribution shift in graph anomaly detection. In *WSDM*. ACM, 357–365.
[22] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. 2022. Good: A graph out-of-distribution benchmark. In *NeurIPS*.
[23] Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. 2022. G-mixup: Graph data augmentation for graph classification. In *ICML*. PMLR, 8230–8248.
[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.
[25] Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*.
[26] Miguel A. Hernan and James M. Robins. 2010. *Causal Inference: What If*. CRC Press.
[27] Paul W. Holland. 1986. Statistics and causal inference. *J. Am. Stat. Assoc.* 81, 396 (1986), 945–960.
[28] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *CVPR*.
[29] 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. *Neural Inf. Process.* 33 (2020), 22118–22133.
[30] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. 2021. Distilling causal effect of data in class-incremental learning. In *CVPR*.
[31] Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. 2023. Empowering graph representation learning with test-time graph transformation. In *ICLR*.
[32] Dongkwan Kim and Alice Oh. 2020. How to find your friendly neighborhood: Graph attention design with self-supervision. In *ICLR*.
[33] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
[34] Boris Knyazev, Graham W. Taylor, and Mohamed R. Amer. 2019. Understanding attention and generalization in graph neural networks. In *NeurIPS*. 4204–4214.
[35] Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. 2022. Robust optimization as data augmentation for large-scale graphs. In *CVPR*. 60–69.
[36] 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 *ICML*. PMLR, 5815–5826.
[37] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. In *ICML*. 3734–3743.
[38] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. 2018. Graph classification using structural attention. In *KDD*. 1666–1674.
[39] John Boaz Lee, Ryan A. Rossi, Xiangnan Kong, Sungchul Kim, Eunyee Koh, and Anup Rao. 2019. Graph convolutional networks with motif-based attention. In *CIKM*. 499–508.
[40] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2023. OOD-GNN: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering* 35, 7 (2023), 7328–7340. DOI: 10.1109/TKDE.2022.3193725
[41] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2022. Out-of-distribution generalization on graphs: A survey. *arXiv preprint arXiv:2202.07987* (2022).
[42] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. 2022. Learning invariant graph representations for out-of-distribution generalization. In *NeurIPS*.
[43] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. 2023. Invariant node representation learning under distribution shifts with multiple latent environments. *ACM Trans. Inf. Syst.* 42, 1 (2023), 1–30.
[44] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated graph sequence neural networks. In *ICLR*.
[45] Wanyu Lin, Hao Lan, and Baochun Li. 2021. Generative causal explanations for graph neural networks. In *ICML*. PMLR, 6666–6679.

[46] Yong Lin, Shengyu Zhu, Lu Tan, and Peng Cui. 2022. ZIN: When and how to learn invariance without environment partition? *Neural Inf. Process* 35 (2022), 24529–24542.
[47] Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. 2022. Graph rationalization with environment-based augmentations. In *KDD*. 1069–1078.
[48] Divyat Mahajan, Shruti Tople, and Amit Sharma. 2021. Domain generalization using causal matching. In *ICML*. PMLR, 7313–7324.
[49] Siqi Miao, Mia Liu, and Pan Li. 2022. Interpretable and generalizable graph learning via stochastic attention mechanism. In *ICML*. PMLR, 15524–15543.
[50] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. 2020. Tu-dataset: A collection of benchmark datasets for learning with graphs. In *ICMLW*.
[51] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual VQA: A cause-effect look at language bias. In *CVPR*. 12700–12710.
[52] Judea Pearl. 2010. Causal inference. *Proceedings of Workshop on Causality: Objectives and Assessment at NIPS 2008 (PMLR), Proceedings of Machine Learning Research*, Vol. 6, 39–58.
[53] Judea Pearl. 2014. Interpretation and identification of causal mediation. *Psychol. Meth.* 19, 4 (2014), 459.
[54] Judea Pearl. 2000. Models, reasoning and inference. *Cambridge, UK: Cambridge University Press* 19 (2000).
[55] Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect*. Basic Books.
[56] Qi Qi, Jiameng Lyu, Kung sik Chan, Er Wei Bai, and Tianbao Yang. 2022. Stochastic constrained DRO with a complexity independent of sample size. *arXiv preprint arXiv:2210.05740* (2022).
[57] Qi Qi, Yi Xu, Rong Jin, Wotao Yin, and Tianbao Yang. 2020. Attentional biased stochastic gradient for imbalanced classification. *arXiv preprint arXiv:2012.06951* (2020).
[58] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. DropEdge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903* (2019).
[59] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. 2020. The risks of invariant risk minimization. In *ICLR*.
[60] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *ICLR*.
[61] Yongduo Sui, Tianlong Chen, Pengfei Xia, Shuyao Wang, and Bin Li. 2022. Towards robust detection and segmentation using vertical and horizontal adversarial training. In *IJCNN*. IEEE, 1–8.
[62] Yongduo Sui, Xiang Wang, Tianlong Chen, Meng Wang, Xiangnan He, and Tat-Seng Chua. 2023. Inductive lottery ticket learning for graph neural networks. *J. Comput. Sci. Technol.* (2023).
[63] Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. 2022. Causal attention for interpretable and generalizable graph classification. In *KDD*. 1696–1705.
[64] Yongduo Sui, Qitian Wu, Jiancan Wu, Qing Cui, Longfei Li, Jun Zhou, Xiang Wang, and Xiangnan He. 2023. Unleashing the power of graph data augmentation on covariate distribution shift. In *NeurIPS*.
[65] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. 2020. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*.
[66] Kiran K. Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li. 2018. Attention-based graph neural network for semi-supervised learning. *arXiv preprint arXiv:1803.03735* (2018).
[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.
[68] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
[69] Tan Wang, Chang Zhou, Qianru Sun, and Hanwang Zhang. 2021. Causal attention for unbiased visual recognition. In *CVPR*. 3091–3100.
[70] Xiang Wang, Yingxin Wu, An Zhang, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2023. Reinforced causal explainer for graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 2 (2023), 2297–2309. DOI: 10.1109/TPAMI.2022.3170302
[71] Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat seng Chua. 2021. Towards multi-grained explainability for graph neural networks. In *NeurIPS*.
[72] Yiqi Wang, Yao Ma, Wei Jin, Chaozhuo Li, Charu Aggarwal, and Jiliang Tang. 2020. Customized graph neural networks. *arXiv preprint arXiv:2005.12386* (2020).
[73] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. 2021. Mixup for node and graph classification. In *WWW*. 3663–3674.
[74] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. 2022. Handling distribution shifts on graphs: An invariance perspective. In *ICLR*.

[75] Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. 2022. Discovering invariant rationales for graph neural networks. In *ICLR*.
[76] 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. *Chem. Sci.* 9, 2 (2018), 513–530.
[77] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*. 3733–3742.
[78] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.
[79] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? In *ICLR*.
[80] Nianzu Yang, Kaipeng Zeng, Qitian Wu, Xiaosong Jia, and Junchi Yan. 2022. Learning substructure invariance for out-of-distribution molecular representations. In *NeurIPS*.
[81] Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. 2021. Causal attention for vision-language tasks. In *CVPR*. 9847–9857.
[82] Gilad Yehudai, Ethan Fetaya, Eli Meirom, Gal Chechik, and Haggai Maron. 2021. From local structures to size generalization in graph neural networks. In *ICML*. PMLR, 11975–11986.
[83] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating explanations for graph neural networks. In *NeurIPS*. 9240–9251.
[84] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *NeurIPS*. 4805–4815.
[85] Junchi Yu, Jian Liang, and Ran He. 2023. Mind the label shift of augmentation-based graph OOD generalization. In *CVPR*.
[86] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards model-level explanations of graph neural networks. In *KDD*. 430–438.
[87] Matej Zečević, Devendra Singh Dhami, Petar Veličković, and Kristian Kersting. 2021. Relating graph neural networks to structural causal models. *arXiv preprint arXiv:2109.04173* (2021).
[88] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. 2020. Causal intervention for weakly-supervised semantic segmentation. In *NeurIPS*.
[89] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An end-to-end deep learning architecture for graph classification. In *AAAI*.
[90] Michael Zhang, Nimit S. Sohoni, Hongyang R. Zhang, Chelsea Finn, and Christopher Re. 2022. Correct-N-contrast: A contrastive approach for improving robustness to spurious correlations. In *ICML*. PMLR, 26484–26516.
[91] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. 2021. Deep stable learning for out-of-distribution generalization. In *CVPR*. 5372–5382.
[92] Qi Zhu, Natalia Ponomareva, Jiawei Han, and Bryan Perozzi. 2021. Shift-robust GNNS: Overcoming the limitations of localized graph training data. *Neural Inf. Process.* 34 (2021), 27965–27977.

Received 7 June 2023; revised 19 November 2023; accepted 16 January 2024