Causal Attention for Interpretable and Generalizable Graph Classification

Yongduo Sui
syd2019@mail.ustc.edu.cn
University of Science and Technology of China

Xiang Wang*
xiangwang1223@gmail.com
University of Science and Technology of China

Jiancan Wu
wujcan@gmail.com
University of Science and Technology of China

Min Lin
linmin@sea.com
Sea AI Lab

Xiangnan He
xiangnanhe@gmail.com
University of Science and Technology of China

Tat-Seng Chua
dcscts@nus.edu.sg
National University of Singapore

ABSTRACT

In graph classification, attention- and pooling-based graph neural networks (GNNs) prevail to extract the critical 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 makes GNN classifiers recklessly absorb all the statistical correlations between input features and labels in the training data, without distinguishing the causal and noncausal effects of features. Instead of underscoring the causal features, the attended graphs are prone to visit the noncausal features as the shortcut to predictions. Such shortcut features might easily change outside the training distribution, thereby making the GNN classifiers suffer from poor generalization.

In this work, we take a causal look at the GNN modeling for graph classification. With our causal assumption, the shortcut feature serves as a confounder between the causal feature and prediction. It tricks the classifier to learn spurious correlations that facilitate the prediction in in-distribution (ID) test evaluation, while causing the performance drop in out-of-distribution (OOD) test data. To endow the classifier with better interpretation and generalization, we propose the Causal Attention Learning (CAL) strategy, which discovers the causal patterns and mitigates the confounding effect of shortcuts. Specifically, we employ attention modules to estimate the causal and shortcut features of the input graph. We then parameterize the backdoor adjustment of causal theory — combine each causal feature with various shortcut features. It encourages the stable relationships between the causal estimation and the prediction, regardless of the changes in shortcut parts and distributions. Extensive experiments on synthetic and real-world datasets demonstrate the effectiveness of CAL.

*Xiang Wang is the corresponding author.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539366

CCS CONCEPTS

• **Mathematics of computing → Graph algorithms**; • **Computing methodologies → Learning latent representations**.

KEYWORDS

Graph Neural Networks, Graph Classification, Causal Intervention

ACM Reference Format:
Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. 2022. Causal Attention for Interpretable and Generalizable Graph Classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3534678.3539366

1 INTRODUCTION

Graph neural networks (GNNs) [8, 17] have exhibited impressive performance of graph classification across various domains, such as chemical molecules, social networks, and transaction graphs. Such a success mainly comes from the powerful representation learning of GNNs, which incorporates the graph structure and encodes them into the representations in an end-to-end way. Hence, it is crucial to emphasize the critical part of the input graph, while filtering the trivial part out [23, 37–39]. For example, when classifying the mutagenic property of a molecular graph [24], GNNs are expected to latch on the functional groups (*i.e.,* nitrogen dioxide (NO_2)), instead of the irrelevant patterns (*i.e.,* carbon rings) [5, 46]; when detecting fraud in a transaction network, malicious behaviors or coalitions of users are more informative than benign features.

Towards specifying the critical parts in graphs, some follow-on studies [10, 16, 35, 45] adopt the paradigm of "learning to attend" [34, 40] — maximizing the mutual information between the attended graph and the ground-truth label — to find the attended graph that maximizes the predictive performance. Specifically, there are two research lines in this paradigm: (1) Attention-based methods [4, 16, 22, 33, 35]. They often utilize the attention modules for nodes or edges to locate the attended graphs. These attention modules act like soft masks to identify the importance of each edge and node to the final representations and predictions. (2) Pooling-based methods [10, 19, 45, 49]. They directly adopt hard masks to select a subset of nodes or edges as the attended graphs, to perform the information propagations. These attended graphs aim to approach

the features that are beneficial for minimizing the training loss, instead of distinguishing the causal and noncausal effects.

Unfortunately, recent efforts [2, 11, 12, 18] have shown that the current attention or pooling learning methods are prone to exploit the shortcut features to make decisions. These shortcuts usually come from the data selection biases, noisy features, or some trivial patterns from graphs, which are noncausal but discriminative in training data. Due to the existence of these shortcuts, models can capture shortcut features to finish the classification tasks without struggling to learn causal features. For example, instead of probing into the causal effect of the functional groups, the attended graphs prefer "carbon rings" as the cues of the "mutagenic" class, because most training "mutagenic" molecules are in the "carbon rings" context. While such correlations represent statistical relations inherent in the training data and are beneficial to the in-distribution (ID) test evaluations, they inevitably cause a huge performance drop in the out-of-distribution (OOD) test data that are at odds with the training distribution. Taking the molecule classification as an example again, when most test "non-mutagenic" molecules appear in the "carbon rings" context, the attended graphs mislead the GNNs to still predict "mutagenic". As the assumption that the test data conforms to the training distribution is often infeasible in real-world scenarios, the poor generalization of these methods hinders their deployment on critical applications.

To resolve this issue, we first take a causal look at the decision-making process of GNNs for graph classification, which delineates the relationships among the causal feature, shortcut feature, and prediction. With our causal assumption in Figure 1, the shortcut feature serves as a confounder [27]. It opens a backdoor path [26] and makes the causal feature and prediction spuriously correlated, *e.g.,* misclassifying "non-mutagenic" molecules with "carbon rings" to the "mutagenic" molecules. Hence, mitigating the confounding effect is promising to exploit the causal features while filtering out the shortcut patterns, thereby enhancing the generalization.

Towards this end, we propose the Causal Attention Learning (CAL) strategy — maximizing the causal effect of the attended graph on predicting the label, while reducing the confounding effect of the shortcut features. Our attended graph aims to approach the causal features in the graph (*e.g.,* nitrogen dioxide), while its complement targets the shortcut features (*e.g.,* carbon rings). Specifically, we first apply attention modules to generate the estimations of the causal and shortcut features from the input graphs. We then parameterize the backdoor adjustment in the causal theory [26, 27], which combines each causal estimation with various shortcut estimations and encourages these combinations to maintain a stable prediction. It encourages the invariant relationships between the causal patterns and the predictions, regardless of the changes in the shortcut parts and distribution shifts. We apply CAL to various GNN architectures for graph classification. Experimental results on numerous synthetic and real-world datasets demonstrate the better generalization and insightful interpretations of CAL.

Our technical contributions are summarized as:

- We emphasize the generalization issue of current attention- and pooling-based GNNs in graph classification. From the causal perspective, we ascribe such an issue to the confounding effect of the shortcut features.
- We present a novel Causal Attention Learning (CAL) strategy for graph classification. It makes GNNs exploit the causal features while filtering out the shortcut patterns.
- Extensive experiments on synthetic and real-world datasets justify the effectiveness of CAL. More visualizations with in-depth analyses demonstrate the interpretability and rationality of CAL.

2 PRELIMINARIES

2.1 Notations

We denote a graph by $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$ 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 i-th 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, 16, 21, 33, 35], 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 [18, 19, 22] 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 [19] for the output node representations $\mathbf{H}^{out}$ and summarize the graph representation $\mathbf{h}_\mathcal{G}$ for graph $\mathcal{G}$ via the readout function $f_{\mathrm{readout}}(\cdot)$. Then we use a classifier Φ to project the graph representation into a probability distribution $\mathbf{z}_\mathcal{G}$:

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

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

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{|\mathcal{D}|}\sum_{\mathcal{G}\in\mathcal{D}} \mathbf{y}_\mathcal{G}^\top \log(\mathbf{z}_\mathcal{G}). \tag{4}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss over the training data $\mathcal{D}$, and $\mathbf{y}_\mathcal{G}$ is the ground-truth label vector of $\mathcal{G}$. However, this learning strategy heavily relies on the statistical correlations between the input graphs and labels. Hence, they will inevitably capture the noncausal shortcut features to make predictions.

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.

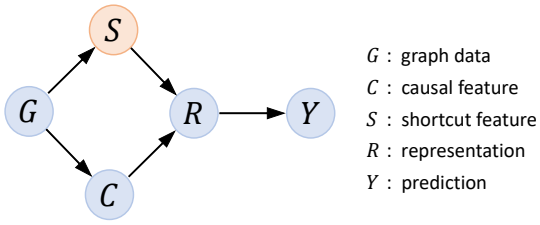

Figure 1: Structural causal model for graph classification.

3.1 A Causal View on GNNs

We take a causal look at the GNN modeling and construct a Structural Causal Model (SCM) [27] in Figure 1. It presents the causalities among five variables: graph data G, causal feature C, shortcut feature S, graph representation R, and prediction Y, where the link from one variable to another indicates the cause-effect relationship: cause $\rightarrow$ effect. We list the following explanations for SCM:

- $\boldsymbol{C \leftarrow G \rightarrow S}$. The variable C denotes the causal feature that truly reflects the intrinsic property of the graph data G. While S represents the shortcut feature which 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.
- $\boldsymbol{C \rightarrow R \leftarrow S}$. 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 the causal feature C as input to distill discriminative information.
- $\boldsymbol{R \rightarrow Y}$. 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 [26, 27] 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:

- 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, $P(s) = P_m(s)$.
- The conditional probability $P(Y|C, s)$ is invariant, because Y's 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 causal intervention, which we have: $P_m(s|C) = P_m(s)$.

Based on the above conclusions, we have:

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

where $\mathcal{T}$ denotes the confounder set; $P(Y|C, s)$ represents 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** [26], 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 $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$, 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 $\mathcal{G}$ into two attended-graphs: $\mathcal{G}_1 = \{\mathbf{A} \odot \mathbf{M}_a, \mathbf{X} \odot \mathbf{M}_x\}$ and $\mathcal{G}_2 = \{\mathbf{A} \odot \overline{\mathbf{M}}_a, \mathbf{X} \odot \overline{\mathbf{M}}_x\}$.

With the inspection on the data-generating process, recent studies [18, 23, 39, 44] 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 [37]; 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 $\mathcal{G}$, we define the attended graph collecting all causal features as the **causal attended-graph** $\mathcal{G}_c$, while the counterpart forms the **trivial attended-graph** $\mathcal{G}_t$. 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: $\mathcal{G}_c = \{\mathbf{A} \odot \mathbf{M}_a, \mathbf{X} \odot \mathbf{M}_x\}$ and $\mathcal{G}_t = \{\mathbf{A} \odot \overline{\mathbf{M}}_a, \mathbf{X} \odot \overline{\mathbf{M}}_x\}$. 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 aforementioned backdoor adjustment, we propose the **C**ausal **A**ttention **L**earning (CAL) framework:

3.4.1 ***Estimating soft masks.*** 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 $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$, we can obtain

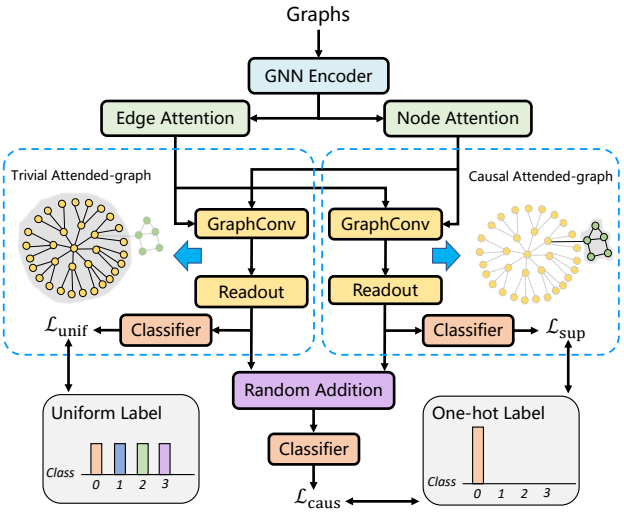

Figure 2: The overview of the proposed Causal Attention Learning (CAL) framework.

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 estimate the attention scores from two orthogonal perspectives: node-level and edge-level. For node v_i and edge (v_i, v_j) we can obtain:

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

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

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 original graph $\mathcal{G}$ into the initial causal and trivial attended-graphs: $\mathcal{G}_c = \{\mathbf{A} \odot \mathbf{M}_a, \mathbf{X} \odot \mathbf{M}_x\}$ and $\mathcal{G}_t = \{\mathbf{A} \odot \overline{\mathbf{M}}_a, \mathbf{X} \odot \overline{\mathbf{M}}_x\}$.

3.4.2 ***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:

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

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

The causal attended-graph aims to estimate the causal features, so we classify its representation to the ground-truth label. Thus, we define the supervised classification loss as:

$$\mathcal{L}_{\text{sup}} = -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{G} \in \mathcal{D}} \mathbf{y}_{\mathcal{G}}^{\top} \log(\mathbf{z}_{\mathcal{G}_c}). \tag{11}$$

where $\mathcal{L}_{\text{sup}}$ is the cross-entropy loss over the training data $\mathcal{D}$. The trivial attended-graph aims to approach the trivial patterns that are unnecessary for classification. Hence, we push its prediction evenly to all categories and define the uniform classification loss as:

$$\mathcal{L}_{\text{unif}} = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{G} \in \mathcal{D}} \text{KL}(\mathbf{y}_{\text{unif}}, \mathbf{z}_{\mathcal{G}_t}). \tag{12}$$

where KL denotes the KL-Divergence, $\mathbf{y}_{\text{unif}}$ represents the uniform distribution. By optimizing the above two objectives, we can effectively disentangle causal and trivial features. Please note that prior efforts [18, 23, 37, 38, 44] 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 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:

$$\mathbf{z}_{\mathcal{G}'} = \Phi(\mathbf{h}_{\mathcal{G}_c} + \mathbf{h}_{\mathcal{G}_{t'}}), \tag{13}$$

$$\mathcal{L}_{\text{caus}} = -\frac{1}{|\mathcal{D}| \cdot |\hat{\mathcal{T}}|} \sum_{\mathcal{G} \in \mathcal{D}} \sum_{t' \in \hat{\mathcal{T}}} \mathbf{y}_{\mathcal{G}}^{\top} \log(\mathbf{z}_{\mathcal{G}'}), \tag{14}$$

where $\mathbf{z}_{\mathcal{G}'}$ is the prediction from a classifier Φ on "implicit intervened graph" $\mathcal{G}'$; $\mathbf{h}_{\mathcal{G}_c}$ is the representation of causal attended-graph $\mathcal{G}_c$ derived from Equation (9); while $\mathbf{h}_{\mathcal{G}_{t'}}$ is the representation of stratification $\mathcal{G}_{t'}$ obtained via Equation (10); $\hat{\mathcal{T}}$ is the estimated stratification set of the trivial attended-graph, which collects the appearing trivial features from training data. In practice, we apply random addition to make the intervention in Equation (13). We define the Equation (14) as the causal intervention loss. It pushes the predictions of such intervened graphs to be invariant and stable across different stratifications, due to the shared causal features. Finally, the objective of CAL can be defined as the sum of the losses:

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \lambda_1 \mathcal{L}_{\text{unif}} + \lambda_2 \mathcal{L}_{\text{caus}} \tag{15}$$

where λ_1 and λ_2 are hyper-parameters that determine the strength of disentanglement and causal intervention, respectively. The detailed algorithm of CAL is provided in Appendix A.1, Alg.1, and the overview of CAL is depicted in Figure 2.

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 in alleviating the out-of-distribution (OOD) issue?

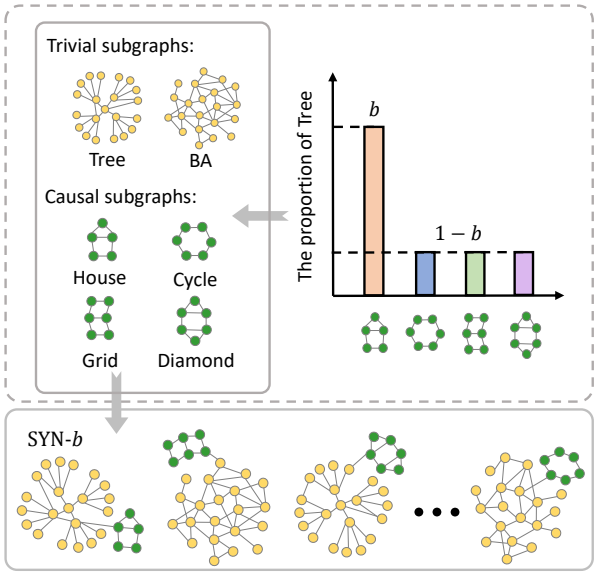

Figure 3: Illustration of the synthetic datasets.

- **RQ2:** Can the proposed CAL achieve performance improvements on real-world datasets?
- **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 graphs:** Following [44], 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 3, each sample consists of two parts: causal subgraph and trivial subgraph. More details about the causal and trivial subgraph are provided in Appendix A.2. 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:

$$b = \frac{\#\text{Tree-House}}{\#\text{House}} \tag{16}$$

  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. Please refer to Appendix A.2 for more details.
- **Real-world graphs:** We conduct experiments on three biological datasets (MUTAG, NCI1, PROTEINS), three social datasets (COLLAB, IMDB-B, IMDB-M) [24], and two superpixel datasets (MNIST, CIFAR-10) [18]. More details, such as statistics and splitting of datasets, are provided in Appendix A.2.

4.1.2 ***Baselines.*** To verify the superiority of CAL, we adopt the following prevalent graph classification solutions as baselines:

- **Attention-based methods:** GAT [35], GATv2 [4], SuperGAT [16], GlobalAttention [22], AGNN [33].
- **Pooling-based methods:** SortPool [49], DiffPool [45], Top-k Pool [10], SAGPool [19].
- **Kernel-based methods:** Graphlet kernel (GK) [31], Weisfeiler Lehman Kernel (WL) [30], Deep Graph kernels (DGK) [42].
- **GNN-based methods:** GCN [17], GIN [41]

Besides these methods, we also consider the state-of-the-art algorithms: IRM [2] and DRO [29], which are particularly designed for OOD issues. Please note that these methods require specific environments or group annotations for each training example, therefore we consider them as the methods with upper bound performance.

4.1.3 ***Hyper-parameters.*** All training hyper-parameters and model configurations are summarized in Appendix A.3. Codes are released at https://github.com/yongduosui/CAL.

4.2 Performance on Synthetic Graphs (RQ1)

To explore whether CAL can alleviate the OOD issue, we first conduct experiments on SYN-b with different biases: $b \in \{0.1, 0.2, ..., 0.9\}$. The experimental results are summarized in Table 1 and Figure 4. We have the following **Obs**ervations:

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 the proposed CAL, this phenomenon is significantly alleviated, which verifies the effectiveness of CAL in overcoming the OOD issue.

Table 1: Test Accuracy (%) of graph classification on synthetic datasets with diverse biases. The number in brackets represents the performance degradation compared with the unbiased dataset. Our methods are highlighted with a gray background.

Method	SYN-0.1	SYN-0.3	Unbiased	SYN-0.7	SYN-0.9
GATv2 [4]	87.25 (↓ 7.37%)	92.19 (↓ 2.12%)	94.19	93.31 (↓ 0.93%)	90.62 (↓ 3.79%)
SuperGAT [16]	83.81 (↓ 12.75%)	91.94 (↓ 4.29%)	96.06	88.50 (↓ 7.89%)	82.81 (↓ 13.79%)
GlobalAtt [22]	87.19 (↓ 10.40%)	93.75 (↓ 3.66%)	97.31	94.62 (↓ 2.76%)	91.50 (↓ 5.97%)
AGNN [33]	84.56 (↓ 11.69%)	93.06 (↓ 2.81%)	95.75	94.81 (↓ 0.98%)	88.12 (↓ 7.97%)
DiffPool [45]	82.28 (↓ 8.69%)	88.02 (↓ 2.32%)	90.11	88.83 (↓ 1.42%)	84.50 (↓ 6.23%)
SortPool [49]	80.70 (↓ 14.24%)	92.33 (↓ 1.88%)	94.10	92.14 (↓ 2.08%)	90.35 (↓ 3.99%)
Top-k Pool [10]	84.31 (↓ 11.81%)	93.53 (↓ 2.17%)	95.60	94.44 (↓ 1.21%)	88.02 (↓ 7.93%)
SAGPool [19]	88.08 (↓ 7.82%)	90.86 (↓ 4.91%)	95.55	92.22 (↓ 3.49%)	83.99 (↓ 12.10%)
GCN [17]	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%)
GIN [41]	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%)
GAT [35]	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%)

Table 2: Test Accuracy (%) of classification. For TUDataset, we perform 10-fold cross-validation and report the mean and standard derivations. Our methods are highlighted with gray background. If the performance improves, the number is bolded.

Dataset	MUTAG	NCI1	PROTEINS	COLLAB	IMDB-B	IMDB-M	MNIST	CIFAR-10
GK [31]	81.58±2.11	62.49±0.27	71.67±0.55	72.84±0.28	65.87±0.98	43.89±0.38	-	-
WL [30]	82.05±0.36	82.19±0.18	74.68±0.50	79.02±1.77	73.40±4.63	49.33±4.75	-	-
DGK [42]	87.44±2.72	80.31±0.46	75.68±0.54	73.09±0.25	66.96±0.56	44.55±0.52	-	-
GlobalAtt [22]	88.27±8.65	81.17±1.04	72.60±4.37	81.48±1.46	69.10±3.80	51.40±2.91	-	-
AGNN [33]	79.77±8.54	79.96±2.37	75.66±3.94	81.10±2.39	73.10±4.07	49.73±3.72	-	-
DiffPool [45]	85.61±6.22	75.06±3.66	76.25±4.21	79.24±1.66	74.47±3.84	49.20±3.10	-	-
SortPool [49]	86.17±7.53	79.00±1.68	75.48±1.62	77.84±1.22	73.00±3.50	49.53±2.29	-	-
GCN [17]	88.20±7.33	82.97±2.34	75.65±3.24	81.72±1.64	73.89±5.74	51.53±3.28	90.49	54.68
GCN + CAL	**89.24±8.72**	**83.48±1.94**	**76.28±3.65**	**82.08±2.40**	**74.40±4.55**	**52.13±2.96**	**94.58**	**56.21**
GIN [41]	89.42±7.40	82.71±1.52	76.21±3.83	82.08±1.51	73.40±3.78	51.53±2.97	96.51	56.36
GIN + CAL	**89.91±8.34**	**83.89±1.93**	**76.92±3.31**	**82.68±1.25**	**74.13±5.21**	**52.60±2.36**	**96.93**	**56.63**
GAT [35]	88.58±7.54	82.11±1.43	75.96±3.26	81.42±1.41	72.70±4.37	50.60±3.75	95.53	64.22
GAT + CAL	**89.94±8.78**	**83.55±1.42**	**76.39±3.65**	**82.12±1.95**	**73.30±4.16**	**50.93±3.84**	**95.91**	**66.16**

Obs 3: Mitigating the confounder achieves more stable performance on OOD datasets. 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 $b \in \{0.1, 0.2, ..., 0.9\}$. As depicted in Figure 4, 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 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, GIN (red curve) 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, and the effectiveness of CAL. For comprehensive comparisons, we also plot two upper bound methods: IRM and DRO (dash lines), which require additional annotation information of trivial subgraphs for training. We observe that, even without additional information, CAL achieves comparable performance with these upper bound methods.

4.3 Performance on Real-world Graphs (RQ2)

Unlike synthetic graphs, there may not exist visible or specific patterns of the causal/trivial subgraphs in real-world graphs. However, there still exist irregular core-subgraphs [18, 23, 38, 44] that determine the predictions, which will inevitably involve different degrees of biases caused by the complementary parts. Similar to SYN-b, they mislead the GNNs to learn the spurious correlations. Hence, we verify the practicability of CAL on eight real-world

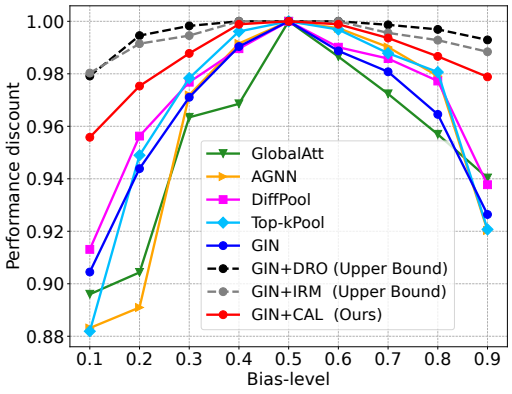

Figure 4: The performance discount on synthetic datasets with different bias-levels.

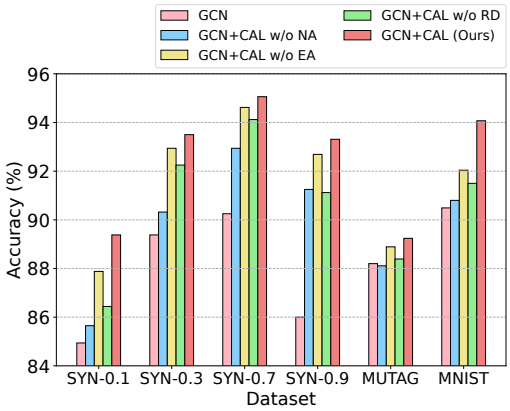

Figure 5: The comparison of different components in CAL.

datasets. We report the results of the baselines from the original papers by default and reproduce the missing results. The results are summarized in Table 2 and we make the following **Obs**ervations:

Obs 4: The OOD issue is widespread in real-world datasets. Attention-based and pooling-based methods are on a par with GNNs, and they both outperform graph kernel-based methods in most cases. It can be seen from the last six rows in Table 2, when CAL is applied to different GNN models, it consistently produces further performance improvements. It demonstrates that the distribution shifts also widely exist in real-world datasets. Specifically, we can find that GCN often performs worse than other GNNs, attention-based or pooling-based methods, while the performance significantly improves after adopting CAL. For instance, on IMDB-B and MNIST datasets, GCN+CAL achieves 1.92% and 4.52% relative improvements, respectively. This indicates that GCN is vulnerable to the distribution shift in certain datasets. Thanks to the causality, CAL will push GCN to pay more attention to causal features, which can establish robustness against the widespread OOD issues and achieve better generalization.

4.4 Ablation Study (RQ3)

In this section, we investigate the impact of the node/edge attention, random combination and the loss coefficients λ_1 and λ_2.

Node Attention v.s. Edge Attention. Node Attention (NA) and Edge Attention (EA) refine the features from two orthogonal 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 node/edge attention scores in Equation (7)/(8) are evenly set as 0.5. The experimental results are shown in Figure 5. 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.

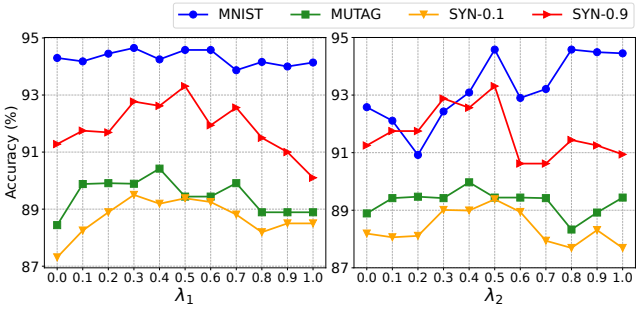

Figure 6: Parameter sensitivity of loss coefficients λ_1 and λ_2.

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 Addition" module in Figure 2 to "Addition", which just adopts the addition operation orderly, and we rename it as "GCN+CAL w/o RD". The experimental results are shown in Figure 5. 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.

Loss coefficients λ_1 and λ_2. According to Equation (15), λ_1 denotes the strength of the disentanglement for the causal/trivial features, while λ_2 controls the strength of the causal intervention. To explore their impacts, we use GCN as the encoder and conduct experiments on two biased synthetic datasets and two real-world datasets. We fix one coefficient as 0.5 and change the other one in $(0, 1)$ with a step size of 0.1. The experimental results are shown in Figure 6. We can find that: (1) λ_1 achieves better performance in a range of $0.3 \sim 0.7$. Too small or too large values will cause performance degradation. (2) λ_2 is not as stable as λ_1. The optimal range is around $0.3 \sim 0.5$. It leads to a strong decline at $0.5 \sim 0.8$, which indicates that coefficient λ_2 should be set prudently.

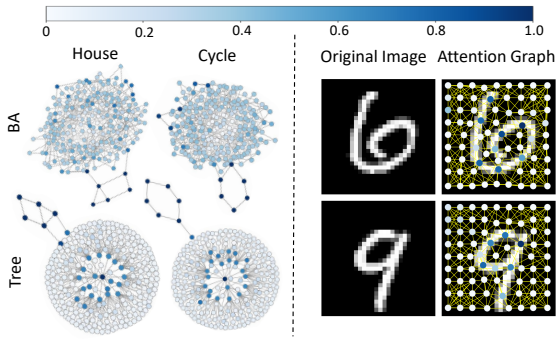

Figure 7: Visualizations of causal attended-graphs. (Left): Synthetic graphs, (Right): MNIST superpixel graphs.

4.5 Visualization and Analysis (RQ4)

Causal attended-graphs. We plot node/edge attention areas of the causal attended-graphs based on the attention scores in CAL. We adopt a GCN-based encoder and apply CAL on SYN-*b* and MNIST superpixel graphs. The visualizations are shown in Figure 7. Nodes with darker colors and edges with wider lines indicate higher attention scores. We surprisingly find that almost all the darker colors and wider lines precisely distribute on the deterministic areas, such as the causal subgraphs we defined in the synthetic dataset and the nodes located on digit pixels in MNIST superpixel graphs. It further demonstrates that the proposed CAL can effectively capture the causal features with insightful interpretations.

The explanation for performance improvements. Figure 8 displays the distribution of misclassification on SYN-*b*. The abscissa represents the predictions, and the ordinate denotes the ground-truth types. The numbers in each row denote the proportion for each class. Figure 8 (Left) shows that the wrong predictions of graphs with "BA" are mainly distributed in "Cycle", "Grid" and "Diamond" classes, while the wrong predictions of graphs with "Tree" mainly concentrate on the "House" class (highlighted by the red circle). On one hand, most of the "House" co-occur with "Tree" in the training data, GCN tends to capture the shortcut features, *e.g.*, "Tree" patterns, to make decisions. Therefore, the other three causal subgraphs with "Tree" will mainly be misclassified as "House" in the testing set. On the other hand, only a few "House" causal subgraphs co-occur with "BA", so the other three causal subgraphs with "BA" will almost not be misclassified as "House". In contrast, Figure 8 (Right) shows that, by applying CAL, the concentration of misclassification is obviously alleviated. This demonstrates that CAL improves performance by mitigating the confounding effect.

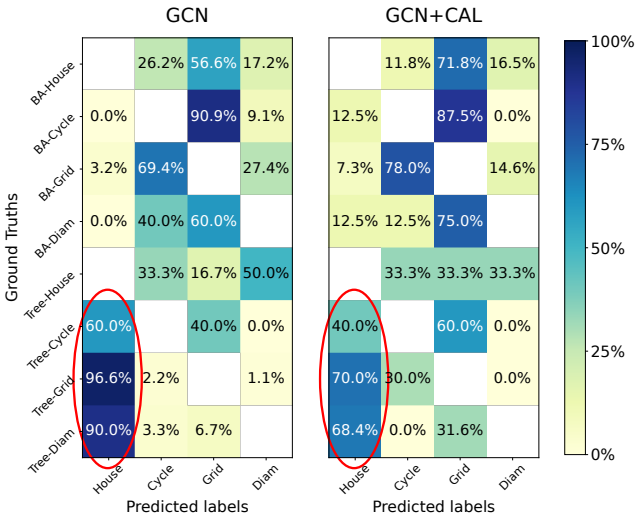

Figure 8: The misclassification distribution. Red circle highlights the concentration degree of misclassification.

5 RELATED WORK

Attention Mechanism selects the informative features from data, which has obtained great success in computer vision [7, 14, 36, 43] and natural language processing tasks [6, 34]. In recent years, attention mechanism has gradually become prevalent in the GNN field. The attention modules for GNNs can be defined over edges [4, 16, 21, 33, 35] or over nodes [19, 20, 22]. Despite effectiveness, attention learning still stays at how to better fit the statistical correlations between data and labels. Hence, the learned attentions are inherently biased in OOD settings. Recent studies [36, 43] propose the causal attention modules to alleviate the bias. CaaM [36] adopts the adversarial training to generate the data partition in each iteration to achieve the causal intervention. CATT [43] proposes in-sample and cross-sample attentions based on front-door adjustment. However, they are both tailored for computer vision tasks, while 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.

OOD Generalization [2, 13, 28, 29] has been extensively explored in recent years. IRM [2] minimizes the empirical risk under different environments. Group-DRO [29] adversarially explores the group with the worst risk and achieves generalization by minimizing the empirical risk of the worst group. Existing efforts [2, 28, 29] mainly focus on computer vision or natural language processing tasks, while the GNN field is of great need but largely unexplored. Furthermore, these methods require the environment or group prior information for each training sample, which is expensive in practice. To alleviate this dilemma, we adopt causal intervention to strengthen the causal relationship between the causal feature and prediction, thereby achieving better generalization.

Causal Inferences [26, 27] endows the model with the ability to pursue real causality. A growing number of studies [15, 25, 32, 48] have shown that causal inference is beneficial to diverse computer vision tasks. CONTA [48] uses backdoor adjustment to eliminate the confounder in weakly supervised semantic segmentation tasks. DDE [15] 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 [9] explores how to select trustworthy neighbors for GNN in the inference stage, and demonstrates its effectiveness in node classification. Recent work [47] 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 perspective. 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 causal attention learning (CAL) strategy for GNNs. CAL 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. Future studies include adopting powerful disentanglement methods and more advanced causal intervention strategies to improve the CAL. We will also make efforts to apply CAL to other graph learning tasks, such as node classification or link prediction.

7 ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2020AAA0106000), and the National Natural Science Foundation of China (U19A2079, U21B2026). This research is also supported by CCCD Key Lab of Ministry of Culture and Tourism and Sea-NExT Joint Lab.

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 TPAMI* 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] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.
[4] Shaked Brody, Uri Alon, and Eran Yahav. 2022. How Attentive are Graph Attention Networks?. In *ICLR*.
[5] 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. *Journal of medicinal chemistry* 34, 2 (1991).
[6] 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.
[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.
[8] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2020. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982* (2020).
[9] 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.
[10] Hongyang Gao and Shuiwang Ji. 2019. Graph u-nets. In *ICML*. 2083–2092.
[11] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence* 2, 11 (2020), 665–673.
[12] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. 2019. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*.
[13] Dan Hendrycks and Kevin Gimpel. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *ICLR*.
[14] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *CVPR*.
[15] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. 2021. Distilling Causal Effect of Data in Class-Incremental Learning. In *CVPR*.
[16] Dongkwan Kim and Alice Oh. 2020. How to find your friendly neighborhood: Graph attention design with self-supervision. In *ICLR*.
[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.
[18] Boris Knyazev, Graham W. Taylor, and Mohamed R. Amer. 2019. Understanding Attention and Generalization in Graph Neural Networks. In *NeurIPS*. 4204–4214.
[19] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. In *ICML*. 3734–3743.
[20] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. 2018. Graph classification using structural attention. In *SIGKDD*. 1666–1674.
[21] 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.
[22] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated Graph Sequence Neural Networks. In *ICLR*.
[23] Wanyu Lin, Hao Lan, and Baochun Li. 2021. Generative causal explanations for graph neural networks. In *ICML*. 6666–6679.
[24] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. 2020. Tudataset: A collection of benchmark datasets for learning with graphs. *ICMLW*.
[25] 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.
[26] Judea Pearl. 2014. Interpretation and identification of causal mediation. *Psychological methods* 19, 4 (2014), 459.
[27] Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: Cambridge University Press* 19 (2000).
[28] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. 2020. The Risks of Invariant Risk Minimization. In *ICLR*.
[29] 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*.
[30] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. 2011. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research* 12, 9 (2011).
[31] Nino Shervashidze, S. V. N. Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten M. Borgwardt. 2009. Efficient graphlet kernels for large graph comparison. In *AISTATS*.
[32] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. 2020. Long-Tailed Classification by Keeping the Good and Removing the Bad Momentum Causal Effect. In *NeurIPS*.
[33] 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).
[34] 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.
[35] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.
[36] Tan Wang, Chang Zhou, Qianru Sun, and Hanwang Zhang. 2021. Causal Attention for Unbiased Visual Recognition. In *CVPR*. 3091–3100.
[37] Xiang Wang, Yingxin Wu, An Zhang, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2022. Reinforced Causal Explainer for Graph Neural Networks. *TPAMI* (2022).
[38] Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat seng Chua. 2021. Towards Multi-Grained Explainability for Graph Neural Networks. In *NeurIPS*.
[39] Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. 2022. Discovering Invariant Rationales for Graph Neural Networks. In *ICLR*.
[40] 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*.
[41] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR*.
[42] Pinar Yanardag and SVN Vishwanathan. 2015. Deep graph kernels. In *SIGKDD*.
[43] Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. 2021. Causal attention for vision-language tasks. In *CVPR*. 9847–9857.
[44] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating Explanations for Graph Neural Networks. In *NeurIPS*. 9240–9251.
[45] 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.
[46] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In *SIGKDD*. 430–438.
[47] 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).
[48] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. 2020. Causal Intervention for Weakly-Supervised Semantic Segmentation. In *NeurIPS*.
[49] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An end-to-end deep learning architecture for graph classification. In *AAAI*.

A APPENDIX

A.1 Algorithm

We provide the detailed implementation of the proposed casual attention learning (CAL) in Algorithm 1. We adopt the causal attended-graph for prediction in the inference stage.

Algorithm 1: Casual Attention Learning

Input: Dataset $\mathcal{D}$, $f(\cdot)$, attention modules, classifiers, λ_1, λ_2
Output: The trained parameters.

for sampled M graphs $\{\mathcal{G}^k = \{\mathbf{A}^k, \mathbf{X}^k\}\}_{k=1}^{M}$ **do**
  for $k \leftarrow 1$ **to** M **do**
    $\mathbf{H}^k \leftarrow f(\mathbf{A}^k, \mathbf{X}^k)$
    Compute $\alpha_{c_i}, \alpha_{t_i} \leftarrow$ Equation (7) for all nodes of $\mathcal{G}^k$
    Compute $\beta_{c_{ij}}, \beta_{t_{ij}} \leftarrow$ Equation (8) for all edges of $\mathcal{G}^k$
    Get masks $\mathbf{M}_a^k$ and $\mathbf{M}_x^k$ based on $\beta_{c_{ij}}$ and α_{c_i}
    Get masks $\overline{\mathbf{M}_a^k}$ and $\overline{\mathbf{M}_x^k}$ based on $\beta_{t_{ij}}$ and α_{t_i}
    $\mathcal{G}_c^k \leftarrow \{\mathbf{A}^k \odot \mathbf{M}_a^k, \mathbf{X}^k \odot \mathbf{M}_x^k\}$ `// causal`
    $\mathcal{G}_t^k \leftarrow \{\mathbf{A}^k \odot \overline{\mathbf{M}_a^k}, \mathbf{X}^k \odot \overline{\mathbf{M}_x^k}\}$ `// trivial`
    $\mathbf{h}_{\mathcal{G}_c^k}, \mathbf{z}_{\mathcal{G}_c^k} \leftarrow$ Equation (9)
    $\mathbf{h}_{\mathcal{G}_t^k}, \mathbf{z}_{\mathcal{G}_t^k} \leftarrow$ Equation (10)
  end for
  Supervised loss: $\mathcal{L}_{\text{sup}} \leftarrow$ Equation (11)
  Uniform loss: $\mathcal{L}_{\text{unif}} \leftarrow$ Equation (12)
  $I \leftarrow$ Shuffle($[1, ..., M]$)
  for $k \leftarrow 1$ **to** M **do**
    $i \leftarrow I[k]$
    $\mathbf{h}_{\mathcal{G}^k} \leftarrow \mathbf{h}_{\mathcal{G}_c^k} + \mathbf{h}_{\mathcal{G}_t^i}$ `// random combination`
    $\mathbf{z}_{\mathcal{G}^k} \leftarrow \Phi(\mathbf{h}_{\mathcal{G}^k})$
  end for
  Causal loss: $\mathcal{L}_{\text{caus}} \leftarrow$ Equation (14)
  Total loss: $\mathcal{L} \leftarrow \mathcal{L}_{\text{sup}} + \lambda_1 \mathcal{L}_{\text{unif}} + \lambda_2 \mathcal{L}_{\text{caus}}$
  Update all the trainable parameters to minimize $\mathcal{L}$
end for

A.2 Datasets Details

In this section, we give more details about the synthetic datasets and real-world datasets.

1) Synthetic graphs. For each synthetic graph instance, it consists of two subgraphs: trivial and critical subgraphs. We introduce the proposed trivial subgraph and critical subgraph as follows:

- **Trivial subgraph**. There exist two types of trivial subgraphs: BA-SHAPES and Tree. The BA-SHAPES is a Barabási-Albert (BA) graph [3], and we abbreviate it as "BA" in this paper. The "Tree" graph is a base 12-level balanced binary tree [44]. To reduce the influence, we control the number of nodes in the two kinds of trivial subgraphs to be similar.
- **Causal subgraph**. There are four types of causal subgraphs: "House", "Cycle", "Grid", "Diamond". The visualizations of these trivial subgraphs and causal subgraphs are depicted in Figure 3.

For each synthetic graph instance, a causal subgraph is randomly attached on one node of a trivial subgraph. Then the resulting graph is further perturbed by adding 10% random edges. We take the one-hot form of the node degree as the node feature and set the dimension of node feature to 20. The synthetic graph examples are displayed in Figure 3. The statistics of the synthetic datasets are summarized in Table 3. We split the dataset into training, validation and testing set with the ratio of 7: 1: 2.

Table 3: Statistics of datasets used in experiments.

Dataset	#Graphs	#Nodes	#Edges	#Classes
SYN-b	8000	230~247	542~1000	4
MUTAG	188	17.93	19.79	2
NCI1	4110	29.87	32.30	2
PROTEINS	1113	39.06	72.82	2
COLLAB	5000	74.49	2457.78	3
IMDB-B	1000	19.77	96.53	2
IMDB-M	1500	13.00	65.94	3
MNIST	70000	70.57	564.66	10
CIFAR-10	60000	117.63	941.04	10

2) Real-world graphs. To demonstrate the practicality of the proposed CAL, we conduct experiments on TUDataset [24] and Superpixel graphs [18]. For TUDataset, we gather three biological datasets (MUTAG, NCI1, PROTEINS) and three social networks datasets (COLLAB, IMDB-B, IMDB-M), which are commonly used in graph classification benchmarks [8, 41]. Following [8, 41, 45], we use 10-fold cross-validation and report average accuracy and standard deviation. The superpixel graphs [8, 18] includes MNIST and CIFAR-10, which are classical image classification datasets converted into graphs using superpixels technology [1] and assigning each node's features as the superpixel coordinates and intensity. Following [8, 18], we split the MNIST and CIFAR-10 to 55K training/5K validation/10K testing, and 45K training/5K validation/10K testing, respectively. All the detailed statistics about the real-world datasets are summarized in Table 3.

A.3 Hyper-parameters

As for training parameters, we train the models for 100 epochs with batch size of 128. We optimize all models with the Adam optimizer. For SYN-b and TUDataset, we use GCN, GIN and GAT as GNN encoders with 3 layers and 128 hidden units. For Superpixel graphs MNIST and CIFAR-10, we use the GNN encoders with 4 layers and 146 hidden units as [8]. For all the baselines, we follow the default settings from original papers and reproduce the missing results. For the proposed CAL, we search λ_1 and λ_2 in $(0.1, 1.0)$ with a step size of 0.1 and report the results with the best settings. We adopt NVIDIA 2080 Ti (11GB GPU) to conduct all our experiments, the training time comparison is shown as Table 4.

Table 4: Training time (minutes) comparison.

Method	SYN-b	MUTAG	NCI1	IMDB-M	MNIST
GCN	4.16	1.03	12.71	4.61	57.20
GCN + CAL	6.67	1.35	17.37	6.16	75.80