Front. Comput. Sci., 2025, 19(8): 198338
https://doi.org/10.1007/s11704-024-40452-4

RESEARCH ARTICLE

Learning from shortcut: a shortcut-guided approach for explainable graph learning

Linan YUE[1], **Qi LIU** (✉)[1,2], **Ye LIU**[1], **Weibo GAO**[1], **Fangzhou YAO**[1]

1 State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei 230026, China
2 Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230026, China

Abstract The remarkable success in graph neural networks (GNNs) promotes the explainable graph learning methods. Among them, the graph rationalization methods draw significant attentions, which aim to provide explanations to support the prediction results by identifying a small subset of the original graph (i.e., *rationale*). Although existing methods have achieved promising results, recent studies have proved that these methods still suffer from exploiting shortcuts in the data to yield task results and compose rationales. Different from previous methods plagued by shortcuts, in this paper, we propose a Shortcut-guided Graph Rationalization (SGR) method, which identifies rationales by learning from shortcuts. Specifically, SGR consists of two training stages. In the first stage, we train a *shortcut guider* with an early stop strategy to obtain shortcut information. During the second stage, SGR separates the graph into the rationale and non-rationale subgraphs. Then SGR lets them learn from the shortcut information generated by the frozen *shortcut guider* to identify which information belongs to shortcuts and which does not. Finally, we employ the non-rationale subgraphs as environments and identify the invariant rationales which filter out the shortcuts under environment shifts. Extensive experiments conducted on synthetic and real-world datasets provide clear validation of the effectiveness of the proposed SGR method, underscoring its ability to provide faithful explanations.

Keywords explainable graph learning, graph rationalization, shortcut learning

1 Introduction

Graph Neural Networks (GNNs) have gained widespread adoption in various applications and have demonstrated high performance [1–4]. One prominent category of applications is graph classification tasks, such as predicting molecular graph properties [5–7]. Despite their success, GNNs in graph classification tasks often lack explainability and reliability in their prediction results. This limitation has motivated researchers [8,9] to explore explainable graph learning methods for providing explanations for GNNs. Among these methods, graph rationalization techniques [10,11] have gained increasing attention. These methods aim to produce task results while identifying a small subset of the original graph, known as the *rationale*. This rationale typically consists of significant nodes or edges in the graph. By extracting such a rationale, these methods provide an explanation for the prediction results obtained by GNNs. The extracted rationale serves as a concise representation that highlights the key elements or components influencing the prediction outcome.

Despite the appeal of graph rationalization methods, recent studies [12,13] have indicated that these approaches are susceptible to exploiting shortcuts (also known as spurious correlations) in the data to yield task results and compose rationales. This exploitation of shortcuts can potentially lead to the derivation of invalid or erroneous conclusions, thereby undermining the trustworthiness and reliability of the outputs produced by the model.

Considering Fig. 1, we predict the motif type based on the graph that consists of motifs and bases subgraphs. In the training dataset, the *Cycle*-motifs are frequently co-occurring with the *Tree* bases and *House*-motifs are predominantly accompanied by the *Wheel* bases, which may mislead the GNNs over-reliance on these associations for achieving high accuracy, rather than discerning the true relationships between critical subgraphs (i.e., *rationales*) and the predicted labels. For example, GNNs may predict the motif type as *Cycle* when identifying the *Tree* bases or classify the motif type as *House* when recognizing the *Wheel* bases. However, this dependency on biases can result in inaccuracies when faced out-of-distribution (OOD) data (e.g., the test dataset in Fig. 1), such as incorrectly predicting a *Cycle*-motif with *Wheel* bases as a *House* or misclassifying *House*-motifs with *Tree* as *Cycles*.

In order to address this issue, a range of methods [14–16] have emerged in recent times, aiming to construct genuine rationales by capturing the invariant relationship that exists between rationales and their corresponding labels. These methods contend that the underlying rationale behind the labels remains consistent even when subjected to different environments. As a result, they utilize environment inference

Received May 7, 2024; accepted August 18, 2024

E-mail: qiliuql@ustc.edu.cn

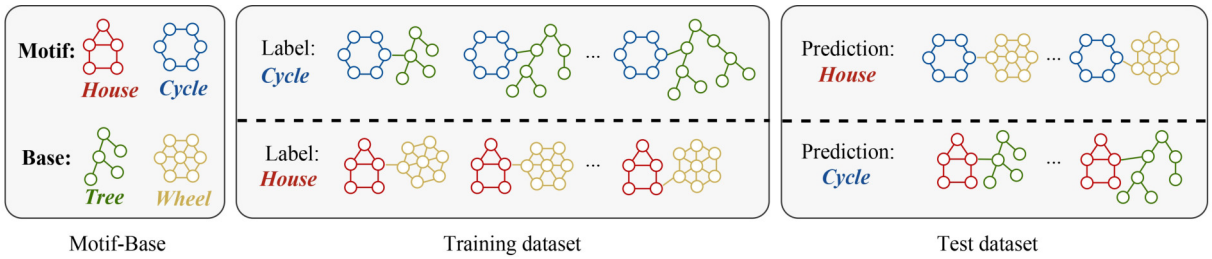


Fig. 1 An example of the motif type prediction, where the *Cycle* and *House* are motif labels, and *Tree* and *Wheel* are bases that are irrelevant to the motif prediction. In the training dataset, the data distribution is *Cycle* with *Tree* and *House* with *Wheel*. When the model depends too much on this data distribution (i.e., shortcuts) for prediction, the model is likely to misclassify when facing the test dataset with a shift in the distribution

techniques to derive diverse latent environments and subsequently identify the invariant rationales that remain unchanged under shifts in the environment. By leveraging this approach, these methods aim to mitigate the impact of shortcuts and enhance the robustness and reliability of the rationales extracted from the data.

These methods are all based on the assumption that shortcuts are unknown. However, a direct approach is to explicitly identify which nodes in the graph are shortcuts, enabling us to use these shortcut nodes to train a de-biased model. Unfortunately, annotating nodes for shortcuts in each graph can indeed be a labor-intensive task. However, an interesting alternative approach is to obtain latent shortcut representations, even without explicit knowledge of the shortcut nodes. Previous research [14,17–19] has shown that shortcut features are easier for models to learn compared to rationale features. Therefore, we can get an important assumption:

Assumption 1 During the initial stages of training, the features learned by the model are more likely to capture the features of shortcuts [20].

Based on this assumption, we can employ an early stopping strategy to obtain the shortcut representations. By doing so, we can capture the features learned by the model in the early stages, which are more likely to align with the shortcuts present in the data.

Along this line, in this paper, we propose a **S**hortcut-guided **G**raph **R**ationalization (SGR) method, which identifies significant nodes as rationales by learning from shortcuts. Specifically, our method involves two stages. In the first stage, we train a *shortcut guider* which is designed to intentionally capture the shortcut in data with the early stop strategy. In the second stage, we first freeze the trained *shortcut guider* and adopt it to generate the shortcut representation. Then, we separate the original input graph into rationale and non-rationale subgraphs, which are respectively encoded into representations. Next, we employ the *shortcut guider* to eliminate the shortcut information from the rationale subgraphs by minimizing the Mutual Information (MI) [21–23] between the shortcut and rationale representation. Meanwhile, we also let the *shortcut guider* encourage the non-rationale subgraphs and shortcut representations to encode the same information by maximizing MI [24]. Based on the MI estimation methods, rationale and non-rationale subgraph representations can fully learn which information belongs to shortcuts and which does not. Finally, to further identify the invariant rationales under environment shifts, we consider non-rationale representations which sufficiently capture the shortcut information as the environment. We then combine each rationale representation with various non-rationale representations, and encourage these combinations to maintain a stable prediction and yield rationales. Experiments over ten datasets, including various synthetic [8,13] and OGBG [5] benchmark datasets, validate the effectiveness of our proposed SGR.

2 Related work

Graph rationalization The application of Graph Neural Networks (GNNs) to graph classification tasks has demonstrated significant achievements [6,25,26]. However, the interpretability of the prediction results remains a challenge, rendering many GNN models unreliable. To address this issue, recent studies [8,9,27] have proposed post-hoc methods to explain the prediction results of GNNs. These methods aim to provide explanations for the predictions made by GNNs after the models have been trained. By adopting post-hoc approaches, researchers seek to enhance the interpretability and transparency of GNN predictions, improving the trustworthiness and reliability of GNNs.

In contrast to the post-hoc methods mentioned, recent research has focused on inherently explainable methods [28–32] specifically designed for GNNs in graph classification tasks. Among them, graph rationalization methods have received considerable attention. However, recent studies [12] have demonstrated that rationalization methods tend to exploit shortcuts within the data to generate predictions and construct rationales, leading to potential inaccuracies. In response to this concern, a pioneering study [13] introduced the concept of discovering invariant rationales by creating multiple environments. The researchers initially divided the graph into rationale and non-rationale subgraphs, and then utilized the non-rationale subgraphs as distinct environments. By analyzing how rationales remain consistent across different environments, they identified invariant rationales that are unaffected by shifts in the environment. This approach aims to mitigate the influence of shortcuts and enhance the reliability of the extracted rationales in graph classification tasks. Various recent works [14–16,33] have followed this framework. The difference is that they consider non-rationale subgraph representations as potential environments not the explicit non-rationale subgraph structures. Along another line of research, information bottleneck theory [34–38] was

introduced into the rationalization. Among them, GSAT [39] constrained the information flow from the input graph to the prediction and learned stochasticity-reduced attention to yield rationales.

Although most methods are effective in removing shortcuts and discovering rationales, few consider incorporating shortcuts information into the model, enabling the model to learn which information belongs to shortcuts and which is not.

Shortcut learning Shortcut learning [40,41] refers to the phenomenon where deep neural networks heavily rely on spurious correlations in the data as shortcuts to make predictions. While methods employing shortcuts can achieve high performance on identically distributed datasets, they often fail to capture the true underlying correlations between the input and the label. Therefore, when faced with out-of-distribution (OOD) data, their performance tends to degrade.

To address this issue, researchers have proposed various approaches [42–44]. Among them, [44,45] leveraged adversarial training to learn debiased representations, aiming to reduce the reliance on shortcuts and uncover the genuine correlations in the data. [33,46] partitioned the data into different environments and formulate predictions that are robust to shifts in the environment. [47] introduced the product-of-expert method, which involves training a bias-only model to obtain a debiased model that mitigates the influence of shortcuts. These techniques aim to enhance the generalization capability of models, improve performance on OOD data, and uncover the true causal relationships between inputs and labels by combating the issue of shortcut learning in deep neural networks.

3 Shortcut-guided graph rationalization

In this section, after describing the problem definition of graph rationalization, we present the detailed architecture of SGR, including the *shortcut guider*, *selector*, *predictor* and the strategy of learning from shortcuts. Finally, we provide a comprehensive overview of the training and inference procedures adopted within the SGR framework.

3.1 Problem definition

Considering graph classification tasks, given an input graph instance $g=(\mathcal{V},\mathcal{E})$ with N nodes and Z edges and its graph-level ground truth y, where $(g,y)\in\mathcal{D}_G$, $\mathcal{D}_G$ is the dataset, $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges and the matrix $A\in\{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$, our goal is first to yield a rationale mask vector $\mathbf{M}\in\mathbb{R}^N$ that represents the probability of each node being selected as the rationale. Then, the rationale subgraph is calculated as $\mathbf{h}_r=\mathrm{READOUT}\left(\mathbf{M}\odot\mathrm{GNN}_g(g)\right)$, where $\mathrm{GNN}_g(\cdot)$ can be any GNN encoder (e.g., GIN [48]). Finally, the rationale representation $\mathbf{h}_r$ is employed to yield task results. Take the case in Fig. 1 for example, our goal is to predict the motif type while identifying the *Cycle* or *House* structure as the rationale to support the prediction results.

3.2 Architecture of SGR

To explicitly utilize the shortcut information to compose unbiased rationales, we propose the SGR method consisting of two stages. ***In the first stage***, based on Assumption 1, we employ an early stopping strategy to obtain a *shortcut guider* that can fully learn the shortcut information. ***In the second stage***, as shown in Fig. 2, SGR involves a *shortcut guider*, *selector*, and *predictor*. Initially, we freeze the *shortcut guider* and further obtain the shortcut representation. We then adopt the *selector* to separate the original graph into rationale and non-rationale representations. Next, we use the MI estimation method to transfer the generated shortcut information to the rationale and non-rationale representation, ensuring these representations can learn from shortcuts. Finally, the *predictor* yields prediction results based on the above rationale and non-rationale representation cooperatively.

3.2.1 Shortcut guider

While discerning the precise identification of shortcut nodes poses a challenge, we make the assumption that representations of shortcuts are accessible. Specifically, previous research [14,18,19] suggests that shortcut features are easier to learn than rationale features, indicating that the features learned in the initial training stages are more inclined to shortcuts [20]. Consequently, in the initial stage, we deliberately train the *shortcut guider* to effectively capture the pertinent shortcut information, employing an early stop strategy to optimize its performance. Initially, we train the *shortcut guider* on the dataset $\mathcal{D}_G$ to predict the graph label:

$$\begin{aligned}\mathbf{H}_s&=\mathrm{GNN}_s(g),\\ \mathbf{h}_s&=\mathrm{READOUT}(\mathbf{H}_s),\\ \hat{y}_s&=\Phi_s(\mathbf{h}_s).\end{aligned}\tag{1}$$

Among them, $\mathrm{GNN}(\cdot)$ can be any GNN encoder such as GCN [1]. $\mathbf{H}_s\in\mathbb{R}^{N\times d}$ denotes the node representation, and $\mathbf{h}_s\in\mathbb{R}^d$ is the graph-level representation that is generated by a readout operator (employing mean pooling in this paper). $\Phi_s(\cdot)$ is a classifier which is applied to project $\mathbf{h}_s$ to the graph label. Then, the prediction loss can be defined as:

$$\mathcal{L}_s=\mathbb{E}_{(g,y)\sim\mathcal{D}_G}\left[l(\hat{y}_s,y)\right],\tag{2}$$

where $l(\cdot)$ is the cross entropy loss. We then train the *shortcut guider* only for a few epochs (e.g., 3 epochs) to ensure the *shortcut guider* capture more shortcut information rather than rationale information. Finally, we freeze the parameters of the *shortcut guider* and apply it to the second stage.

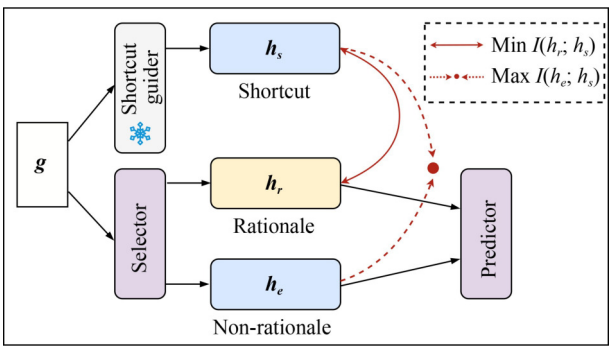


Fig. 2 Architecture of SGR in the second stage, including the *selector*, *predictor* and the freezed *shortcut guider*

3.2.2 Selector

To separate the original input into rationale and non-rationale subgraphs, the *selector* first generates $\mathbf{M} \in \mathbb{R}^N$ that represents the probability of each node being selected as the rationale [33,49]:

$$\mathbf{M} = \sigma(\Phi_m(\mathrm{GNN}_m(g))), \tag{3}$$

where $\Phi_m(\cdot)$ encodes each node into a value of selecting the node as the rationale, and $\sigma(\cdot)$ denotes the sigmoid function, indicating the probability of nodes being the rationale.

Then, the *selector* employs another GNN encoder to obtain the node representation $\mathbf{H}_g = \mathrm{GNN}_g(g)$. Next, the rationale node representation can be defined as $\mathbf{M} \odot \mathbf{H}_g$, while the non-rationale node representation is formulated as $(1-\mathbf{M}) \odot \mathbf{H}_g$. Finally, the rationale subgraphs representation $\mathbf{h}_r$ and the non-rationale ones $\mathbf{h}_e$ can be obtained by a READOUT operation:

$$\begin{aligned} \mathbf{h}_r &= \mathrm{READOUT}\left(\mathbf{M} \odot \mathbf{H}_g\right), \\ \mathbf{h}_e &= \mathrm{READOUT}\left((1-\mathbf{M}) \odot \mathbf{H}_g\right). \end{aligned} \tag{4}$$

3.2.3 Learning from shortcut by MI estimation

To eliminate the shortcut information in the rationale and alleviate the problem of employing shortcuts in the data for prediction, we adopt the *shortcut guider* to reduce the mutual information (MI) between rationale subgraphs representations and shortcut representations. Among them, MI is employed to measure the mutual dependence of two random variables (e.g., X and Y) in the probability theory and information theory:

$$I(X;Y) = \mathbb{E}_{p(x,y)}\left[\log \frac{p(x,y)}{p(x)p(y)}\right]. \tag{5}$$

To achieve the goal of the *shortcut guider*, we first input the original graph g into the *selector* to obtain the subgraphs representations $\mathbf{h}_r$ and $\mathbf{h}_e$, respectively, as described in Section 3.2.2. We then keep the *shortcut guider* frozen and employ it to generate shortcut representations $\mathbf{h}_s$. Next, we employ the MI minimization method to ensure that the shortcut information can be removed from the rationale (i.e., min $I(\mathbf{h}_r;\mathbf{h}_s)$), where $I(;)$ denotes the MI, and MI is a measure of the mutual dependence between the two variables.

Meanwhile, we employ the MI maximization method to facilitate the matching of non-rationale representations with shortcut representations (i.e., max $I(\mathbf{h}_e;\mathbf{h}_s)$), with the goal of enabling the full learning of shortcut information. Then, we consider the matched non-rationale representations as the environment and apply them to the *predictor*. Finally, the objective of learning from shortcut is

$$\mathcal{L}_{shortcut} = I(\mathbf{h}_r;\mathbf{h}_s) - I(\mathbf{h}_e;\mathbf{h}_s). \tag{6}$$

In the implementation phase, calculating the MI values becomes challenging when dealing with high-dimensional random variables. To overcome this issue, we resort to estimating the lower and upper bounds of MI in order to optimize MI maximization and minimization.

For MI maximization tasks, we utilize the InfoNCE method proposed by Oord et al. [24]:

$$\begin{aligned} I_{nce} &= \frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{f(x_i,y_i)}}{\frac{1}{N}\sum_{j=1}^{N} e^{f(x_i,y_j)}} \\ &= \frac{1}{N}\sum_{i=1}^{N} f(x_i,y_i) - \frac{1}{N}\sum_{i=1}^{N}\left[\log \frac{1}{N}\sum_{j=1}^{N} e^{f(x_i,y_j)}\right], \end{aligned} \tag{7}$$

where $\{(x_i,y_i)\}_{i=1}^N$ represents a batch of sample pairs of (X,Y). This method enables us to estimate a lower-bound approximation of the mutual information and serves as our approach for achieving MI maximization.

For MI minimization tasks, we adopt CLUB_NCE [23] to achieve MI minimization, where CLUB_NCE is a variant of CLUB [22] that is designed to estimate the upper bound of MI. To be specific, the criterion of CLUB is

$$I_{club} = \frac{1}{N}\sum_{i=1}^{N} \log p(y_i|x_i) - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \log p(y_j|x_i), \tag{8}$$

where $p(y|x)$ is a conditional distribution. Further, [23] develop a new MI minimization method CLUB_NCE which combines InfoNCE and CLUB. CLUB_NCE first adopts the trained $f(x,y)$ by InfoNCE to replace $\log(p(y|x))$ in CLUB. Then, it calculates the value of I_{club} based on the trained $f(x,y)$ and minimizes I_{club} to achieve MI minimization. Detailed description of CLUB_NCE can be found in [23].

3.2.4 Predictor

Within the *predictor* module, we first adopt the rationale representation to predict the graph label, employing the cross entropy loss function denoted as $\mathcal{L}_r = \mathbb{E}_{(g,y)\sim\mathcal{D}_G}[l(\hat{y}_r, y)]$. Here, $\hat{y}_r = \Phi_p(\mathbf{h}_r)$, with $\Phi_p(\cdot)$ representing a shared classifier. To ensure the generation of invariant rationales amidst environmental shifts, we consider the non-rationale representations as distinct environments, drawing inspiration from previous works [14,33]. Specifically, we transfer a batch of sample pairs $\left\{\left(g^i, y^i\right)\right\}_{i=1}^K$ to their respective representations $\left\{\left(\mathbf{h}_r^i, \mathbf{h}_e^i, y^i\right)\right\}_{i=1}^K$. Subsequently, as the non-rationale (i.e., environment) components do not influence the task prediction, we combine each rationale representation $\mathbf{h}_r^i$ with all non-rationale representations $\mathbf{h}_e^j$ $(\mathbf{h}_e^j \neq \mathbf{h}_e^i)$ within the same batch. This process facilitates the induction of environment shifts:

$$\mathbf{h}^{i,j} = \mathbf{h}_r^i + \mathbf{h}_e^j. \tag{9}$$

It is important to note that the corresponding labels remain unchanged as the rationale information in the synthetic data remains unaltered. Subsequently, we proceed by inputting the merged graph representations into the shared classifier $\Phi_p(\cdot)$ to obtain the task-related outcomes. The loss is then computed using the cross entropy function:

$$\hat{y}^{i,j} = \Phi_p(\mathbf{h}^{i,j}), \mathcal{L}_e = \mathbb{E}_i\left[\mathbb{E}_j\left[l(\hat{y}^{i,j}, y)\right]\right]. \tag{10}$$

Finally, in order to ensure consistent predictions across different environments and address potential instability resulting from environmental changes between augmented and original data, we commence by quantifying the differences between $\hat{y}_r^i$ and $\hat{y}^{i,j}$ (i.e., $\mathcal{D}_f(\hat{y}_r^i;\hat{y}^{i,j})$), where $\mathcal{D}_f(\cdot)$ represents any distance function, such as squared Euclidean distance.

Subsequently, to align the predicted distributions across environments with those predicted using rationale representations, we minimize the mean and variance of these differences:

$$\mathcal{L}_{diff} = \mathbb{E}_i\left[\mathbb{E}_j\left[\mathcal{D}_f(\hat{y}_r^i;\hat{y}^{i,j})\right] + \mathrm{Var}_j\left[\mathcal{D}_f(\hat{y}_r^i;\hat{y}^{i,j})\right]\right]. \quad (11)$$

3.3 Training and inference

During the training process, in order to encourage the model to regulate the anticipated size of rationale subgraphs, we adopt a sparsity constraint on the probability $\mathbf{M}$ pertaining to the selection as rationale, inspired by the approach outlined in [33].

$$\mathcal{L}_{sp} = \left|\frac{1}{N}\sum_{i=1}^{N} M_i - \alpha\right|, \quad (12)$$

where $\alpha \in [0,1]$ is the predefined sparsity level. The objective of SGR in the second stage is

$$\mathcal{L}_{sgr} = \mathcal{L}_r + \mathcal{L}_e + \lambda_{diff}\mathcal{L}_{diff} + \lambda_{shortcut}\mathcal{L}_{shortcut} + \lambda_{sp}\mathcal{L}_{sp}, \quad (13)$$

where λ_{diff}, $\lambda_{shortcut}$, and λ_{sp} are hyperparameters. Detailed training process is present in Algorithm 1. At inference time, only $\mathbf{h}_r$ is employed to yield the task results.

Algorithm 1 Training process of SGR

Initialize the early stop epoch $\mathcal{E}_{early}$ and the regular epoch $\mathcal{E}_{SGR}$, where $\mathcal{E}_{early} \ll \mathcal{E}_{SGR}$.
In the first stage of SGR:
 # Train the shortcut guider by the early stop.
for epoch e=1 **to** $\mathcal{E}_{early}$ **do**
  Update the *shortcut guider* by Eq.(2).
end for
Freeze the parameters of the *shortcut guider* and apply this guider to the second stage.
In the second stage of SGR:
 # Train SGR by learning from shortcuts.
for epoch e=1 **to** $\mathcal{E}_{SGR}$ **do**
  1. Partition the graph into the rationale $\mathbf{h}_r$ and non-rationale subgraph $\mathbf{h}_e$ with the *selector* by Eq.(4)
  2. Generate shortcut representations $\mathbf{h}_s$ with the freezed *shortcut guider*.
  3. Learn from the shortcuts by Eq.(6).
  4. Create the counterfactual samples with environment shifts by Eq.(9).
  5. The *predictor* yields the task results with both original and counterfactual samples.
  6. Align the predicted distributions across environments with those predicted using rationale by Eq.(11).
  7. Control the expected size of rationale subgraphs with a sparsity constraint (Eq.(12)).
end for

4 Experiments

In this section, to verify the reasonableness and effectiveness and of SGR, we design experiments to address the following research questions:

- **RQ1:** Do GNNs learn shortcuts during the initial training?
- **RQ2:** How does SGR perform in terms of improving task prediction and rationale extraction?
- **RQ3:** What are the roles and impacts of different components within SGR on its overall performance?
- **RQ4:** Can the "*learning from shortcuts*" framework enhance the performance of existing rationale-based methods?
- **RQ5:** Does SGR effectively capture significant rationales for predictions within the given dataset?

By addressing these research questions, we aim to provide comprehensive insights into the performance and capabilities of SGR, thereby verifying its viability and effectiveness in the context of graph rationalization.

4.1 Datasets

Here, we make experiments on four synthetic datasets and six real-world benchmark datasets to evaluate the performance of our proposed approach for graph rationalization. Details of dataset statistics are summarized in Tables 1 and 2.

- **Spurious-Motif** [8,13] is a synthetic dataset for predicting the motif category of each graph. Each graph consists of two subgraphs, the motif subgraph (*Cycle*, *House*, *Crane* denoted by $R = 0, 1, 2$, respectively) and the base one (*Tree*, *Ladder*, *Wheel* denoted by $E = 0, 1, 2$ respectively). Among them, the motif subgraph is regarded as the ground-truth explanation (i.e., rationale) for the graph label, which suggests the

Table 1 Statistics of Spurious-Motif and Graph-SST2 datasets

	Spurious-Motif				Graph-SST2
	$b = 0.5$	$b = 0.7$	$b = 0.9$	*Cycle-Tree*	
Train/Val/Test	3,000/3,000/6,000	3,000/3,000/6,000	3,000/3,000/6,000	4,000/4,000/6,000	28,327/3,147/12,305
Classes	3	3	3	3	2
Avg. Nodes	29.6	30.8	29.4	28.9	13.7
Avg. Edges	42.0	45.9	42.5	45.1	25.3

Table 2 Statistics of OGBG datasets

	MolHIV	MolToxCast	MolBACE	MolBBBP	MolSIDER
Train/Val/Test	32,901/4,113/4,113	6,860/858/858	1,210/151/152	1,631/204/204	1,141/143/143
Classes	2	617	2	2	27
Avg. Nodes	25.5	18.8	34.1	24.1	33.6
Avg. Edges	27.5	19.3	36.9	26.0	35.4

graph label is solely determined by the motif subgraph. The base subgraph can be considered as the non-rationale (or environment). To verify the effectiveness of SGR, we manually generate several datasets containing shortcuts.

Specifically, we construct the training dataset by sampling each motif uniformly, while controlling the distribution of the base through

$$P(E) = b \times \mathbb{I}(E = R) + \frac{1-b}{2} \times \mathbb{I}(E \neq R), \tag{14}$$

where the degree of spurious correlation is controlled by b. In this paper, we set $b = \{0.5, 0.7, 0.9\}$.

Besides, to verify the shortcut whether will be captured in the initial training stages, we first create a balance dataset (i.e., $b = \frac{1}{3}$, where each motif contains 1,000 training instances, for a total of 3,000 instances.). Then, based on this balance dataset, we intentionally conduct additional 1,000 instances that are all *Cycle* motifs with *Tree* bases, achieving the spurious correlations in *Cycle-Tree*. In the test dataset, we match the motif and base randomly ($b = \frac{1}{3}$) to construct an unbiased test dataset.

- **Graph-SST2** [50] is a text sentiment analysis dataset, where each text instance in SST2 is converted to a graph. Following [13,14], to create distribution shifts, we divide the graphs into different sets according to their average node degrees, where the node degrees in the training set are higher than degrees in the test set. By dividing the graphs based on node degrees, the dataset aims to create variations in the graph characteristics between the training and test sets, allowing for the evaluation of models under distribution shifts.
- **Open Graph Benchmark (OGBG)** [5] is a benchmark dataset for machine learning on graphs, where we consider five OGBG-Mol datasets that are employed for molecular property prediction (i.e., MolHIV, MolToxCast, MolBACE, MolBBBP and MolSIDER). Each dataset focuses on a specific molecular property prediction task. We split datasets into train, validation and test set. The default approach is to divide them based on scaffolds. Scaffolds are molecular substructures that serve as a structural framework for molecules. The goal of using scaffolds is to ensure that the splits contain molecules with different structural features.

4.2 Comparison methods

First, we compare SGR with classical GNNs methods GCN [1] and GIN [48]. We then compare several competitive baselines specifically designed for explainable GNNs:

- **DIR** [13] discovers invariant rationales by separating the graph as the rationale subgraphs and the non-rationale ones. Different from SGR, DIR takes a different approach by explicitly generating multiple environments through the utilization of non-rationale subgraphs.
- **DisC** [14] proposes a comprehensive disentangling framework to learn the causal and bias substructure by partitioning the graph into rationale and non-rationale subgraphs. Then, DisC employs a general approach that incorporates the synthesis of counterfactual training samples to further de-correlate the causal and bias variables.
- **RGDA** [49] proposes a general counterfactual data augmentation of the graph node classification and graph-level classification. In this paper, we employ RGDA for the graph-level classification. It employs each non-rationale as the environment to create the counterfactual samples. Different from SGR, RGDA ignores the instability of the prediction results between the counterfactual data and the original data due to environmental changes.
- **CAL** [15] introduces the Causal Attention Learning strategy as a novel approach for discovering causal rationales and mitigating the confounding effect of shortcuts by disentangling the graph into rationale and non-rationale subgraphs. Although DisC, RGDA, and CAL all compose rationales by taking non-rationale subgraphs representations as environments, there exists serval difference among them. Specifically, DisC selects edges as rationales, RGDA identifies nodes as rationales, and CAL considers both edges and nodes.
- **GSAT** [39] injects stochasticity to the attention weights to block the information from non-rationale subgraphs. Meanwhile, it learns stochasticity-reduced attention to select rationale subgraphs for explainability based on the information bottleneck principle [34,35].
- **DARE** [23] proposes a disentanglement-augmented method to extract rationales. Meanwhile, it introduces CLUB_NCE to improve MI minimization. Although DARE is designed for explaining natural language understanding tasks [10,51,52]. It can naturally be applied to explain GNNs.

Besides, in our experiments, we implement all of explainable baselines (**DIR** [13], **DisC** [14], **RGDA** [49], **CAL** [15], **GSAT** [39], and **DARE** [23]) based on their released codes by employing both GCN [1] and GIN [48] as the graph encoder, respectively.

4.3 Experimental setup

Metrics In this paper, following the metric setting of [13,14], we employ ACC to evaluate the task prediction performance for Spurious-Motif and Graph-SST2, ROC-AUC for OGBG. Furthermore, considering that the Spurious-Motif dataset contains ground-truth rationales, the Precision@5 metric is adopted to assess the disparity between the predicted rationales and the actual rationales. This particular metric quantifies the precision of the top 5 extracted rationales in comparison to the ground truth rationales.

Optimization and hyperparameters Across all our experiments, we have maintained the following parameter settings: λ_{diff}, $\lambda_{shortcut}$, and λ_{sp} are set to 0.1, 0.01, and 1.0, respectively. The hidden dimensionality, denoted as d, is configured as 32 for the Spurious-Motif dataset, 64 for the

Graph-SST2 dataset, and 128 for the OGBG dataset. Regarding the Adam optimizer [53], we initialize the learning rate as 1e–2 for both the Spurious-Motif and Graph-SST2 datasets, while for the OGBG dataset, the learning rate is set to 1e–3. For the predefined sparsity parameter, denoted as α, we set it as 0.1 for the MolHIV dataset, 0.5 for the MolSIDER, MolToxCast, and MolBBBP datasets, and 0.4 for the remaining datasets. The early stop epoch is set to 2 or 3 for the Spurious-Motif dataset and 3 or 4 for both the Graph-SST2 and OGBG datasets.

We employ the squared Euclidean distance as $\mathcal{D}_f(\cdot)$. All methods are trained using five different random seeds on a single A100 GPU.

How to decide the epoch of the early stop strategy? In the process of determining the epoch for the early stop strategy, a specific range of epochs, [1,5], is initially defined. Subsequently, for the Spurious-Motif dataset, the selection of the epoch is based on the results depicted in Fig. 3. From these results, either epoch 2 or epoch 3 is chosen. To further refine the selection, a grid search is conducted to identify the optimal epoch for the early stop strategy. Through this iterative process, it is determined that epoch 2 yields the best results for the Spurious-Motif dataset. Similarly, for the Graph-SST2 and OGBG datasets, following the grid search, either epoch 3 or epoch 4 is selected as the most suitable epoch for both the Graph-SST2 and OGBG datasets.

4.4 Shortcuts capturing (RQ1)

Due to the inability to explicitly identify which nodes serve as shortcuts, a hypothesis is proposed that assumes shortcut features are relatively easier to learn compared to rationale features. To validate this assumption, a series of experiments are conducted in this section. These experiments aim to provide empirical evidence and insights regarding the ease of learning shortcut information compared to rationale information within the given context. First, [18,20] have empirically proved:

Theorem 1 If the malignant bias is easier to learn than the real relationship between the input and label, the neural network tends to memorize it first.

As a type of neural networks, we argue that GNNs also adhere to the theorem mentioned. Therefore, in line with Theorem 1, we propose that by demonstrating shortcuts as malignant biases and the fact that shortcuts are easier to learn than the actual relationships between the input and the label, we can infer that these shortcuts are more likely to be captured during the initial training phase. To validate this hypothesis, we conduct experiments on the Spurious-Motif dataset. In addition to the unbiased test set, we construct biased test sets with varying degrees of bias (b) to match the biases present in the corresponding training sets. Subsequently, we apply GCN and GIN models on these datasets. The results, as shown in Fig. 4, indicate that GCN and GIN achieve promising results (almost 100% accuracy) on the biased test set. However, when evaluating on the unbiased test set, the performance of GCN and GIN exhibits a significant degradation. These findings suggest that the shortcuts present in the Spurious-Motif dataset are indeed malignant biases. Furthermore, as the degree of bias (b) increases (ranging from 0.5 to 0.9), the performance of GNNs on the biased test set improves, while the performance on the unbiased test set declines. This trend illustrates that GNNs are more inclined to make predictions based on shortcuts rather than the actual relationships between the input and the label. Consequently, based on the above conclusions, these findings further support the notion that GNNs tend to learn shortcuts during the initial training phase. In summary, the experimental results on the Spurious-Motif dataset provide evidence that GNNs capture shortcuts as malignant biases, prioritize them over the actual relationships, and are more likely to learn these shortcuts during the initial stages of training.

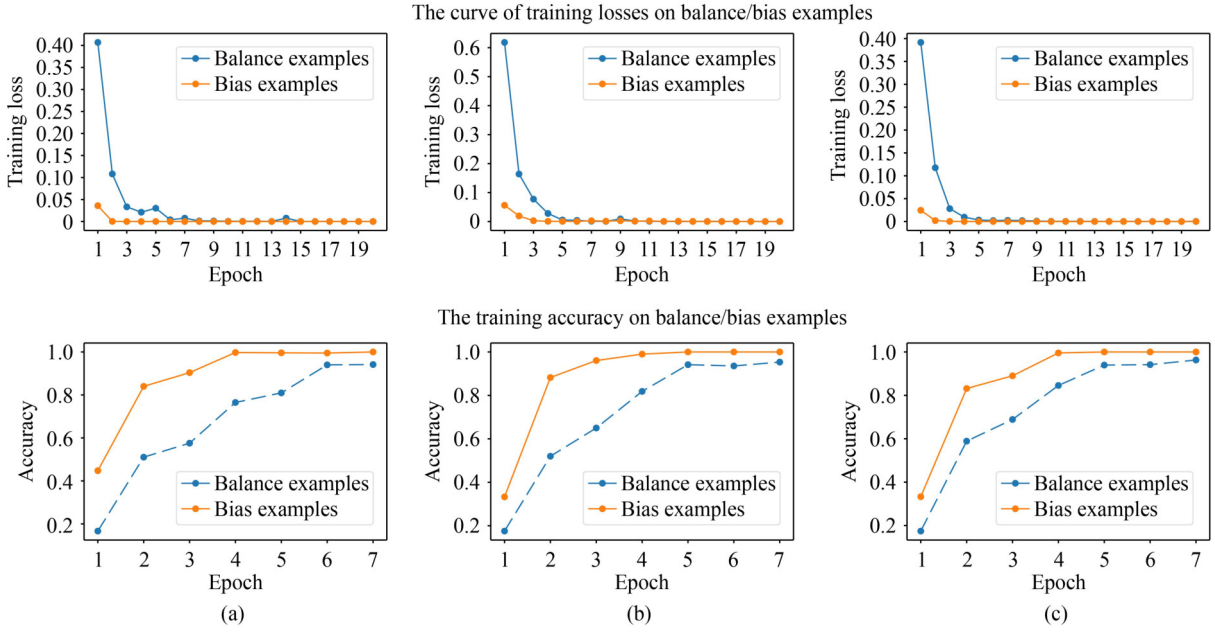


Fig. 3 Training losses and accuracy on balance and bias examples with different datasets. (a) *Cycle-Tree*; (b) *Cycle-Ladder*; (c) *Cycle-Wheel*

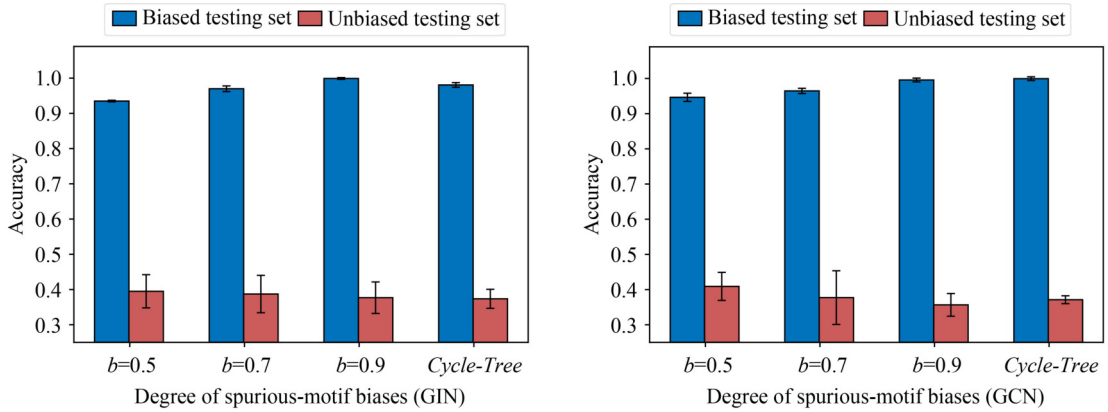


Fig. 4 Accuracy of GIN and GCN on the unbiased and biased test set. GIN and GCN achieve promising results on the biased test set, but perform badly on the unbiased test set

Besides, we conduct additional experiments to demonstrate that shortcuts are captured during the early epochs of training. Specifically, we carry out experiments on the Spurious-Motif dataset, focusing on the *Cycle-Tree* scenario. As described in Section 4.1, this dataset comprises balanced data and biased data. The biased data consists of *Cycle* motifs accompanied by *Tree* bases. Figure 3(a) illustrates the training loss curve for both the balanced and biased examples. It can be observed that after only 2 epochs of training, the training loss for the biased examples approaches zero, while the loss for the balanced examples does not converge. This indicates that the biased examples are significantly easier to learn compared to the balanced examples during the training process. These findings provide compelling evidence that shortcuts are indeed captured early on during the training phase, as demonstrated by the rapid convergence of the biased examples' training loss compared to the more persistent loss of the balanced examples.

Furthermore, we also conduct experiments on two additional synthetic datasets. These datasets are structured such that the bias data consists of *Cycle* motifs accompanied by *Ladder* bases (referred to as *Cycle-Ladder*), and *Cycle* motifs accompanied by *Wheel* bases (referred to as *Cycle-Wheel*). The results obtained from these experiments show a similar pattern to the observations in Fig. 3(a), as depicted in Figs. 3(b) and 3(c). These consistent findings indicate that the bias features are generally easier to learn compared to the rationale features across different dataset variations. This further strengthens the empirical evidence supporting the assumption made in the study regarding the relative ease of learning shortcut information compared to rationale information.

Meanwhile, we examine the variation of Accuracy for both balance and bias examples throughout the training phase. Figure 3(a) illustrates this variation, revealing an interesting observation. It is evident that the Accuracy of bias examples reaches a high performance level early on in the training process, while the balance examples require a greater number of epochs to achieve a comparable level of performance. This discrepancy implies that the model is able to learn and generalize the patterns present in bias examples more quickly and effectively compared to the more complex patterns exhibited by the balance examples.

Moreover, we also conduct experiments on the real-world dataset (MolBACE). Since we cannot determine which data belongs to the biased category in this dataset, we focus on demonstrating the effectiveness of the *shortcut guider*. To achieve this, we obtain different *shortcut guiders* at various training epochs ranging from 1 to 10 during the first stage of SGR. Subsequently, these trained *shortcut guiders* are incorporated into SGR during the second stage, and the results are presented in Fig. 5. From the figure, it can be observed that the model's performance initially improves as the number of epochs increases. This indicates that the *shortcut guider* successfully captures shortcut features during the early stages of training. However, after surpassing the third epoch, the effectiveness of SGR gradually diminishes. This suggests that the *shortcut guider* gradually transitions from capturing shortcut features to rationale features as the training progresses. The above observations confirm that shortcut information is more likely to be learned in the early stages of training.

4.5 Overall performance on both synthetic and real-world datasets (RQ2)

To assess the effectiveness of SGR, we first conduct a comparative analysis with several baseline methods on task prediction. The results of these experiments are presented in Tables 3 and 4. Upon analyzing the results, we observe that SGR outperforms classical GCN and GIN in terms of task

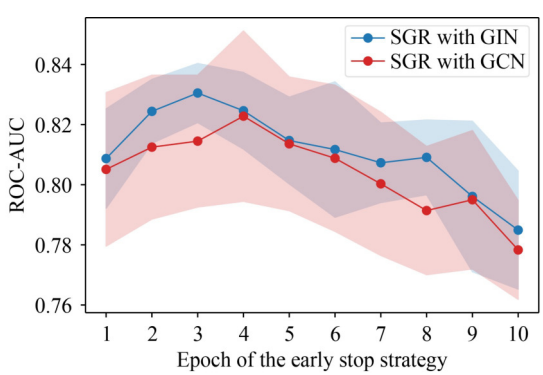


Fig. 5 Performance of SGR with different shortcut guiders that are trained with the early stop strategy

Table 3 The graph classification ROC-AUC on testing datasets of OGBG

		MolHIV	MolToxCast	MolBACE	MolBBBP	MolSIDER
GIN is the backbone	GIN	0.7447 ± 0.0293	0.6521 ± 0.0172	0.8047 ± 0.0172	0.6584 ± 0.0224	0.5977 ± 0.0176
	DIR	0.6303 ± 0.0607	0.5451 ± 0.0092	0.7391 ± 0.0282	0.6460 ± 0.0139	0.4989 ± 0.0115
	DisC	0.7731 ± 0.0101	0.6662 ± 0.0089	0.8293 ± 0.0171	0.6963 ± 0.0206	0.5846 ± 0.0169
	RGDA	0.7714 ± 0.0153	0.6694 ± 0.0043	0.8187 ± 0.0195	0.6953 ± 0.0229	0.5864 ± 0.0052
	CAL	0.7339 ± 0.0077	0.6476 ± 0.0066	0.7848 ± 0.0107	0.6582 ± 0.0397	0.5965 ± 0.0116
	GSAT	0.7524 ± 0.0166	0.6174 ± 0.0069	0.7021 ± 0.0354	0.6722 ± 0.0197	0.6041 ± 0.0096
	DARE	0.7836 ± 0.0015	0.6677 ± 0.0058	0.8239 ± 0.0192	0.6820 ± 0.0246	0.5921 ± 0.0260
	SGR	**0.7945 ± 0.0071**	**0.6723 ± 0.0061**	**0.8305 ± 0.0098**	**0.7021 ± 0.0190**	**0.6092 ± 0.0288**
GCN is the backbone	GCN	0.7128 ± 0.0188	0.6497 ± 0.0114	0.8135 ± 0.0256	0.6665 ± 0.0242	0.6108 ± 0.0075
	DIR	0.4258 ± 0.1084	0.5077 ± 0.0094	0.7002 ± 0.0634	0.5069 ± 0.1099	0.5224 ± 0.0243
	DisC	0.7791 ± 0.0137	0.6626 ± 0.0055	0.8104 ± 0.0202	0.7061 ± 0.0105	0.6110 ± 0.0091
	RGDA	0.7816 ± 0.0079	0.6622 ± 0.0045	0.8044 ± 0.0063	0.6970 ± 0.0089	0.6133 ± 0.0239
	CAL	0.7501 ± 0.0094	0.6006 ± 0.0031	0.7802 ± 0.0207	0.6635 ± 0.0257	0.5559 ± 0.0151
	GSAT	0.7598 ± 0.0085	0.6124 ± 0.0082	0.7141 ± 0.0233	0.6437 ± 0.0082	0.6179 ± 0.0041
	DARE	0.7523 ± 0.0041	0.6618 ± 0.0065	0.8066 ± 0.0178	0.6823 ± 0.0068	0.6192 ± 0.0079
	SGR	**0.7822 ± 0.0079**	**0.6668 ± 0.0026**	**0.8228 ± 0.0283**	**0.7116 ± 0.0169**	**0.6217 ± 0.0291**

Table 4 The graph classification ACC on testing datasets of the Spurious-Motif and Graph-SST2

		Spurious-Motif				Graph-SST2
		$b = 0.5$	$b = 0.7$	$b = 0.9$	*Cycle-Tree*	
GIN is the backbone	GIN	0.3950 ± 0.0471	0.3872 ± 0.0531	0.3768 ± 0.0447	0.3736 ± 0.0270	0.8269 ± 0.0259
	DIR	0.4444 ± 0.0621	0.4891 ± 0.0761	0.4131 ± 0.0652	0.4039 ± 0.0425	0.8083 ± 0.0115
	DisC	0.4585 ± 0.0660	0.4885 ± 0.1154	0.3859 ± 0.0400	0.4882 ± 0.1007	0.8279 ± 0.0081
	RGDA	0.4251 ± 0.0458	0.5331 ± 0.1509	0.4568 ± 0.0779	0.3702 ± 0.0223	0.8301 ± 0.0088
	CAL	0.4734 ± 0.0681	0.5541 ± 0.0323	0.4474 ± 0.0128	0.4362 ± 0.0642	0.8181 ± 0.0094
	GSAT	0.4517 ± 0.0422	0.5567 ± 0.0458	**0.4732 ± 0.0367**	0.3769 ± 0.0108	0.8272 ± 0.0064
	DARE	0.4843 ± 0.1080	0.4002 ± 0.0404	0.4331 ± 0.0631	0.4527 ± 0.0562	0.8320 ± 0.0086
	SGR	**0.4941 ± 0.0968**	**0.5686 ± 0.1211**	0.4658 ± 0.0672	**0.5801 ± 0.1264**	**0.8386 ± 0.0077**
GCN is the backbone	GCN	0.4091 ± 0.0398	0.3772 ± 0.0763	0.3566 ± 0.0323	0.3712 ± 0.0012	0.8208 ± 0.0165
	DIR	0.4281 ± 0.0520	0.4471 ± 0.0312	0.4588 ± 0.0840	0.4325 ± 0.0583	0.8012 ± 0.0016
	DisC	0.4698 ± 0.0408	0.4312 ± 0.0358	0.4713 ± 0.1390	0.5058 ± 0.0476	0.8318 ± 0.0105
	RGDA	0.4687 ± 0.0855	0.5467 ± 0.0742	0.4651 ± 0.0881	0.5173 ± 0.0972	0.8269 ± 0.0077
	CAL	0.4245 ± 0.0152	0.4355 ± 0.0278	0.3654 ± 0.0064	0.4593 ± 0.0489	0.8127 ± 0.0077
	GSAT	0.3630 ± 0.0444	0.3601 ± 0.0419	0.3929 ± 0.0289	0.3474 ± 0.0031	0.8342 ± 0.0017
	DARE	0.4609 ± 0.0648	0.5035 ± 0.0247	0.4494 ± 0.0526	0.4576 ± 0.0737	0.8266 ± 0.0046
	SGR	**0.4715 ± 0.0515**	**0.5582 ± 0.0518**	**0.4762 ± 0.1135**	**0.5305 ± 0.1037**	**0.8378 ± 0.0059**

prediction performance and generalizability, showcasing the effectiveness of SGR in achieving improved task prediction and generalization capabilities.

Specifically, on the Spurious-Motif data, all methods are trained on the biased dataset and the results are reported based on the unbiased test set. From the experimental results, it can be found that SGR outperforms these base models by a large margin. Meanwhile, our model consistently performs better than GIN and GCN on both OGDB and Graph-SST2. Among them, SGR gains a 4.98% improvement over GIN and 6.94% improvement over GCN on the MolHIV dataset. Since SGR takes GCN and GIN as the backbone respectively, the experimental results suggest that our proposed method can well help existing GNNs to mitigate the negative impact of bias.

Then, SGR is also superior to de-biased baselines and performs well on most tasks, indicating the effectiveness of SGR. Among them, DIR performs poorly on most of the datasets, a possible reason is that it explicitly takes non-rationale subgraphs as environments, which loses some contextual information. In contrast, the DisC, RGDA, and CAL methods utilize latent non-rationale subgraph representations as environments, resulting in significant improvements compared to the DIR approach. This observation highlights the effectiveness of incorporating non-rationale representations. However, SGR still outperforms these approaches, indicating that introducing shortcuts during the training phase and allowing the model to learn from them is beneficial. On the other hand, the GSAT method, which does not consider the non-rationale information in the data, performs relatively averagely. This further emphasizes the advantage of incorporating non-rationale representations. DARE separates the graph into rationale and non-rationale subgraphs by minimizing mutual information (CLUB_NCE). However, DARE does not address the shortcut problem, resulting in its effectiveness being lower than that of SGR.

Finally, to further analyze whether SGR captures invariant rationales, we conduct additional experiments by comparing it with baseline methods on the Spurious-Motif dataset, which includes ground-truth rationales. Precision@5 is employed as the evaluation metric to measure the correspondence between the identified rationales and the actual ones. Specifically, the experimental results are presented in Fig. 6. From observations, we find that SGR outperforms other methods in

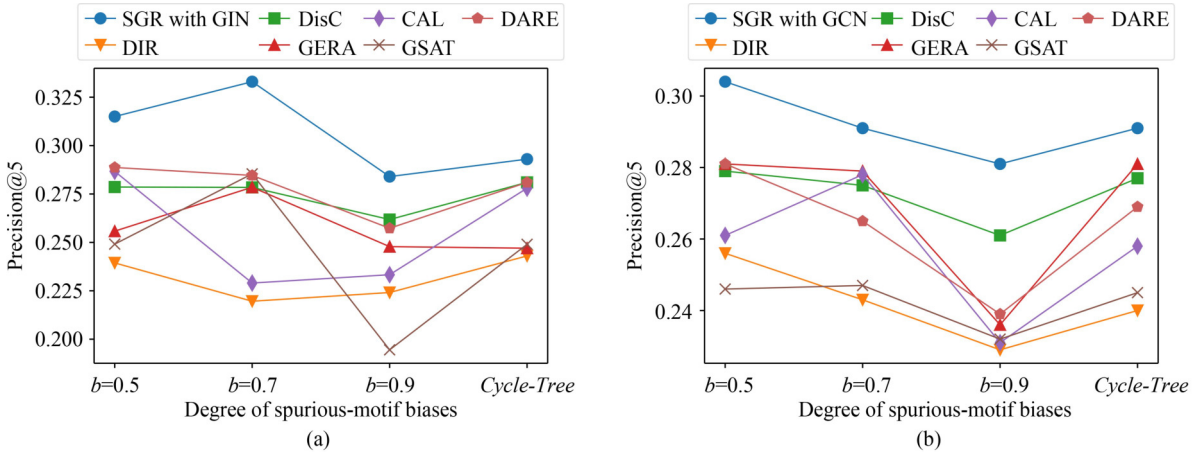


Fig. 6 The results of identifying the ground-truth rationale subgraphs on Spurious-Motif. (a) Precision@5 on Spurious-Motif with GIN as the graph encoder; (b) Precision@5 on Spurious-Motif with GCN as the graph encoder

identifying invariant rationales, irrespective of the varying degrees of shortcuts present in the data. These findings provide empirical evidence that SGR possesses a distinct advantage over alternative approaches when it comes to discovering invariant rationales, regardless of the extent of changes in the presence of shortcuts within the data.

4.6 Ablation studies (RQ3)

To verify the importance of the different components of the model, we construct ablation studies from three aspects:

(*i*). We remove the *shortcut guider* (i.e., we ablate $\mathcal{L}_{shortcut}$ in Eq. (13)). We name this variant as SGR w/o shortcut.

(*ii*). We remove $\mathcal{L}_{diff}$ (denoted by SGR w/o diff) to verify whether $\mathcal{L}_{diff}$ can make the predictions stable across different environments.

(*iii*). We ablate both $\mathcal{L}_e$ and $\mathcal{L}_{diff}$ (SGR w/o env) to demonstrate the effectiveness of non-rationale representations that are considered as environments.

Here, we make ablation studies on the OGBG dataset where SGR is implemented with GIN. As shown in Fig. 7, we observe that the performance of SGR w/o shortcut decreases significantly compared to the original SGR. This emphasizes the importance of learning from shortcut information in SGR. Furthermore, SGR w/o shortcut performs at a similar level to certain baselines, such as CAL, when shortcut information is not incorporated. This further highlights the effectiveness of learning from shortcuts. Additionally, when we remove the environment component (SGR w/o env) from SGR, which retains only the $\mathcal{L}_{shortcut}$ loss, we still observe improved performance compared to several baselines. This demonstrates the significance of our proposed shortcut guider in guiding the learning process. However, SGR w/o env is still less effective than the original SGR, indicating that incorporating non-rationale representations as environments is beneficial for composing rationales. Furthermore, we find that SGR w/o diff performs worse than the original SGR, with a decrease of 0.99% on the MolHIV dataset. This illustrates the instructive nature of the $\mathcal{L}_{diff}$ loss in identifying invariant rationales. These observations collectively demonstrate the importance of each component within SGR, with the incorporation of shortcut information, the presence of the environment component, and the utilization of the difference loss all contributing to the improved performance and effectiveness of the SGR model.

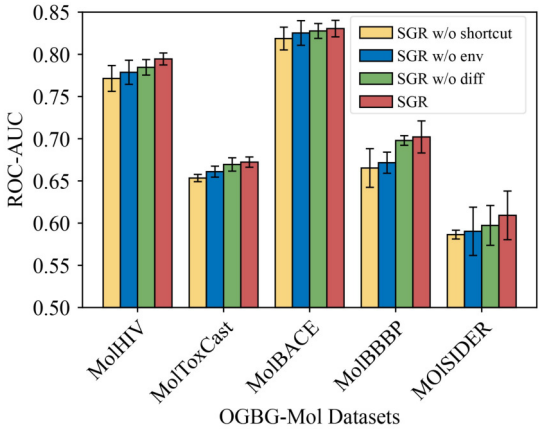


Fig. 7 Ablation studies of SGR with GIN over the OGBG dataset

4.7 Generalizability of the "*learning from shortcuts*" framework (RQ4)

In this paper, we introduce a framework known as "*learning from shortcuts*", which facilitates the learning process of rationale and non-rationale subgraph representations by effectively discerning information associated with shortcuts and distinguishing it from other information. We contend that the integration of our framework with existing rationale-based methods can enhance their performance, as elucidated in Section 3.2.3. To validate this claim, we conduct experiments on OGBG data, combining our framework with established rationalization baselines including DisC, RGDA, CAL, and DARE.

As shown in Table 5, the results consistently demonstrate that the incorporation of the "*learning from shortcuts*" framework leads to performance improvements across all rationale-based baselines. In particular, when GCN is employed as the backbone, CAL augmented with the "*learning from shortcuts*" framework achieves an average improvement of 2.82% on OGBG data compared to the original CAL. The above observation serves as compelling evidence substantiating the effectiveness of our proposed "*learning from shortcuts*" framework.

Table 5 Generalizability of the "*learning from shortcuts*" framework. Each rationalization method implemented with SGR is highlighted with a gray background

		MolHIV	MolToxCast	MolBACE	MolBBBP	MolSIDER
GIN is the backbone	DisC	0.7731	0.6662	0.8293	0.6963	0.5846
	DisC+SGR	0.7883 (↑ 1.52%)	0.6703 (↑ 0.41%)	0.8343 (↑ 0.50%)	0.6991 (↑ 0.28%)	0.5969 (↑ 1.23%)
	RGDA	0.7714	0.6694	0.8187	0.6953	0.5864
	RGDA+SGR	0.7878 (↑ 1.64%)	0.6775 (↑ 0.81%)	0.8256 (↑ 0.69%)	0.6970 (↑ 0.17%)	0.5938 (↑ 0.74%)
	CAL	0.7339	0.6476	0.7848	0.6582	0.5965
	CAL+SGR	0.7699 (↑ 3.59%)	0.6582 (↑ 1.06%)	0.8114 (↑ 2.66%)	0.6883 (↑ 2.93%)	0.6021 (↑ 0.56%)
	DARE	0.7836	0.6677	0.8239	0.6820	0.5921
	DARE+SGR	0.7901 (↑ 0.65%)	0.6698 (↑ 0.21%)	0.8296 (↑ 0.57%)	0.6947 (↑ 1.27%)	0.5998 (↑ 0.77%)
GCN is the backbone	DisC	0.7791	0.6626	0.8104	0.7061	0.6110
	DisC+SGR	0.7813 (↑ 0.22%)	0.6691 (↑ 0.65%)	0.8197 (↑ 0.93%)	0.7098 (↑ 0.37%)	0.6189 (↑ 0.79%)
	RGDA	0.7816	0.6622	0.8044	0.6970	0.6133
	RGDA+SGR	0.7856 (↑ 0.40%)	0.6688 (↑ 0.66%)	0.8193 (↑ 1.49%)	0.7078 (↑ 1.08%)	0.6193 (↑ 0.60%)
	CAL	0.7501	0.6006	0.7802	0.6635	0.5559
	CAL+SGR	0.7737 (↑ 2.36%)	0.6414 (↑ 4.08%)	0.7936 (↑ 1.34%)	0.6849 (↑ 2.14%)	0.5976 (↑ 4.17%)
	DARE	0.7523	0.6618	0.8066	0.6823	0.6192
	DARE+SGR	0.7748 (↑ 2.25%)	0.6704 (↑ 0.86%)	0.8146 (↑ 0.80%)	0.7076 (↑ 2.53%)	0.6211 (↑ 0.19%)

4.8 Visualizations (RQ5)

In order to provide qualitative insights into the identified rationale subgraphs, a comprehensive analysis is conducted on Figs. 8 and 9. First, Fig. 8 showcases the rationales selected by different methods, including baselines and SGR, on the *Cycle-Tree* dataset. A testing example with a motif label of *House*[1)] is visualized in this figure. The red lines represent the edges of the rationale subgraph, while the navy blue nodes indicate the rationale nodes.

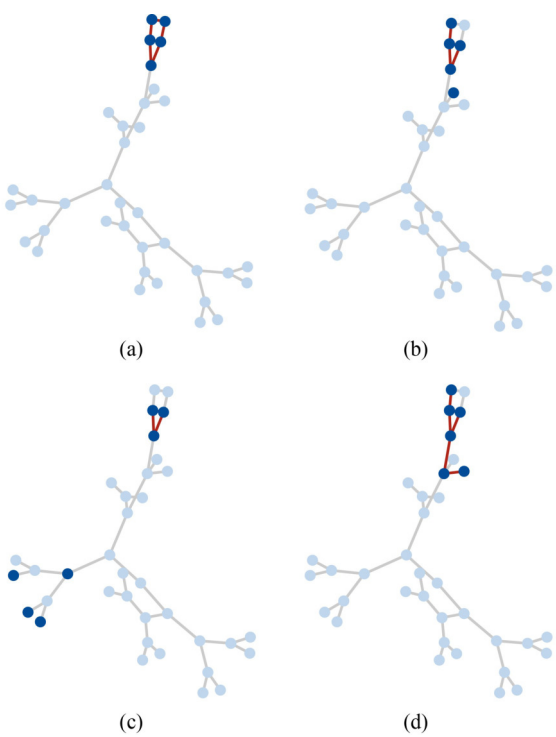


Fig. 8 Visualization of rationale subgraphs identified by different methods that are trained with the Spurious-Motif dataset *Cycle-Tree*. (a) SGR; (b) RGDA; (c) DIR; (d) GSAT

Among them, DIR and GSAT identify the edges as rationales. SGR and RGDA select the significant nodes as rationales. To make the visualization more intuitive, in SGR and RGDA, we assume that if there is an edge between the two identified nodes, we will visualize this edge as well. From this figure, we can observe that our method can identify more accurate rationales than baselines. More cases can be found in Figs. 10−12, where we present the extracted rationale subgraphs with different types of graphs.

In addition to the previous analyses, several cases of identified rationales for the Graph-SST2 dataset are presented in Fig. 9. This dataset encompasses both positive and negative text sentiments. In Figs. 9(a) and 9(b), the positive and negative examples from the training set are visualized, respectively. Among them, we find SGR can accurately highlight some positive tokens ("*the film was better*") in Fig. 9(a) and some negative tokens such as "*the opposite of... magical movie*" in Fig. 9(b). Furthermore, the effectiveness of the extracted rationales is showcased on the out-of-distribution (OOD) test set. This test set comprises nodes with degrees lower than those in the training set. Specifically, SGR selects "*quite effective*" and "*astonishingly witless*" in Figs. 9(c) and 9(d) to support the prediction results, respectively. These examples indicate that SGR can effectively extract the genuine rationale subgraphs, even in scenarios where the node degrees differ between the training and test sets. Based on these observations, it can be concluded that SGR exhibits effectiveness in extracting real rationale subgraphs, contributing to improved understanding and explainability of the model's predictions.

5 Discussion

5.1 Limitations

Although in Section 4.4 our experiments validated that graph

[1)] Recall Fig. 1 for more details about *House* and *Tree*.

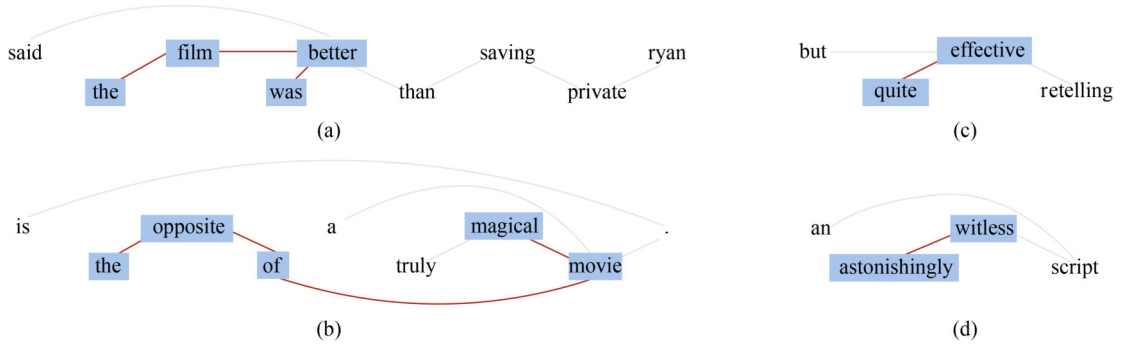


Fig. 9 Visualization of SGR rationale subgraphs, where the rationale tokens are highlighted by navy blue colors and the red lines indicate the edges between two identified rationale tokens. Among them, each graph represents a sentiment comment with positive/negative label (e.g., the positive comment "*said the film was better than saving private ryan*" in (a)). (a) Training rationale: Positive sentiment; (b) training rationale: negative sentiment; (c) testing rationale: Positive sentiment; (d) testing rationale: negative sentiment

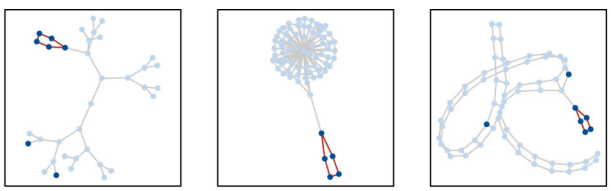


Fig. 10 Visualization of SGR rationale subgraphs, where the selected rationale nodes are highlighted by navy blue colors and the red lines indicate the edges between two identified rationale nodes. Among them, each graph consists of the motif type (*Cycle*) and bases (*Tree*, *Wheel* and *Ladder*). (a) *Cycle-Tree*; (b) *Cycle-Wheel*; (c) *Cycle-Ladder*

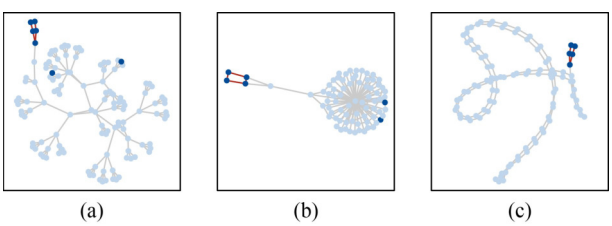


Fig. 11 Visualization of SGR rationale subgraphs, where each graph consists of the motif type (*House*) and bases (*Tree*, *Wheel* and *Ladder*). (a) *House-Tree*; (b) *House-Wheel*; (c) *House-Ladder*

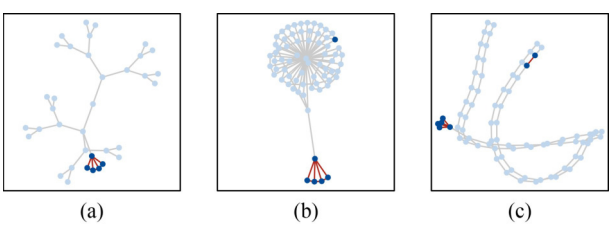


Fig. 12 Visualization of SGR rationale subgraphs, where each graph consists of the motif type (*Crane*) and bases (*Tree*, *Wheel* and *Ladder*). (a) *Crane-Tree*; (b) *Crane-Wheel*; (c) *Crane-Ladder*

learning methods are more likely to learn shortcut features in the early stages of training, this assumption does not hold if there are no significant shortcut learning problems in the dataset. In such cases, using SGR will not be effective and might even reduce the model's performance by treating real explanations as shortcuts. Therefore, before applying SGR, we should first detect whether shortcut learning problems are significant in the current scenario. One feasible detection approach is to test a vanilla rationalization method on an out-of-distribution (OOD) dataset. If the model's performance drops significantly compared to its performance on the in-distribution dataset, we can infer that shortcut learning problems are severe in the current scenario, and our SGR can be used to address them.

5.2 Computational overhead

SGR has two computational overhead bottlenecks. The first is the use of the early stopping strategy, which requires us to first train a shortcut guider. Although this strategy introduces some computational overhead, according to our experiments, the early stopping rounds are generally set to epochs less than 5. Therefore, this overhead is manageable. Besides, the mutual information (MI) estimation method requires the introduction of a parameterized discriminator function (e.g., $f(x,y)$ in Eq. (7)), which increases the computational overhead. This might be challenging to use in some resource-constrained scenarios. To address this problems, one feasible approach is to use other loss functions as alternatives to MI estimation, such as the mean squared error (MSE) function.

6 Conclusion

In this paper, we proposed a shortcut-guided graph rationalization (SGR) method for explainable graph learning, which identified rationale subgraphs by learning from shortcuts. To be specific, SGR involved two stages. In the first stage, a shortcut-only model (*shortcut guider*) was explicitly trained to capture the shortcut information in data with an early stop strategy. During the second stage, SGR separated the input graph into the rationale subgraph representations and the non-rationale ones. Then, the frozen *shortcut guider* was employed to transfer the shortcut information to the above subgraph representations, ensuring the rationale representations could be kept away from the shortcut and the non-rationale ones could encode the same information with shortcuts. Finally, we employed the non-rationale subgraphs as the environment and extracted invariant rationales capable of withstanding shifts in the environment. Experimental evaluations conducted on both synthetic and real-world datasets validated the effectiveness of SGR.

Acknowledgements This research was supported by grants from the Joint Research Project of the Science and Technology Innovation Community in Yangtze River Delta (No. 2023CSJZN0200), the National Natural Science

Foundation of China (Grant No. 62337001) and the Fundamental Research Funds for the Central Universities.

Competing interests The authors declare that they no competing interests or financial conflicts to disclose.

References

1. Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations. 2017
2. Wu Z, Gan Y, Xu T, Wang F. Graph-segmenter: graph transformer with boundary-aware attention for semantic segmentation. Frontiers of Computer Science, 2024, 18(5): 185327.
3. Liang Y, Song Q, Zhao Z, Zhou H, Gong M. BA-GNN: behavior-aware graph neural network for session-based recommendation. Frontiers of Computer Science, 2023, 17(6): 176613.
4. Wu Y, Huang H, Song Y, Jin H. Soft-GNN: towards robust graph neural networks via self-adaptive data utilization. Frontiers of Computer Science, 2025, 19(4): 194311.
5. Hu W, Fey M, Zitnik M, Dong Y, Ren H, Liu B, Catasta M, Leskovec J. Open graph benchmark: datasets for machine learning on graphs. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 1855
6. Guo Z, Zhang C, Yu W, Herr J, Wiest O, Jiang M, Chawla N V. Few-shot graph learning for molecular property prediction. In: Proceedings of the Web Conference 2021. 2021, 2559−2567
7. Yehudai G, Fetaya E, Meirom E A, Chechik G, Maron H. From local structures to size generalization in graph neural networks. In: Proceedings of the 38th International Conference on Machine Learning. 2021, 11975−11986
8. Ying R, Bourgeois D, You J, Zitnik M, Leskovec J. GNNExplainer: generating explanations for graph neural networks. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019, 829
9. Luo D, Cheng W, Xu D, Yu W, Zong B, Chen H, Zhang X. Parameterized explainer for graph neural network. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 1646
10. Lei T, Barzilay R, Jaakkola T. Rationalizing neural predictions. In: Proceedings of 2016 Conference on Empirical Methods in Natural Language Processing. 2016, 107−117
11. Wang X, Wu Y X, Zhang A, He X, Chua T S. Towards multi-grained explainability for graph neural networks. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. 2024, 1410
12. Chang S, Zhang Y, Yu M, Jaakkola T S. Invariant rationalization. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 1448−1458
13. Wu Y, Wang X, Zhang A, He X, Chua T S. Discovering invariant rationales for graph neural networks. In: Proceedings of the 10th International Conference on Learning Representations. 2022
14. Fan S, Wang X, Mo Y, Shi C, Tang J. Debiasing graph neural networks via learning disentangled causal substructure. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1808
15. Sui Y, Wang X, Wu J, Lin M, He X, Chua T S. Causal attention for interpretable and generalizable graph classification. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022, 1696−1705
16. Li H, Zhang Z, Wang X, Zhu W. Learning invariant graph representations for out-of-distribution generalization. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 859
17. Clark C, Yatskar M, Zettlemoyer L. Don't take the easy way out: ensemble based methods for avoiding known dataset biases. In: Proceedings of 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 2019, 4067−4080
18. Nam J, Cha H, Ahn S, Lee J, Shin J. Learning from failure: training debiased classifier from biased classifier. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 1736
19. Li Y, Lyu X, Koren N, Lyu L, Li B, Ma X. Anti-backdoor learning: training clean models on poisoned data. In: Proceedings of the 34th Advances in Neural Information Processing Systems. 2021, 14900−14912
20. Arpit D, Jastrzębski S, Ballas N, Krueger D, Bengio E, Kanwal M S, Maharaj T, Fischer A, Courville A, Bengio Y, Lacoste-Julien S. A closer look at memorization in deep networks. In: Proceedings of the 34th International Conference on Machine Learning. 2017, 233−242
21. Poole B, Ozair S, Van Den Oord A, Alemi A A, Tucker G. On variational bounds of mutual information. In: Proceedings of the 36th International Conference on Machine Learning. 2019, 5171−5180
22. Cheng P, Hao W, Dai S, Liu J, Gan Z, Carin L. CLUB: a contrastive log-ratio upper bound of mutual information. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 166
23. Yue L, Liu Q, Du Y, An Y, Wang L, Chen E. DARE: disentanglement-augmented rationale extraction. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1929
24. van den Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding. 2018, arXiv preprint arXiv: 1807.03748
25. Luo J, He M, Pan W, Ming Z. BGNN: behavior-aware graph neural network for heterogeneous session-based recommendation. Frontiers of Computer Science, 2023, 17(5): 175336.
26. Xiao S, Bai T, Cui X, Wu B, Meng X, Wang B. A graph-based contrastive learning framework for medicare insurance fraud detection. Frontiers of Computer Science, 2023, 17(2): 172341.
27. Schlichtkrull M S, De Cao N, Titov I. Interpreting graph neural networks for NLP with differentiable edge masking. In: Proceedings of the 9th International Conference on Learning Representations. 2021
28. Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: Proceedings of the 6th International Conference on Learning Representations. 2018
29. Chen Y, Zhang Y, Bian Y, Yang H, Ma K, Xie B, Liu T, Han B, Cheng J. Learning causally invariant representations for out-of-distribution generalization on graphs. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1608
30. Li H, Wang X, Zhang Z, Zhu W. Out-of-distribution generalization on graphs: a survey. 2022, arXiv preprint arXiv: 2202.07987
31. Yang N, Zeng K, Wu Q, Jia X, Yan J. Learning substructure invariance for out-of-distribution molecular representations. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 942
32. Wang F, Liu Q, Chen E, Huang Z, Yin Y, Wang S, Su Y. NeuralCD: a general framework for cognitive diagnosis. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(8): 8312−8327.
33. Liu G, Zhao T, Xu J, Luo T, Jiang M. Graph rationalization with environment-based augmentations. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022, 1069−1078
34. Tishby N, Pereira F C, Bialek W. The information bottleneck method. 2000, arXiv preprint arXiv: physics/0004057
35. Alemi A A, Fischer I, Dillon J V, Murphy K. Deep variational information bottleneck. In: Proceedings of the 5th International Conference on Learning Representations. 2017
36. Paranjape B, Joshi M, Thickstun J, Hajishirzi H, Zettlemoyer L. An

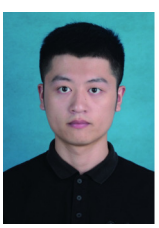
information bottleneck approach for controlling conciseness in rationale extraction. In: Proceedings of 2020 Conference on Empirical Methods in Natural Language Processing. 2020, 1938−1952

37. Wu T, Ren H, Li P, Leskovec J. Graph information bottleneck. In: Proceedings of the 34th Advances in Neural Information Processing Systems. 2020, 20437−20448
38. Yu J, Xu T, Rong Y, Bian Y, Huang J, He R. Graph information bottleneck for subgraph recognition. In: Proceedings of the 9th International Conference on Learning Representations. 2021
39. Miao S, Liu M, Li P. Interpretable and generalizable graph learning via stochastic attention mechanism. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 15524−15543
40. Geirhos R, Jacobsen J H, Michaelis C, Zemel R, Brendel W, Bethge M, Wichmann F A. Shortcut learning in deep neural networks. Nature Machine Intelligence, 2020, 2(11): 665−673.
41. Du M, He F, Zou N, Tao D, Hu X. Shortcut learning of large language models in natural language understanding: a survey. 2022, arXiv preprint arXiv: 2208.11857
42. Yue L, Liu Q, Wang L, An Y, Du Y, Huang Z. Interventional rationalization. In: Proceedings of 2023 Conference on Empirical Methods in Natural Language Processing. 2023, 11404−11418
43. Yue L, Liu Q, Du Y, Wang L, Gao W, An Y. Towards faithful explanations: Boosting rationalization with shortcuts discovery. In: Proceedings of the 12th International Conference on Learning Representations. 2024
44. Rashid A, Lioutas V, Rezagholizadeh M. MATE-KD: Masked adversarial TExt, a companion to knowledge distillation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. 2021, 1062−1071
45. Stacey J, Minervini P, Dubossarsky H, Riedel S, Rocktäschel T. Avoiding the hypothesis-only bias in natural language inference via ensemble adversarial training. In: Proceedings of 2020 Conference on Empirical Methods in Natural Language Processing. 2020, 8281−8291
46. Arjovsky M, Bottou L, Gulrajani I, Lopez-Paz D. Invariant risk minimization. 2019, arXiv preprint arXiv: 1907.02893
47. Sanh V, Wolf T, Belinkov Y, Rush A M. Learning from others' mistakes: Avoiding dataset biases without modeling them. In: Proceedings of the 9th International Conference on Learning Representations. 2021
48. Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? In: Proceedings of the 7th International Conference on Learning Representations. 2019
49. Liu G, Inae E, Luo T, Jiang M. Rationalizing graph neural networks with data augmentation. ACM Transactions on Knowledge Discovery from Data, 2024, 18(4): 86.
50. Socher R, Perelygin A, Wu J, Chuang J, Manning C D, Ng A, Potts C. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of 2013 Conference on Empirical Methods in Natural Language Processing. 2013, 1631−1642
51. Yu M, Chang S, Zhang Y, Jaakkola T. Rethinking cooperative rationalization: Introspective extraction and complement control. In: Proceedings of 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 2019, 4094−4103
52. Sun R, Tao H, Chen Y, Liu Q. HACAN: a hierarchical answer-aware and context-aware network for question generation. Frontiers of Computer Science, 2024, 18(5): 185321.
53. Kingma D P, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations. 2015

Linan Yue received the BE degree in computer science from Hehai University, China in 2019. He is currently pursuing the PhD degree in data science with University of Science and Technology of China under the advisory of Prof. Qi Liu. He has published several papers in referred journals and conference proceedings, such as IEEE TKDE, NeurIPS, ICLR, SIGIR, SIGKDD, and WWW conference. His current research interests include graph data mining, and trustworthy AI.

Qi Liu received the PhD degree from the University of Science and Technology of China (USTC), China in 2013. He is currently a professor with State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China (USTC), China. His research interests include data mining, machine learning, and trustworthy AI. He has published prolifically in refereed journals and conference proceedings (e.g., TKDE, TOIS, KDD). He is an Associate Editor of IEEE TBD and Neurocomputing, and the Young Associate Editor of FCS. He was the recipient of KDD' 18 Best Student Paper Award and ICDM' 11 Best Research Paper Award. He was also the recipient of China Outstanding Youth Science Foundation, in 2019.

Ye Liu is currently pursuing his PhD in the School of Data Science at the University of Science and Technology of China, China under the advisory of Prof. E. Chen, and is a member of State Key Laboratory of Cognitive Intelligence. His current research interests encompass graph learning and trustworthy AI. He has published several papers in referred journals and conference proceedings, such as ACM TKDD, IJCAI, and ACL conference.



Weibo Gao received his BE degree from the School of Software at Hefei University of Technology, China in 2019. He is currently pursuing a PhD in the School of Computer Science and Technology at the University of Science and Technology of China, China under the guidance of Prof. Qi Liu. He has contributed to numerous publications in reputable conference proceedings, including SIGIR, AAAI, and NeurIPS conference. His current research interests encompass data mining and trustworthy AI.

Fangzhou Yao is currently pursuing her PhD in the School of Data Science at the University of Science and Technology of China, China under the advisory of Prof. Qi Liu, and is a member of State Key Laboratory of Cognitive Intelligence. Her current research interests encompass machine learning and trustworthy AI. She has published several papers in referred conference proceedings, such as IJCAI and WWW conference.