Federated Self-Explaining GNNs with Anti-shortcut Augmentations

Linan Yue [1] Qi Liu* [1 2] Weibo Gao [1] Ye Liu [1] Kai Zhang [1] Yichao Du [1] Li Wang [3] Fangzhou Yao [1]

Abstract

Graph Neural Networks (GNNs) have demonstrated remarkable performance in graph classification tasks. However, ensuring the explainability of their predictions remains a challenge. To address this, graph rationalization methods have been introduced to generate concise subsets of the original graph, known as rationales, which serve to explain the predictions made by GNNs. Existing rationalizations often rely on shortcuts in data for prediction and rationale composition. In response, de-shortcut rationalization methods have been proposed, which commonly leverage counterfactual augmentation to enhance data diversity for mitigating the shortcut problem. Nevertheless, these methods have predominantly focused on centralized datasets and have not been extensively explored in the Federated Learning (FL) scenarios. To this end, in this paper, we propose a Federated Graph Rationalization (FedGR) with anti-shortcut augmentations to achieve self-explaining GNNs, which involves two data augmenters. These augmenters are employed to produce client-specific shortcut conflicted samples at each client, which contributes to mitigating the shortcut problem under the FL scenarios. Experiments on real-world benchmarks and synthetic datasets validate the effectiveness of FedGR under the FL scenarios. Code is available at `https://github.com/yuelinan/Codes-of-FedGR`.

1. Introduction

Graph Neural Networks (GNNs) have become ubiquitous in graph classification tasks, demonstrating remarkable performance (Hu et al., 2020; Zhang et al., 2021; Yehudai et al., 2021). Despite their success, GNNs applied to graph classification tasks still face challenges regarding the explainability of their prediction results. Consequently, researchers have actively explored methods to provide explanations for GNNs. Among these methods, graph rationalization (Wu et al., 2022; Fan et al., 2022; Sui et al., 2022; Li et al., 2022b) methods have garnered increasing attention. These methods aim to generate task results while identifying a concise subset of the original graph (i.e., the subgraph), referred to as the rationale. This rationale typically consists of significant nodes or edges that contribute to the prediction results. By extracting and presenting this rationale, it can serve as an explanation for the GNNs' prediction results, achieving self-explaining GNNs.

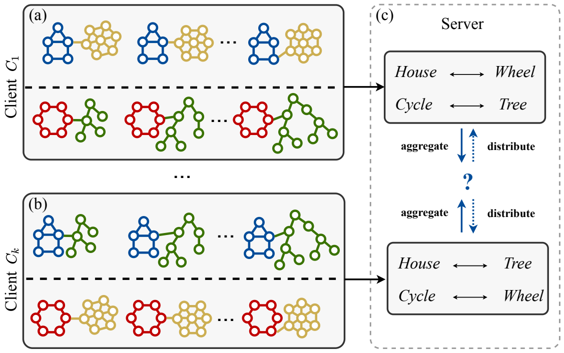


Figure 1. An example of the motif type prediction under the Federated Learning scenarios, where the *House* and *Cycle* are motif labels, and *Tree* and *Wheel* are base subgraphs. (a) and (b) illustrate the distribution of training data among distinct clients, each employing client-specific shortcuts to make predictions. For instance, Client C_1 considers the co-occurrence of *House* with *Wheel* as shortcuts, while Client C_k exhibits a greater co-occurrence of *House* with *Tree*. These divergent shortcuts utilized by different clients pose a challenge for the aggregated model to acquire an appropriate representation required for accurate classification.

While rationalization methods have achieved promising results, (Chang et al., 2020; Wu et al., 2022) highlight certain limitations associated with these approaches. They emphasize that rationalization methods often rely on shortcuts in the data to make predictions and construct rationales. These shortcuts exhibit a correlation with the task results, but lack any real causal relationship. Therefore, this correlation is commonly referred to as a spurious correlation

[1] State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei, China. [2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China. [3] ByteDance, Hangzhou, China. Correspondence to: Qi Liu <qiliuql@ustc.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

or bias. Taking Figure 1(a) for example, in the motif type prediction, we predict the motif type based on the graph that consists of motifs and bases subgraphs. In the training dataset, *House*-motifs frequently co-occur with *Wheel* bases, while *Cycle*-motifs are commonly associated with *Tree*. Graph rationalizations may exploit these statistical dependencies instead of the actual association between motif type and labels to make predictions. These statistical dependencies are classified as spurious correlations in this particular dataset.

It is important to note that the existence of this spurious correlation depends on the specific data distribution at hand. Therefore, rationalization methods can achieve promising task prediction performance in this distribution. However, the identification of shortcuts as rationales diminishes the reliability of the model's results. Moreover, when faced with out-of-distribution data, since the data distribution is changed, the performance of these methods significantly declines. For example, when faced with the *Cycle-Wheel* data, graph rationalizations trained on the Figure 1(a) dataset may incorrectly predict the motif type as *House*.

To this end, numerous approaches have been put forth to mitigate the issue of shortcut reliance in rationalization, referred to as de-shortcut rationalization methods. Noteworthy, several of these methods (e.g. DIR (Wu et al., 2022), RGDA (Liu et al., 2023), and DisC (Fan et al., 2022)) facilitate the data augmentation through counterfactual augmentations. By generating multiple samples that deviate from the existing distribution, these models alleviate employing shortcuts to make predictions within the current data distribution, thereby enhancing prediction capabilities. However, it is important to acknowledge that the aforementioned techniques primarily address the needs of centralized datasets. In the context of distributed training for learning models, particularly in Federated Learning (FL) (McMahan et al., 2017; Yang et al., 2019; Fu et al., 2022) scenarios, rationalization methods have not been extensively explored.

Specifically, in the FL scenario, the collection of data by different clients varies, resulting in distinct data distributions and the inclusion of different shortcuts within each client. Simply aggregating rationalization models learned from local clients to obtain a global model may lead to a significant effectiveness gap compared to a model trained on a centralized dataset. As shown in Figure 1, different clients tend to employ client-specific shortcuts for prediction. These shortcuts vary across clients, such as client C_1 utilizing the co-occurrence between *House* and *Wheel* to predict the motif type, while client C_k relies on the co-occurrence between *House* and *Tree*. This discrepancy in shortcut usage among clients hinders the aggregated model's ability to acquire accurate representations necessary for effective classification.

To address the problem of local shortcuts and enhance the generalization ability of rationalization in federated learning scenarios, we propose a ***Fed***erated ***G***raph ***R***ationalization (FedGR) with anti-shortcut augmentations method to generate shortcut conflicted samples for prediction, which involves two augmenters: the complement-aware augmenter and difference-aware augmenter.

• For the complement-aware augmenter, we first partition the graph into the rationale and the complement subgraph (aka, the non-rationale subgraph). The rationale subgraph is utilized to predict task results, while the complement serves as a means for data augmentation. Specifically, we use a contrastive learning approach (Oord et al., 2018) to satisfy the sufficiency and independence principles (DeYoung et al., 2020; Li et al., 2022c) of the rationalization. This approach can promote the model to compose the invariant rationales and make the complement be irrelevant to labels. Finally, we introduce random permutations to the label-independent complements and merge them with the rationale to generate more shortcut conflicted samples. This augmentation can break up the original data distributions (e.g., the statistical dependencies between rationales and complements).

• Besides, to further explore anti-shortcut augmentations in FL scenarios, we propose a difference-aware augmenter. Initially, we develop a node feature masking technique that perturbs the features of nodes while preserving the underlying graph structure, thereby generating new graph data. Then, we enforce the generated data to go cross the decision boundary of the local model while preserving the predictions of the global model. This is based on the difference between the local and global models in the FL scenario, where local models are more prone to utilizing shortcuts compared to global models when making predictions (Xu et al., 2023). Finally, the generated graph data can be considered as the shortcut conflicted samples that do not conform to the current client data distribution.

After obtaining shortcut conflicted samples through the two data augmenters, FedGR mixes the original and generated data to jointly make the task prediction and compose rationales. Experiments over real-world benchmarks (Hu et al., 2020; Knyazev et al., 2019) and various synthetic datasets (Wu et al., 2022) validate the effectiveness of FedGR.

2. Preliminaries

2.1. Problem Formulation

In this section, we present a formal definition of the graph rationalization problem within the federated learning (FL) scenario. Specifically, we consider a federated setting with N clients denoted as $\{C_1, C_2, \cdots, C_N\}$, each having their respective local private datasets $\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N\}$. It is important to note that the data distributions of different clients are not equal, indicating variability among clients.

Within each client C_k, for each graph-label pair $(G, Y) \in \mathcal{D}_k$, where $G = (\mathcal{V}, \mathcal{T})$ contains $\mathcal{V}$ nodes and $\mathcal{T}$ edges, the objective of local graph rationalization is two-folds. First, it involves learning a mask variable $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}|}$ with the separator function $f_{s_k}(G)$ and node representations $\mathbf{H}_G \in \mathbb{R}^{|\mathcal{V}| \times d}$. Subsequently, the rationale subgraph representation is obtained as the element-wise multiplication between the mask variable and the node representations, denoted as $\mathbf{M} \odot \mathbf{H}_G$. Finally, a predictor $f_{p_k}(\mathbf{M} \odot \mathbf{H}_G)$ is learned to make predictions. The learning process involves finding the optimal selector function $f^*_{s_k}(\cdot)$ and predictor function $f^*_{p_k}(\cdot)$ that minimize the cross-entropy loss, denoted by $\ell(\cdot)$, over the graph-label pairs in the client's dataset $\mathcal{D}_k$:

$$f^*_{s_k}(\cdot), f^*_{p_k}(\cdot) = \\ \arg\min_{f_{s_k}, f_{p_k}} \mathbb{E}_{(G,Y)\sim\mathcal{D}_k}\left[\ell\left(f_{p_k}\left(f_{s_k}(G)\right), Y\right)\right].$$

With a total of T communication rounds, the objective of graph rationalization at the global level is to derive the selector and predictor that satisfy the aggregation process:

$$\hat{\Theta}^s = \sum_{k=1}^{N} \frac{|D_k|}{\sum_{j=1}^{N} |D_j|} \Theta^s_k, \;\; \hat{\Theta}^p = \sum_{k=1}^{N} \frac{|D_k|}{\sum_{j=1}^{N} |D_j|} \Theta^p_k, \quad (1)$$

where $\hat{\Theta}^s$ is the parameters of the global selector $f_s(\cdot)$, and $\hat{\Theta}^p$ is the parameters of the global predictor $f_p(\cdot)$. Meanwhile, Θ^s_k represents the parameters of the selector $f_{s_k}(\cdot)$ in client C_k, and Θ^p_k is the parameters of the predictor $f_{p_k}(\cdot)$.

2.2. Vanilla Graph Rationalization

In this section, we present the detail of the framework of vanilla graph rationalization in the general scenario, which consists of the selector and the predictor.

Selector in Graph Rationalization. Considering the general scenario, given $(G, Y) \in \mathcal{D}$, where $\mathcal{D}$ is the dataset, the process of generating rationales within the separator $f_s(\cdot)$ involves three key steps. Initially, an encoder $\mathrm{GNN}_m(\cdot)$ is utilized to transform each node in graph G into a d-dimensional vector. Simultaneously, the separator predicts a probability distribution for selecting each node as part of the rationale, denoted as:

$$\tilde{\mathbf{M}} = \mathrm{softmax}\left(W_m\left(\mathrm{GNN}_m(G)\right)\right),$$

where $W_m \in \mathbb{R}^{2\times d}$ represents a weight matrix.

Next, the separator samples binary values (0 or 1) from the distribution $\tilde{\mathbf{M}} = \{\tilde{m}_i\}_1^{|\mathcal{V}|}$ to yield the mask variable $\mathbf{M} = \{m_i\}_1^{|\mathcal{V}|}$. To enable differentiability during the sampling, the Gumbel-softmax method (Jang et al., 2017) is used:

$$m_i = \frac{\exp\left(\left(\log\left(\tilde{m}_i\right) + q_i\right)/\tau\right)}{\sum_t \exp\left(\left(\log\left(\tilde{m}_t\right) + q_t\right)/\tau\right)},$$

where τ is a temperature hyperparameter, $q_i = -\log\left(-\log\left(u_i\right)\right)$, and u_i is randomly sampled from a uniform distribution $U(0,1)$. Following this, an additional GNN encoder, denoted as GNN_G, is employed to obtain the node representation $\mathbf{H}_G$ from the graph G. The rationale node representation is defined as the element-wise product of the binary rationale mask $\mathbf{M}$ and the node representation $\mathbf{H}_G$, expressed as $\mathbf{M} \odot \mathbf{H}_G$. Similarly, the complement node representation is computed as $(1 - \mathbf{M}) \odot \mathbf{H}_G$, representing the nodes that are part of the non-rationale.

Predictor in Graph Rationalization. The predictor $f_p(\cdot)$ consists of a readout function and a classifier. Specifically, we first use the readout function to yield the graph-level rationale $\mathbf{h}_r$ and complement $\mathbf{h}_e$ (i.e., the non-rationale) subgraph representation:

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

In this paper, we employ the mean pooling as the readout operator. Finally, the classifier $\Phi(\cdot)$ yields the task results based solely on the rationale subgraphs:

$$\hat{Y}_r = \Phi\left(\mathbf{h}_r\right), \quad \mathcal{L}_r = \mathbb{E}_{(G,Y)\sim\mathcal{D}}\left[\ell(\hat{Y}_r, Y)\right]. \quad (2)$$

Training and Inference. During the training, we introduce a sparsity constraint on the probability $\mathbf{M}$ of being selected as a rationale, as proposed in (Liu et al., 2022). This constraint aims to encourage the model to achieve a controlled level of sparsity in the generated rationale subgraphs.:

$$\mathcal{L}_{sp} = \left|\frac{1}{|\mathbf{M}|}\sum_{i=1}^{|\mathbf{M}|} m_i - \alpha\right|, \quad (3)$$

where $\alpha \in [0, 1]$ is a predefined sparsity level. Finally, the overall objective of the vanilla graph rationalization is:

$$\mathcal{L}_{rat} = \mathcal{L}_r + \lambda_{sp}\mathcal{L}_{sp}.$$

In the inference phase, we use $\mathbf{h}_r$ for the prediction.

3. FedGR:Federated Graph Rationalization with Anti-shortcut Augmentations

Similar to standard FL, training of FedGR requires alternative optimization between the two stages. More specifically, the local update is performed on the client side, and global aggregation is conducted on the server side. Besides, to address local shortcut problems, we uniquely propose an anti-shortcut augmentation method that aims to generate several the shortcut conflicted samples for de-shortcut rationalization. As shown in Figure 2, the anti-shortcut data augmentation method consists of two augmenters:

• First, based on the sufficiency and independence principles (DeYoung et al., 2020; Li et al., 2022c) of the rationalization method, for each client, we design a complement-aware augmenter to enhance the diversity of local data distributions.

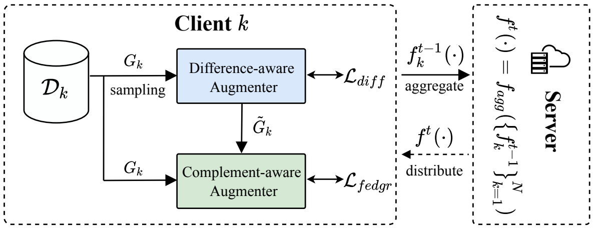


Figure 2. Architecture of FedGR, including the complement-aware and the difference-aware augmenter.

• To further explore anti-shortcut data augmentation in federated scenarios, we propose an augmenter based on the difference between the global and local models. The diversity of data distribution is enhanced by injecting global information into local training.

3.1. Complement-aware Augmenter

In this section, we first introduce the sufficiency and independence principle for rationalization. Then, we propose a contrastive constraint to satisfy the above principles. Finally, based on this, we derive a complement-aware augmenter that composes counterfactual samples under different complements to isolate shortcuts.

3.1.1. Sufficiency and Independence Principle

Definition 3.1. Sufficiency Principle for Rationalization:

$$P(Y \mid G) = P(Y \mid R),$$

where the rationale R is sufficient to preserve the crucial information inherent in G related to the label Y.

Definition 3.2. Independence Principle for Rationalization:

$$Y \perp\!\!\!\perp E \mid R,$$

where the label variable Y exhibits independence from the complement variable of rationale, denoted as E, conditioned on the rationale R. Among $\perp\!\!\!\perp$ is probabilistic independence.

Guided by the sufficiency and independence principle, we get the following objective to compose invariant rationales:

$$P(Y \mid G) = P(Y \mid R) \quad \text{s.t. } Y \perp\!\!\!\perp E \mid R. \tag{4}$$

To achieve Eq(4), we first employ the contrastive constraint:

$$\mathcal{L}_c = -\log \frac{\exp\left(\mathbf{h}_r^\top \mathbf{h}_g / \tau\right)}{\exp\left(\mathbf{h}_r^\top \mathbf{h}_g / \tau\right) + \sum_{\mathbf{h}_e \in \mathcal{E}} \exp\left(\mathbf{h}_r^\top \mathbf{h}_e / \tau\right)}, \tag{5}$$

where $\mathbf{h}_g$ is the graph-level representation of G (i.e., $\mathbf{h}_g = \text{READOUT}(\mathbf{H}_G)$), the set $\mathcal{E}$ encompasses all complement representations present in the mini-batch data, and τ serves as a temperature parameter.

By minimizing Eq(5), we can effectively push the representation of the complement $\mathbf{h}_e$ apart from that of the rationale $\mathbf{h}_r$, ensuring the stability of the captured rationale irrespective of variations in its complement. This aligns with the independence principle. Simultaneously, the rationale and the original input G are drawn closer together, thus achieving the sufficiency principle. Finally, by combining Eq(2) with Eq(5) ($\mathcal{L}_r + \mathcal{L}_c$), we can realize the objective stated in Eq(4) (i.e., encompassing both the sufficiency and independence principles).

3.1.2. Implemention

After satisfying the sufficiency and independence principles, we can pull R and G closer together while pushing R and E, E and Y apart. Then, we derive the following equation:

$$P(Y|G) = P(Y|R) = P(Y|R,E) = P(Y|R,\hat{E}), \tag{6}$$

where $\hat{E}$ is sampled from the complement set randomly.

Based on Eq(6), we can achieve the complement-aware augmenter. To be specific, in the client C_k, given a batch $\left\{\left({G_k}^i, {Y_k}^i\right)\right\}_{i=1}^B$ and the corresponding rationale and complement representation $\left\{\left(\mathbf{h}_{r_k}^i, \mathbf{h}_{e_k}^i\right)\right\}_{i=1}^B$, we first randomly sample a complement representation $\mathbf{h}_{e_k}^j$ from the complement set $\left\{\mathbf{h}_{e_k}^i\right\}_{i=1}^B$, where $\mathbf{h}_{e_k}^i \neq \mathbf{h}_{e_k}^j$. Then, with the concatenation of $\mathbf{h}_{r_k}^i$ and $\mathbf{h}_{e_k}^j$, the complement-aware augmentation sample $\mathbf{h}_{\tilde{g}_i}^{(i,j)}$ can be expressed as:

$$\mathbf{h}_{\tilde{g}_k}^{(i,j)} = \mathbf{h}_{r_k}^i + \mathbf{h}_{e_k}^j, \tag{7}$$

where we have $\tilde{Y}_k^{(i,j)} = {Y_k}^i$ for the augmented sample (i.e., the class label of $\mathbf{h}_{\tilde{g}_k}^{(i,j)}$ is identical to that of $\mathbf{h}_{g_k}^j$).

It is note that although this data augmentation is common in rationalizations (Fan et al., 2022; Liu et al., 2022; Sui et al., 2022; Liu et al., 2023), the fact that we use contrastive learning constraints to satisfy the independence and sufficiency principles makes it more efficient. We have demonstrated this opinion in section 4.3.

Finally, the augmented sample is input into the classifier $\Phi(\cdot)$ to yield task results, achieving $P(Y|R,\hat{E})$ in Eq(6):

$$\hat{Y}_k^{(i,j)} = \Phi\left(\mathbf{h}_{\tilde{g}_k}^{(i,j)}\right), \; \mathcal{L}_e = \mathbb{E}_{\left({G_k}^i, {Y_k}^i\right) \sim \mathcal{D}_k}\left[\ell(\hat{Y}_k^{(i,j)}, \tilde{Y}_k^{(i,j)})\right].$$

3.1.3. Training and Inference

During the training in the FL scenario, for each client C_k, the overall objective of graph rationalization with the complement-aware augmenter is formulated as:

$$\mathcal{L}_{com}^k = \mathcal{L}_{rat} + \lambda_c \mathcal{L}_c + \lambda_e \mathcal{L}_e,$$

where λ_c and λ_e are the adjusted hyperparameters.

In the inference phase, only $\mathbf{h}_{r_k}$ is employed to yield the task results without any augmentations.

Notably, since our approach can be applied to any local model, it can be naturally applied to centralized scenarios.

3.2. Difference-aware Augmenter

In this section, to fully exploit the attributes of FL, we propose a difference-aware augmenter to produce the anti-shortcut samples by utilizing the difference between global server model $f^t(\cdot)$ and local client model $f_k^{t-1}(\cdot)$ of the client C_k in the (t)-th communication round.

3.2.1. Learning of Difference-aware Augmenter

Assumption 3.3. In the FL scenario, given the global server model $f^t(\cdot)$ and the local model $f_k^{t-1}(\cdot)$ generated in the previous iteration, we assume that $f^t(\cdot)$ exhibits a relatively unbiased nature in comparison to $f_k^{t-1}(\cdot)$ (Xu et al., 2023).

Based on Assumption 3.3, the goal of the difference-aware augmentation is to generate an anti-shortcut sample $\tilde{G}_k$ for the client C_k to satisfy the following conditions:

Condition 3.4. Given the bias model $f_k^{t-1}(\cdot)$, the generated sample $\tilde{G}_k$ and the label Y_k are independent:

$$Y_k \perp\!\!\!\perp \tilde{G}_k \mid f_k^{t-1}(\cdot).$$

Condition 3.5. Given the unbias model $f^t(\cdot)$, the generated sample $\tilde{G}_k$ and the label Y_k are dependent:

$$Y_k \not\perp\!\!\!\perp \tilde{G}_k \mid f^t(\cdot).$$

Based on the above conditions, we employ the mutual information (MI) to train the difference-aware augmenter $\Psi(\cdot)$:

$$\max I(Y_k; \tilde{G}_k | f^t(\cdot)) \quad \text{s.t. } I(Y_k; \tilde{G}_k | f_k^{t-1}(\cdot)) \leq I_c, \quad (8)$$

where $I(X;Y)$ represents the MI of X and Y variables, and I_c is the information constraint. Since the MI is hard to calculate, we give an equivalent tractable objective in practical instantiation to achieve Eq(8):

$$\min_{\Psi} \mathcal{L}_{diff} =$$
$$\min_{\Psi} [\underbrace{\ell(f^t(\Psi(G_k)), Y_k)}_{①} - \underbrace{\beta\ell(f_k^{t-1}(\Psi(G_k)), Y_k)}_{②}]. \quad (9)$$

Theorem 3.6. *To train the difference-aware augmenter, minimizing term ① in Eq(9) contributes to* $\max I(Y_k; \tilde{G}_k | f^t(\cdot))$*; maximizing term ② in Eq(9) contributes to* $\min I(Y_k; \tilde{G}_k | f_k^{t-1}(\cdot))$.

The proof of Theorem 3.6 is provided in Appendix A.

Remark 3.7. Eq(9) enables the transformation of the original graph G_k in such a way that it goes across the decision boundary of $f_k^{t-1}(\cdot)$ while simultaneously maintaining the prediction made by $f^t(\cdot)$. This transformation can form the anti-shortcut sample $\tilde{G}_k$ for the client C_k.

3.2.2. Implemention

In this section, we present how to generate anti-shortcut samples in detail. Specifically, in a general scenario, given an input graph G, we employ the difference-aware augmenter $\Psi(\cdot)$ to generate a new graph $\tilde{G}$. Considering the complexity of the graph structure, we do not perturb the edges of the graph (i.e., adding or removing edges) to generate samples. For simplicity, we employ a masking transformation on the node features of the graph to generate the new graph.

Specifically, we assume that the node features of the input graph G are represented by $\mathbf{S} = [s_1, \cdots, s_{|\mathcal{V}|}]^T \in \mathbb{R}^{|\mathcal{V}| \times d_g}$, where each $s_i \in \mathbb{R}^{d_g}$ corresponds to the d_g-dimensional feature vector of node i. To compute the mask probability matrix $\tilde{\mathbf{M}}^{\mathbf{s}} \in \mathbb{R}^{|\mathcal{V}| \times d_g}$, we employ the MLP model that takes $\mathbf{S}$ as input and applies the sigmoid function $\sigma(\cdot)$ to the output. Each entry $\tilde{M}_{ij}^s$ in the resulting matrix $\tilde{\mathbf{M}}^{\mathbf{s}}$ represents the predicted probability of not setting the j-th feature of node i to zero. Then, we utilize the Gumbel-softmax method to sample the mask matrix $\mathbf{M}^{\mathbf{s}}$, where each value is either 0 or 1. This process can be defined as follows:

$$\tilde{\mathbf{M}}^{\mathbf{s}} = \sigma(\text{MLP}(\mathbf{S})), \mathbf{M}^{\mathbf{s}} \sim \text{Gumbel-softmax}(\tilde{\mathbf{M}}^{\mathbf{s}}).$$

Then, the new node feature matrix $\tilde{\mathbf{S}}$ is computed as $\tilde{\mathbf{S}} = \mathbf{M}^{\mathbf{s}} \odot \mathbf{S}$. Finally, the generated sample $\tilde{G}$ retains the same graph structure and label as the original sample G, but utilizes the new node feature matrix $\tilde{\mathbf{S}}$. In this paper, for the client C_k, we denote this process as $\tilde{G}_k = \Psi(G_k)$ and the generated dataset as $\tilde{D}_k$.

3.2.3. Training and Inference

In the client C_k, we first employ Eq(9) to train the difference-aware augmenter $\Psi(G_k)$. After the $\Psi(G_k)$ is trained with several training epochs, we freeze the augmenter $\Psi(G_k)$ and utilize it to infer the potential anti-shortcut sample $\tilde{G}_k$. Next, as shown in Figure 2, the generated sample can be incorporated into the complement-aware augmenter, and the corresponding objective is presented as $\tilde{\mathcal{L}}_{com}^k$. Finally, the overall objective of FedGR with two data augmenters is:

$$\mathcal{L}_{fedgr} = \mathcal{L}_{com}^k + \lambda_d \tilde{\mathcal{L}}_{com}^k. \quad (10)$$

In the inference phase, only $\mathbf{h}_{r_k}$ is adopted to yield the task results without any augmentations. The overall training algorithm of FedGR is presented in Algorithm 1.

4. Experiments

To validate the effectiveness of FedGR, we design experiments to address the following research questions:

Table 1. Performance on the Synthetic Dataset and Real-world Dataset with the GIN backbone. More experimental results about FedGR implemented with the GCN backbone are shown in Appendix E.1.

	Spurious-Motif (ACC)			OGB (AUC)			
	bias=0.5	bias=0.7	bias=0.9	MolHIV	MolToxCast	MolBBBP	MolSIDER
GIN	0.3213 ± 0.0429	0.3489 ± 0.0442	0.2978 ± 0.0382	0.6927 ± 0.0308	0.6091 ± 0.0133	0.6226 ± 0.0133	0.5780 ± 0.0105
Vanilla GR	0.3182 ± 0.0353	0.3681 ± 0.0359	0.3031 ± 0.0291	0.6985 ± 0.0155	0.6111 ± 0.0055	0.6339 ± 0.0142	0.5774 ± 0.0175
DIR	0.3091 ± 0.0314	0.3298 ± 0.0148	0.2893 ± 0.0311	0.6731 ± 0.0337	0.6133 ± 0.0064	0.6245 ± 0.0098	0.5686 ± 0.0162
DisC	0.4418 ± 0.0182	0.4481 ± 0.0381	0.3579 ± 0.0471	0.7212 ± 0.0201	0.6274 ± 0.0018	0.6561 ± 0.0121	0.5869 ± 0.0142
CAL	0.4213 ± 0.0109	0.5289 ± 0.0087	0.4191 ± 0.0248	0.7039 ± 0.0113	0.6170 ± 0.0051	0.6575 ± 0.0076	0.5879 ± 0.0138
GSAT	0.4281 ± 0.0328	0.5259 ± 0.0381	0.4194 ± 0.0338	0.7149 ± 0.0226	0.6255 ± 0.0030	0.6555 ± 0.0085	0.5952 ± 0.0082
DARE	0.4483 ± 0.0193	0.4891 ± 0.0391	0.4288 ± 0.0977	0.7220 ± 0.0165	0.6289 ± 0.0059	0.6621 ± 0.0096	0.5886 ± 0.0113
InterRAT	0.4191 ± 0.0943	0.5283 ± 0.0935	0.4281 ± 0.0189	0.7026 ± 0.0092	0.6095 ± 0.0028	0.6426 ± 0.0223	0.5842 ± 0.0078
RGDA	0.4087 ± 0.0293	0.5089 ± 0.0198	0.4286 ± 0.0313	0.7246 ± 0.0085	0.6235 ± 0.0034	0.6605 ± 0.0157	0.5906 ± 0.0151
FedGR	**0.4610 ± 0.0289**	**0.5538 ± 0.0398**	**0.4977 ± 0.0315**	**0.7387 ± 0.0186**	**0.6316 ± 0.0054**	**0.6690 ± 0.0174**	**0.6017 ± 0.0202**
FedGR w/o diff	0.4493 ± 0.0238	0.5293 ± 0.0483	0.4333 ± 0.0471	0.7214 ± 0.0124	0.6222 ± 0.0055	0.6623 ± 0.0033	0.5886 ± 0.0047
FedGR w/o com	0.4571 ± 0.0372	0.5438 ± 0.0551	0.4682 ± 0.0388	0.7321 ± 0.0233	0.6298 ± 0.0035	0.6668 ± 0.0048	0.5978 ± 0.0021

- **RQ1:** How effective is FedGR in improving task prediction and rationale extraction?

- **RQ2:** How well does the complement-aware augmenter mitigate the shortcut problem?

- **RQ3:** Can the framework of FedGR with the difference-aware augmenter contribute to the performance improvement in existing de-shortcut rationalization methods?

- **RQ4:** What is the performance trajectory of FedGR during the training process?

- **RQ5:** How FedGR scales with an increasing number of clients?

4.1. Datasets

• **Synthetic Dataset.** This study utilizes the Spurious-Motif dataset (Ying et al., 2019; Wu et al., 2022) as the synthetic dataset for the motif type prediction. Each graph consists of two subgraphs: the motif subgraph R and the base subgraph E. The motif subgraph represents the rationale for motif type prediction and includes three types: *Cycle*, *House*, and *Crane*, denoted as $R = \{0, 1, 2\}$. Conversely, the base subgraph varies according to the motif type and serves as the complement, comprising three types: *Tree*, *Ladder*, and *Wheel*, denoted as $E = \{0, 1, 2\}$. Figure 1 illustrates an example of the Spurious-Motif dataset, such as *House-Tree*.

To demonstrate that FedGR can mitigate the shortcut problem, we manually introduce the shortcuts into the Spurious-Motif dataset. During the construction process, we sample the motif subgraph uniformly and select the base subgraph based on $P(E) = b \times \mathbb{I}(E = R) + \frac{1-b}{2} \times \mathbb{I}(E \neq R)$, where b controls the extent of data distributions, with higher values indicating more significant shortcuts in the data. In this study, three datasets are considered with $b = \{0.5, 0.7, 0.9\}$. Next, we distribute the constructed dataset to N clients by the unbalanced partition algorithm Latent Dirichlet Allocation (LDA) (He et al., 2020; 2021). Specifically, a heterogeneous partition is generated by sampling $p_i \sim \text{Dir}_N(\gamma)$, allocating a proportion $p_{i,n}$ of training instances for class i to each local client. In this paper, N is set to 3 and γ to 3. Finally, to ensure fair evaluation, a de-biased (balanced) dataset is created for the test set by setting $b = \frac{1}{3}$.

• **OGB.** For real-world datasets, we utilize the Open Graph Benchmark (OGB) (Hu et al., 2020) as datasets, including MolHIV, MolToxCast, MolBBBP, and MolSIDER. To ensure a fair evaluation, we first adopt the default scaffold splitting method in OGB to partition the datasets into training, validation, and test sets. Then, we employ the LDA the further distribute the training set to 4 clients with $\gamma = 4$, where all clients share the same test set.

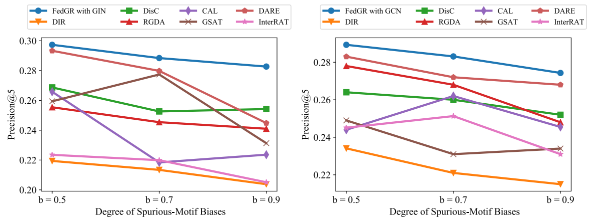
Details of dataset statistics are shown in Appendix D.

4.2. Comparison Methods and Experimental Setup

Comparison Methods. Although graph rationalization methods are widely studied, they are not fully explored in FL scenarios. To this end, we transfer the following rationalization methods that are used in centralized scenarios to FL scenarios by the aggregation approach of Eq(1), including Vanilla GR in section 2.2, DIR (Wu et al., 2022), DisC (Fan et al., 2022), CAL (Sui et al., 2022), GSAT (Miao et al., 2022), DARE (Yue et al., 2022), InterRAT (Yue et al., 2023) and RGDA (Liu et al., 2023). Detailed descriptions of comparison methods are shown in Appendix C.1.

Besides, in the comparative analysis, several conventional GNN architectures are considered for classification tasks, including GCN (Kipf & Welling, 2017) and GIN (Xu et al., 2019). Meanwhile, we employ both GCN and GIN as the backbone of FedGR and other baselines.

Experimental Setup. During the evaluation phase, we employ the ACC metric to evaluate the task prediction perfor-



Precision@5 on Spurious-Motif with GIN as the graph encoder.

Precision@5 on Spurious-Motif with GCN as the graph encoder.

Figure 3. Results of Precision@5 between extracted rationales and the ground-truth rationales on Spurious-Motif.

mance for the Spurious-Motif and AUC for OGB. Then, since the Spurious-Motif dataset includes ground-truth rationales, the precision of the extracted rationales is evaluated using the Precision@5 metric. Precision@5 measures the accuracy of the top 5 extracted rationales compared to the ground truth rationales. All methods, including the FedGR approach and other baselines, are trained on a single A100 GPU with 5 different random seeds. The reported test performance includes the mean results and standard deviations obtained from the epoch that achieves the highest validation prediction performance. Detailed experimental and hyperparameter setups can be found in Appendix C.2.

4.3. Overall Performance (RQ1).

Performance of the Task Prediction. To evaluate the effectiveness of FedGR, a comparative analysis is conducted against various baseline methods in the task prediction. Specifically, examining Table 1 reveals that FedGR exhibits optimal performance, thereby highlighting the effectiveness of the two proposed data augmenters. Besides, it is observed that certain de-shortcut methods (e.g. DIR, CAL, and InterRAT) demonstrate similar performance to the traditional GIN, suggesting that de-shortcut models designed for centralized scenarios may not be well-suited for direct implementation in FL scenarios. This underscores the importance of exploring the de-shortcut rationalization approaches within FL settings. Furthermore, several data augmentation-based methods (e.g. RGDA and DisC) exhibit sub-optimal performance, illustrating the benefits of employing counterfactual data augmentation approaches in mitigating the shortcut problem within FL scenarios. Consequently, an in-depth investigation (ablation study) is conducted to assess the impact of the proposed complement-aware and difference-aware augmenters on the effectiveness of FedGR.

Ablation Study. We first exclude the difference-aware augmenter and retain only the complement-aware augmenter, denoting it as FedGR w/o diff. As shown in Table 1, we find that FedGR w/o diff surpasses the baseline methods. Among them, DisC, CAL and RGDA employ the similar counterfactual data augmentation method as the complement-aware augmenter (i.e., Eq(7)). However, our method performs better than them, and the reason is that compared to these methods, the contrastive learning constraint we used satisfies the sufficiency and independence principles of rationalizations.

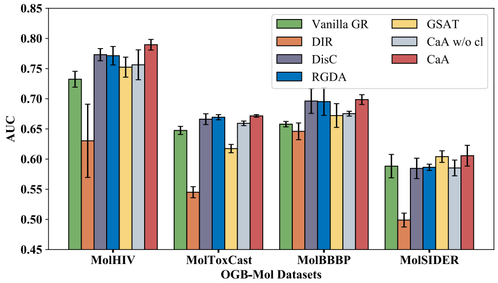


Figure 4. Performance on the Real-world Dataset.

Besides, we also remove the complement-aware augmenter while keeping the difference-aware augmenter. This variant is referred to as FedGR w/o com. Table 1 reveals that FedGR w/o com performs better than FedGR w/o diff. This result can be attributed to the fact that the difference-aware augmenter, compared to the complement-aware augmenter, more fully exploits the properties of FL, which utilizes the global model to assist the local model for removing shortcuts. This finding once again highlights the necessity of exploring de-shortcut rationalization within FL scenarios.

Performance of the Rationale Extraction. Furthermore, to delve deeper into the ability of FedGR to capture the true rationales rather than relying on shortcuts, we conduct experiments on the Spurious-Motif dataset, which contains the ground rationales. In Figure 3, we present Precision@5 that measures the precision of the top 5 extracted rationales compared to the ground truth. From the figure, we find that regardless of the degree of bias in Spurious-Motif, the rationales extracted by FedGR are more accurate compared to the baseline method, thereby demonstrating the capability of FedGR to overcome shortcuts and provide more reliable rationales. Besides, in Appendix E.6, we provide several visualized rationales extracted by FedGR in Spurious-Motif.

4.4. Performance of Complement-aware augmenter in centralized scenarios (RQ2).

In section 4.3, we validate the effectiveness of the complement-aware augmenter (referred to as CaA) through ablation experiments. In this section, we further investigate the performance of CaA. Specifically, in section 3, we state that CaA can be naturally applied to centralized scenarios. Therefore, we make experiments on the centralized scenarios with the GIN backbone, and the results are presented in Figure 4. From the observation, we find that CaA outperforms baselines, illustrating that our complement-aware augmenter is effective in both the centralized and FL scenarios. Then, we also remove the contrastive constraint (i.e., $\mathcal{L}_c$) and denote it as CaA w/o cl. From the figure, we can observe that CaA w/o cl has a significant decrease compared to

Table 2. Structural Generalizability of FedGR with the GIN backbone. Each rationalization method in FedGR is highlighted in gray.

	MolHIV	MolToxCast	MolBBBP	MolSIDER
DisC	0.7212	0.6274	0.6561	0.5869
DisC+FedGR	0.7313 (↑1.01%)	0.6301 (↑0.27%)	0.6618 (↑0.57%)	0.5942 (↑0.73%)
RGDA	0.7246	0.6235	0.6605	0.5906
RGDA+FedGR	0.7344 (↑0.98%)	0.6326 (↑0.91%)	0.6673 (↑0.68%)	0.6008 (↑1.02%)
GSAT	0.7149	0.6255	0.6555	0.5952
GSAT+FedGR	0.7267 (↑1.18%)	0.6293 (↑0.38%)	0.6628 (↑0.73%)	0.5980 (↑0.28%)
InterRAT	0.7026	0.6095	0.6426	0.5842
InterRAT+FedGR	0.7193 (↑1.67%)	0.6245 (↑1.50%)	0.6587 (↑1.61%)	0.5927 (↑0.85%)
DARE	0.7220	0.6289	0.6621	0.5886
DARE+FedGR	0.7291 (↑0.71%)	0.6331 (↑0.42%)	0.6686 (↑0.65%)	0.5945 (↑0.59%)

complement-aware augmenter and performs similarly to the GSAT and DisC baselines. This observation illustrates the effectiveness of employing contrastive learning constraints to satisfy the principles of sufficiency and independence of rationalization for extracting faithful rationales. More experimental results are shown in Appendix E.2.

4.5. Structural Generalizability of FedGR (RQ3).

From Figure 2, we can observe that our two data augmenters are decoupled from the model structure. This insight leads to an interesting research question: *Can our difference-aware augmenter enhance the performance of other rationalization methods in FL scenarios?* To investigate this, we replace the complement-aware augmenter in FedGR with DisC, RGDA, GSAT, InterRAT and DARE, and conduct experiments on the OGB dataset. From Table 2, we observe a consistent improvement in performance across all rationalization methods when our difference-aware augmenter is employed. This finding suggests that our FedGR framework possesses generalizability and can effectively aid other rationalization methods in achieving better performance in FL scenarios. More experimental results are shown in Appendix E.3.

4.6. Training Process of FedGR (RQ4).

In this section, we investigate the training process of FedGR and Vanilla GR in Figure 5. Specifically, we take GIN as the backbone and present the AUC changes of the global and local model of FedGR and Vanilla GR on MolSIDER test set with communication rounds. Among them, the global model of FedGR and Vanilla GR are our main models that are tested in Table 1, and we only show one client's local model (i.e., Client1) in Figure 5. The performance of other local clients can be found in Appendix E.4. From the figure, we find that the Vanilla GR global model performs better than its local model Client1, which is consistent with Assumption 3.3 that local models are relatively biased compared to the global one. Besides, we also find that both local and global FedGR surpass Vanilla GR, which illustrates that data augmentations in FedGR can isolate shortcuts to compose faithful rationales and make predictions effectively.

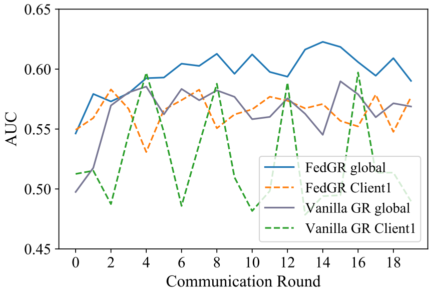


Figure 5. Training process of FedGR and Vanilla GR on MolSIDER, where the test set is considered as an unbiased test set.

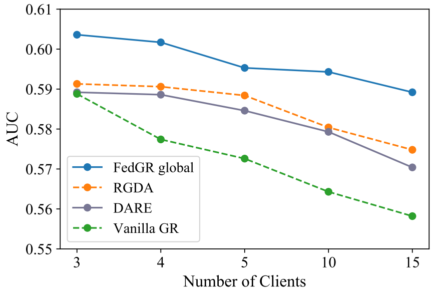


Figure 6. Performance of FedGR with different number of clients on MolSIDER.

4.7. Scalability of FedGR (RQ5).

Scalability is a critical factor in federated learning, and understanding the method's performance under such conditions would be valuable. Therefore, we conduct experiments by increasing the number of clients, and then experimental results are shown in Figure 6. From the figure, we can find that FedGR outperforms baselines as the number of clients increases, which demonstrates the scalability of FedGR.

5. Related Work

5.1. Graph Rationalization.

The success of GNNs has led to an increased research focus on the explainability of graph classification tasks (Veličković et al., 2017; Chen et al., 2022; Zhang et al., 2022; Li et al., 2022a; Yang et al., 2022; Peng et al., 2024; Yue et al., 2024). Among them, graph rationalization methods have gained significant attention. (Wu et al., 2022) first proposed a framework which involved dividing the graph into rationale and non-rationale subgraphs and using the rationale for prediction. Meanwhile, since (Chang et al., 2020) have shown that rationalization methods tend to exploit potential shortcuts in the data for prediction, (Wu et al., 2022) further proposed to discover invariant rationales. They utilized the structures of the non-rationale subgraphs as distinct environments and combined the rationale with different environments to generate new counterfactual samples. Based on this framework, several works have been developed (Fan et al., 2022; Liu et al., 2022; Sui et al., 2022; Li et al., 2022b). The main difference is that DIR explicitly considered non-rationale subgraph structures as potential environments while other methods used the non-rationale subgraph representations. Besides, many research works have started with the structures of rationalizations (Yu et al., 2021; Miao et al., 2022; Seo et al., 2023). Among them, (Yue et al., 2022) proposed a self-guided framework that to extract rationales by encapsulating sufficient information from the input. While rationalization methods have been extensively explored on centralized datasets, their applications to FL scenarios are not well explored.

5.2. Federated Learning.

Federated Learning (FL) algorithms (McMahan et al., 2017; Yang et al., 2019; Tan et al., 2022) have gained significant attention due to their ability to address data security and privacy concerns. Recently, there has been a growing interest in developing methods to eliminate spurious correlations in the training data. (Ezzeldin et al., 2023) proposed a FL framework aimed at mitigating the spurious correlations and preventing the trained model from being biased towards a particular demographic group. (Xu et al., 2023) introduced a bias-eliminating augmentation method in the FL setting. They identified and introduced desirable causal and shortcut attributes to augmented samples, aiming to reduce spurious correlations. While these methods have shown promising results in addressing spurious correlations, the problem of shortcuts in rationalization methods in FL scenarios remains relatively unexplored.

6. Conclusion

In this paper, we proposed a Federated Graph Rationalization (FedGR) with anti-shortcut augmentations method to achieve self-explaining GNNs. The method includes two types of augmenters: the complement-aware and the difference-aware augmenter, which are designed to generate shortcut conflicted samples to further address the problem of local shortcuts. For complement-aware augmenter, we first partitioned the graph into the rationale and complement subgraphs. Then, conditioned on satisfying the sufficiency and independence principles of rationalization, we randomly permuted the complement with the rationale to conduct shortcut conflicted samples. For difference-aware augmenter, it utilized the assumption that local models were more likely to utilize shortcuts compared to global models when making predictions. It generated shortcut conflicted samples that cross the decision boundary of the local model while preserving the predictions of the global model. Finally, we employed all anti-shortcut samples to yield the task results and compose rationales. Experimental results have clearly demonstrated the effectiveness of FedGR.

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 (No. 62337001) and the Fundamental Research Funds for the Central Universities.

Impact Statement

There is a growing interest in the field of explaining the results of graph classification generated by GNNs. Graph rationalizations have emerged as a means to provide intuitive explanations supporting the prediction results. The advantage of FedGR over the other rationalization approaches is that it can eliminate shortcuts or spurious correlations in data, thereby composing faithful rationales. More essentially, since our method removes shortcuts from data in federated scenarios, it can be applied to multiple decision-critical and privacy-sensitive systems, such as the healthcare system. Furthermore, it is crucial to note that our method solely provides suggestions for decision-making and enhances the credibility of model predictions, while refraining from interfering with real-world decision-making processes. Overall, we believe the positive influence of our work outweighs the potential negative impacts.

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. S. Invariant rationalization. In *Proceedings of the 37th International Conference on Machine Learning, (ICML)*, 2020.

Chen, Y., Zhang, Y., Bian, Y., Yang, H., Kaili, M., Xie, B., Liu, T., Han, B., and Cheng, J. Learning causally invariant representations for out-of-distribution generalization on graphs. *Advances in Neural Information Processing Systems*, 35:22131–22148, 2022.

DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R., and Wallace, B. C. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2020.

Ezzeldin, Y. H., Yan, S., He, C., Ferrara, E., and Avestimehr, A. S. Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7494–7502, 2023.

Fan, S., Wang, X., Mo, Y., Shi, C., and Tang, J. Debiasing graph neural networks via learning disentangled causal substructure. 2022.

Fu, X., Zhang, B., Dong, Y., Chen, C., and Li, J. Federated graph machine learning: A survey of concepts, techniques, and applications. *ACM SIGKDD Explorations Newsletter*, 24(2):32–47, 2022.

Glymour, M., Pearl, J., and Jewell, N. P. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

He, C., Balasubramanian, K., Ceyani, E., Yang, C., Xie, H., Sun, L., He, L., Yang, L., Philip, S. Y., Rong, Y., et al. Fedgraphnn: A federated learning benchmark system for graph neural networks. In *ICLR 2021 Workshop on Distributed and Private Machine Learning (DPML)*, 2021.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Jang, E., Gu, S., and Poole, B. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Knyazev, B., Taylor, G. W., and Amer, M. Understanding attention and generalization in graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Li, H., Wang, X., Zhang, Z., and Zhu, W. Out-of-distribution generalization on graphs: A survey. *arXiv preprint arXiv:2202.07987*, 2022a.

Li, H., Zhang, Z., Wang, X., and Zhu, W. Learning invariant graph representations for out-of-distribution generalization. In *Advances in Neural Information Processing Systems*, 2022b.

Li, S., Wang, X., Zhang, A., Wu, Y., He, X., and Chua, T.-S. Let invariant rationale discovery inspire graph contrastive learning. In *International conference on machine learning*, pp. 13052–13065. PMLR, 2022c.

Liu, G., Zhao, T., Xu, J., Luo, T., and Jiang, M. Graph rationalization with environment-based augmentations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1069–1078, 2022.

Liu, G., Inae, E., Luo, T., and Jiang, M. Rationalizing graph neural networks with data augmentation. *ACM Transactions on Knowledge Discovery from Data*, 2023.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Miao, S., Liu, M., and Li, P. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*, pp. 15524–15543. PMLR, 2022.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Peng, J., Liu, Q., Yue, L., Zhang, Z., Zhang, K., and Sha, Y. Towards few-shot self-explaining graph neural networks. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery*, 2024.

Seo, S., Kim, S., and Park, C. Interpretable prototype-based graph information bottleneck. *Advances in Neural Information Processing Systems*, 2023.

Sui, Y., Wang, X., Wu, J., Lin, M., He, X., and 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*, pp. 1696–1705, 2022.

Tan, A. Z., Yu, H., Cui, L., and Yang, Q. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. *6th International Conference on Learning Representations*, 2017.

Wang, T., Huang, J., Zhang, H., and Sun, Q. Visual commonsense r-cnn. In *Proceedings of CVPR*, pp. 10760–10770, 2020.

Wu, Y.-X., Wang, X., Zhang, A., He, X., and seng Chua, T. Discovering invariant rationales for graph neural networks. In *ICLR*, 2022.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Xu, Y.-Y., Lin, C.-S., and Wang, Y.-C. F. Bias-eliminating augmentation learning for debiased federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20442–20452, 2023.

Yang, N., Zeng, K., Wu, Q., Jia, X., and Yan, J. Learning substructure invariance for out-of-distribution molecular representations. In *Advances in Neural Information Processing Systems*, 2022.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Yehudai, G., Fetaya, E., Meirom, E., Chechik, G., and Maron, H. From local structures to size generalization in graph neural networks. In *International Conference on Machine Learning*, pp. 11975–11986. PMLR, 2021.

Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Yu, J., Xu, T., Rong, Y., Bian, Y., Huang, J., and He, R. Graph information bottleneck for subgraph recognition. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Yue, L., Liu, Q., Du, Y., An, Y., Wang, L., and Chen, E. Dare: Disentanglement-augmented rationale extraction. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26603–26617, 2022.

Yue, L., Liu, Q., Wang, L., An, Y., Du, Y., and Huang, Z. Interventional rationalization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 11404–11418, 2023.

Yue, L., Liu, Q., Liu, Y., Gao, W., Yao, F., and Li, W. Cooperative classification and rationalization for graph generalization. In *Proceedings of the ACM Web Conference*, volume 2024, 2024.

Zhang, Z., Liu, Q., Wang, H., Lu, C., and Lee, C.-K. Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems*, 34:15870–15882, 2021.

Zhang, Z., Liu, Q., Wang, H., Lu, C., and Lee, C. Protgnn: Towards self-explaining graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9127–9135, 2022.

A. Proof of Theorem 3.6

Theorem 3.6. *To train the difference-aware augmenter, minimizing term ① in Eq(9) contributes to* $\max I(Y_k; \tilde{G}_k | f^t(\cdot))$*; maximizing term ② in Eq(9) contributes to* $\min I(Y_k; \tilde{G}_k | f_k^{t-1}(\cdot))$.

Proof. According to the objective of Eq(8), equivalently, with the introduction of a Lagrange multiplier β, we can maximize the objective:

$$\begin{aligned} &\max I(Y_k; \tilde{G}_k \mid f^t(\cdot)) - \beta I(Y_k; \tilde{G}_k \mid f_k^{t-1}(\cdot)) \\ =&\max_{\Psi} I(Y_k; \Psi(G_k) \mid f^t(\cdot)) - \beta I(Y_k; \Psi(G_k) \mid f_k^{t-1}(\cdot)). \end{aligned} \tag{11}$$

Then, to calculate Eq(11), we should calculate the conditional mutual information. Below, we present how to calculate it with a general form:

$$\begin{aligned} I(Y; X \mid E) &= H(Y \mid E) - H(Y \mid E, X) \\ &= H(Y) - H(Y \mid E, X) \\ &\Rightarrow -H(Y \mid E, X). \end{aligned} \tag{12}$$

where $H(\cdot)$ denotes the entropy and Y is the given label.

Based on Eq(12), we can have:

$$\begin{aligned} I(Y_k; \Psi(G_k) \mid f^t(\cdot)) \Rightarrow -H(Y_k \mid f^t(\cdot), \Psi(G_k)) = -H(Y_k \mid f^t(\Psi(G_k))), \\ I(Y_k; \Psi(G_k) \mid f_k^{t-1}(\cdot)) \Rightarrow -H(Y_k \mid f_k^{t-1}(\cdot), \Psi(G_k)) = -H(Y_k \mid f_k^{t-1}(\Psi(G_k))). \end{aligned} \tag{13}$$

By incorporating Eq(13) into Eq(11), we can achieve:

$$\begin{aligned} &\max_{\Psi} I(Y_k; \Psi(G_k) \mid f^t(\cdot)) - \beta I(Y_k; \Psi(G_k) \mid f_k^{t-1}(\cdot)) \\ \Rightarrow &\max_{\Psi} -H(Y_k \mid f^t(\Psi(G_k))) + \beta H(Y_k \mid f_k^{t-1}(\Psi(G_k))) \\ =&\min_{\Psi} [\ell(f^t(\Psi(G_k)), Y_k) \, - \, \beta \ell(f_k^{t-1}(\Psi(G_k)), Y_k)], \end{aligned}$$

which finishes the proof.

B. Training algorithm of FedGR

The overall training algorithm of FedGR with anti-shortcut augmentations is presented in Algorithm 1.

C. Comparison Methods and Experimental Setups

In this section, we present the detailed description of comparison methods and experimental setups.

C.1. Comparison Methods

Although graph rationalization methods are widely studied, these methods are not fully explored in federated scenarios. For a fair comparison, we transfer the following rationalization methods that are employed in centralized scenarios to federated scenarios with the aggregation approach of Eq(1):

- **Vanilla GR** denotes the vanilla graph rationalization method presented in section 2.2.
- **DIR** (Wu et al., 2022) conducts interventions on the training distribution to create multiple counterfactual samples to compose rationales.
- **DisC** (Fan et al., 2022) designs a disentangling method to learn the causal and shortcut substructures within the graph data. By synthesizing counterfactual training samples, DisC aims to further de-correlate causal and shortcut variables, mitigating the influence of shortcuts.

Algorithm 1 Training algorithm of FedGR

```
Server Executes:
Initialize the warm-up communication round $T_w$ as 1, the communication round $T$, the epoch $E$, the numbers of clients $N$ and the shared global/local model $f^0(\cdot)$.
for each communication round $t$=1 to $T_w + T$ do
  for each client id $k$=1 to $N$ in parallel do
    if $t \le T_w$ then
      ClientUpdate($k$,$f_k^{t-1}(\cdot)$).
    else
      ClientUpdate($k$,$f_k^{t-1}(\cdot)$,$f^t(\cdot)$).
    end if
  end for
  Receive all local updated model: $\left\{f_k^t(\cdot)\right\}_{k=1}^N$.
  Perform aggregation by Eq.(1) to get $f^{t+1}(\cdot)$.
end for
ClientUpdate($k$, $f_k^{t-1}(\cdot)$, $f^t(\cdot)$=None):
for epoch $e$=1 to $E$ do
  if $f^t(\cdot)$ is None then
    Update local model by Eq.(8).
  else
    1. Train difference-aware augmenter $\Psi(\cdot)$ by Eq.(9).
    2. Employ the freezed $\Psi(\cdot)$ to generate $\tilde{G}_k$ for each $G_k$.
    3. Update local model with the mixed data by Eq.(10).
  end if
end for
Return local parameters $f_k^t(\cdot)$ to server.
```

- **CAL** (Sui et al., 2022) discovers the causal rationales and mitigates the confounding effect of shortcuts with a causal attention learning strategy.
- **GSAT** (Miao et al., 2022) introduces stochasticity to block label-irrelevant information in the graph and selectively identifies label-relevant subgraphs. This method is guided by the information bottleneck principle (Tishby et al., 2000; Alemi et al., 2017) to extract interpretable and relevant rationales.
- **DARE** (Yue et al., 2022) introduces a self-guided method with the disentanglement operation with the mutual information minimization to encapsulate sufficient information from the input to extract rationales. Although DARE is designed for explaining natural language understanding (NLU) tasks, we can naturally apply it to explain GNNs.
- **InterRAT** (Yue et al., 2023) develops an interventional rationalization to remove the spurious correlations in data and further discover the causal rationales with the backdoor adjustment method (Glymour et al., 2016; Wang et al., 2020). Similar to DARE, we transfer InterRAT from NLU to graph-level classifications.
- **RGDA** (Liu et al., 2023) propose 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, which generates counterfactual samples by combining the causal substructure with the shortcut substructure.

Besides, in the comparative analysis, several conventional GNN architectures are considered for classification tasks, including GCN (Kipf & Welling, 2017) and GIN (Xu et al., 2019). Meanwhile, we employ both GCN and GIN as the backbone of FedGR and other baselines.

C.2. Experimental Setups

In all experimental settings, the values of the hyperparameters λ_{sp}, λ_c, λ_e and λ_d are uniformly set to 0.01, 1.0, 1.0 and 1.0, respectively. The hidden dimensionality d is 32 for the Spurious-Motif dataset, and 128 for the OGB dataset. The

Table 3. Statistics of Spurious-Motif Datasets. Among them, different clients share the same valid and test set.

	Spurious-Motif		
	b=0.5	b=0.7	b=0.9
Client1/Client2/Client3/Val/Test	377/662/1961/3,000/6,000	377/662/1,961/3,000/6,000	377/662/1,961/3,000/6,000
Classes	3	3	3
Avg. Nodes	18.60/18.29/18.48/18.50/88.80	18.73/18.27/18.8/18.50/88.80	19.02/18.54/18.66/18.50/88.80
Avg. Edges	27.72/27.31/27.55/27.54/125.14	28.29/27.3/28.05/27.54/125.14	28.74/27.63/27.81/27.54/125.14

Table 4. Statistics of OGB Datasets.

	MolHIV	MolToxCast
Client1/Client2/Client3/Client4/Val/Test	9,380/6,148/10,113/7,260/4,113/4,113	871/614/3,819/1,556/858/858
Classes	2	617
Avg. Nodes	25.31/25.32/25.15/25.27/27.79/25.27	16.41/16.86/16.63/16.91/26.17/28.19
Avg. Edges	54.19/54.2/53.89/54.15/61.05/55.59	32.91/33.93/33.45/33.99/56.09/60.71
	MolBBBP	MolSIDER
Client1/Client2/Client3/Client4/Val/Test	472/299/325/535/204/204	422/333/201/185/143/143
Classes	2	27
Avg. Nodes	22.44/22.15/22.34/22.81/33.20/27.51	28.85/30.96/30.97/29.7/43.24/53.27
Avg. Edges	48.42/47.53/48.05/49.19/71.84/59.75	60.53/64.77/64.87/62.25/91.85/112.66

original node feature dimensionality d_g is 4 for the Spurious-Motif dataset, and 9 for the OGB dataset. During the training process, we employ the Adam optimizer (Kingma & Ba, 2014) with a learning rate initialized as 1e-2 for the Spurious-Motif, and 1e-3 for the OGB dataset. We set the predefined sparsity α as 0.1 for MolHIV, 0.5 for MolSIDER, MolToxCast and MolBBBP, and 0.4 for other datasets. The communication round T is 20 and the epoch in each communication is 10, for a total of 200 iterations.

D. Data Statistics

We evaluate our FedGR on three synthetic datasets from Spurious-Motif (Ying et al., 2019; Wu et al., 2022), and four real-world datasets from Open Graph Benchmark (OGB) (Hu et al., 2020). Details of dataset statistics are summarized in Table 3 and Table 4. Among them, in the Spurious-Motif dataset, different clients share the same valid and test set.

E. More Experimental Results

E.1. Performance of FedGR with both the GIN and GCN backbone

To assess the efficacy of FedGR, we conduct a comparative analysis against various baseline methods in the task prediction. The results are presented in Table 5. The findings demonstrate that FedGR achieves optimal performance, highlighting the effectiveness of the two proposed data augmenters.

Additionally, it is worth noting that certain de-shortcut methods, such as DIR, CAL, and InterRAT, exhibit comparable performance to the traditional GIN and GCN models. This suggests that de-shortcut models designed for centralized scenarios may not be suitable for direct application in federated learning (FL) scenarios. Consequently, it emphasizes the significance of exploring de-shortcut rationalization approaches specifically tailored for FL settings.

Moreover, several data augmentation-based methods, including RGDA and DisC, demonstrate sub-optimal performance. This highlights the advantages of employing counterfactual data augmentation approaches in mitigating the shortcut problem that arises within FL scenarios.

Table 5. Performance on the Synthetic Dataset and Real-world Dataset in FL scenarios.

		Spurious-Motif (ACC)			OGB (AUC)			
		bias=0.5	bias=0.7	bias=0.9	MolHIV	MolToxCast	MolBBBP	MolSIDER
GIN is the backbone	GIN	0.3213 ± 0.0429	0.3489 ± 0.0442	0.2978 ± 0.0382	0.6927 ± 0.0308	0.6091 ± 0.0133	0.6226 ± 0.0133	0.5780 ± 0.0105
	Vanilla GR	0.3182 ± 0.0353	0.3681 ± 0.0359	0.3031 ± 0.0291	0.6985 ± 0.0155	0.6111 ± 0.0055	0.6339 ± 0.0142	0.5774 ± 0.0175
	DIR	0.3091 ± 0.0314	0.3298 ± 0.0148	0.2893 ± 0.0311	0.6731 ± 0.0337	0.6133 ± 0.0064	0.6245 ± 0.0098	0.5686 ± 0.0162
	DisC	0.4418 ± 0.0182	0.4481 ± 0.0381	0.3579 ± 0.0471	0.7212 ± 0.0201	0.6274 ± 0.0018	0.6561 ± 0.0121	0.5869 ± 0.0142
	CAL	0.4213 ± 0.0109	0.5289 ± 0.0087	0.4191 ± 0.0248	0.7039 ± 0.0113	0.6170 ± 0.0051	0.6575 ± 0.0076	0.5879 ± 0.0138
	GSAT	0.4281 ± 0.0328	0.5259 ± 0.0381	0.4194 ± 0.0338	0.7149 ± 0.0226	0.6255 ± 0.0030	0.6555 ± 0.0085	0.5952 ± 0.0082
	DARE	0.4483 ± 0.0193	0.4891 ± 0.0391	0.4288 ± 0.0977	0.7220 ± 0.0165	0.6289 ± 0.0059	0.6621 ± 0.0096	0.5886 ± 0.0113
	InterRAT	0.4191 ± 0.0943	0.5283 ± 0.0935	0.4281 ± 0.0189	0.7026 ± 0.0092	0.6095 ± 0.0028	0.6426 ± 0.0223	0.5842 ± 0.0078
	RGDA	0.4087 ± 0.0293	0.5089 ± 0.0198	0.4286 ± 0.0313	0.7246 ± 0.0085	0.6235 ± 0.0034	0.6605 ± 0.0157	0.5906 ± 0.0151
	FedGR	**0.4610 ± 0.0289**	**0.5538 ± 0.0398**	**0.4977 ± 0.0315**	**0.7387 ± 0.0186**	**0.6316 ± 0.0054**	**0.6690 ± 0.0174**	**0.6017 ± 0.0202**
	FedGR w/o diff	0.4493 ± 0.0238	0.5293 ± 0.0483	0.4333 ± 0.0471	0.7214 ± 0.0124	0.6222 ± 0.0055	0.6623 ± 0.0033	0.5886 ± 0.0047
	FedGR w/o com	0.4571 ± 0.0372	0.5438 ± 0.0551	0.4682 ± 0.0388	0.7321 ± 0.0233	0.6298 ± 0.0035	0.6668 ± 0.0048	0.5978 ± 0.0021
GCN is the backbone	GCN	0.3491 ± 0.0211	0.3348 ± 0.0384	0.3081 ± 0.0392	0.6983 ± 0.0154	0.6059 ± 0.0074	0.6380 ± 0.0143	0.5747 ± 0.0092
	Vanilla GR	0.3219 ± 0.0401	0.3589 ± 0.0292	0.3024 ± 0.0487	0.6998 ± 0.0111	0.6043 ± 0.0149	0.6393 ± 0.0080	0.5718 ± 0.0108
	DIR	0.3148 ± 0.0392	0.3173 ± 0.0471	0.2973 ± 0.0357	0.6912 ± 0.0103	0.5863 ± 0.0022	0.6282 ± 0.0136	0.5673 ± 0.0172
	DisC	0.4369 ± 0.0486	0.4584 ± 0.0378	0.3673 ± 0.0931	0.7342 ± 0.0186	0.6171 ± 0.0094	0.6406 ± 0.0039	0.5894 ± 0.0076
	CAL	0.4438 ± 0.0477	0.5173 ± 0.0462	0.4284 ± 0.0832	0.7485 ± 0.0127	0.6205 ± 0.0046	0.6524 ± 0.0257	0.5957 ± 0.0116
	GSAT	0.4394 ± 0.0915	0.5383 ± 0.0326	0.4398 ± 0.0534	0.7457 ± 0.0079	0.6125 ± 0.0032	0.6536 ± 0.0085	0.6004 ± 0.0246
	DARE	0.4472 ± 0.0471	0.4782 ± 0.0474	0.4327 ± 0.0372	0.7424 ± 0.0368	0.6094 ± 0.0089	0.6416 ± 0.0159	0.5968 ± 0.0292
	InterRAT	0.4064 ± 0.0471	0.5173 ± 0.0347	0.4377 ± 0.0362	0.7180 ± 0.0284	0.6079 ± 0.0095	0.6398 ± 0.0098	0.5827 ± 0.0083
	RGDA	0.4187 ± 0.0375	0.4987 ± 0.0744	0.4377 ± 0.0432	0.7293 ± 0.0166	0.6197 ± 0.0049	0.6456 ± 0.0061	0.5958 ± 0.0143
	FedGR	**0.4580 ± 0.0531**	**0.5526 ± 0.0624**	**0.4918 ± 0.0619**	**0.7571 ± 0.0104**	**0.6282 ± 0.0092**	**0.6693 ± 0.0149**	**0.6093 ± 0.0039**
	FedGR w/o diff	0.4488 ± 0.0831	0.5219 ± 0.0739	0.4485 ± 0.0365	0.7489 ± 0.0172	0.6188 ± 0.0038	0.6575 ± 0.0024	0.5983 ± 0.0035
	FedGR w/o com	0.4521 ± 0.0464	0.5397 ± 0.0348	0.4771 ± 0.0492	0.7532 ± 0.0143	0.6254 ± 0.0042	0.6645 ± 0.0044	0.6032 ± 0.0073

Table 6. Performance on the Synthetic Dataset and Real-world Datasets in centralized scenarios.

		Spurious-Motif (ACC)			OGB (AUC)			
		bias=0.5	bias=0.7	bias=0.9	MolHIV	MolToxCast	MolBBBP	MolSIDER
GIN is the backbone	GIN	0.3950 ± 0.0471	0.3872 ± 0.0531	0.3768 ± 0.0447	0.7447 ± 0.0293	0.6521 ± 0.0172	0.6584 ± 0.0224	0.5977 ± 0.0176
	Vanilla GR	0.4528 ± 0.0384	0.4971 ± 0.0482	0.4218 ± 0.0363	0.7324 ± 0.0131	0.6475 ± 0.0067	0.6579 ± 0.0045	0.5883 ± 0.0194
	DIR	0.4444 ± 0.0621	0.4891 ± 0.0761	0.4131 ± 0.0652	0.6303 ± 0.0607	0.5451 ± 0.0092	0.6460 ± 0.0139	0.4989 ± 0.0115
	DisC	0.4585 ± 0.0660	0.4885 ± 0.1154	0.3859 ± 0.0400	0.7731 ± 0.0101	0.6662 ± 0.0089	0.6963 ± 0.0206	0.5846 ± 0.0169
	CAL	0.4734 ± 0.0681	0.5541 ± 0.0323	0.4474 ± 0.0128	0.7339 ± 0.0077	0.6476 ± 0.0066	0.6582 ± 0.0397	0.5965 ± 0.0116
	GSAT	0.4517 ± 0.0422	0.5567 ± 0.0458	0.4732 ± 0.0367	0.7524 ± 0.0166	0.6174 ± 0.0069	0.6722 ± 0.0197	0.6041 ± 0.0096
	DARE	0.4843 ± 0.1080	0.4002 ± 0.0404	0.4331 ± 0.0631	0.7836 ± 0.0015	0.6677 ± 0.0058	0.6820 ± 0.0246	0.5921 ± 0.0260
	InterRAT	0.4628 ± 0.0234	0.5182 ± 0.0214	0.4983 ± 0.0523	0.7446 ± 0.0131	0.6531 ± 0.0045	0.6753 ± 0.0034	0.5821 ± 0.0031
	RGDA	0.4251 ± 0.0458	0.5331 ± 0.1509	0.4568 ± 0.0779	0.7714 ± 0.0153	0.6694 ± 0.0043	0.6953 ± 0.0229	0.5864 ± 0.0052
	CaA	**0.4963 ± 0.0311**	**0.5678 ± 0.0482**	**0.5518 ± 0.0318**	**0.7896 ± 0.0088**	**0.6716 ± 0.0021**	**0.6986 ± 0.0081**	**0.6056 ± 0.0172**
	CaA w/o cl	0.4567 ± 0.0563	0.5486 ± 0.0522	0.4532 ± 0.0724	0.7563 ± 0.0248	0.6593 ± 0.0038	0.6753 ± 0.0040	0.5853 ± 0.0131
GCN is the backbone	GCN	0.4091 ± 0.0398	0.3772 ± 0.0763	0.3566 ± 0.0323	0.7128 ± 0.0188	0.6497 ± 0.0114	0.6665 ± 0.0242	0.6108 ± 0.0075
	Vanilla GR	0.4434 ± 0.0518	0.4513 ± 0.0558	0.4482 ± 0.0359	0.7421 ± 0.0144	0.6482 ± 0.0034	0.6631 ± 0.0074	0.5857 ± 0.0064
	DIR	0.4281 ± 0.0520	0.4471 ± 0.0312	0.4588 ± 0.0840	0.4258 ± 0.1084	0.5077 ± 0.0094	0.5069 ± 0.1099	0.5224 ± 0.0243
	DisC	0.4698 ± 0.0408	0.4312 ± 0.0358	0.4713 ± 0.1390	0.7791 ± 0.0137	0.6626 ± 0.0055	**0.7061 ± 0.0105**	0.6110 ± 0.0091
	CAL	0.4245 ± 0.0152	0.4355 ± 0.0278	0.3654 ± 0.0064	0.7501 ± 0.0094	0.6006 ± 0.0031	0.6635 ± 0.0257	0.5559 ± 0.0151
	GSAT	0.3630 ± 0.0444	0.3601 ± 0.0419	0.3929 ± 0.0289	0.7598 ± 0.0085	0.6124 ± 0.0082	0.6437 ± 0.0082	0.6179 ± 0.0041
	DARE	0.4609 ± 0.0648	0.5035 ± 0.0247	0.4494 ± 0.0526	0.7523 ± 0.0041	0.6618 ± 0.0065	0.6823 ± 0.0068	0.6192 ± 0.0079
	InterRAT	0.4521 ± 0.0471	0.5211 ± 0.0578	0.4379 ± 0.0345	0.7583 ± 0.0137	0.6583 ± 0.0048	0.6519 ± 0.0063	0.5938 ± 0.0038
	RGDA	0.4687 ± 0.0855	0.5467 ± 0.0742	0.4651 ± 0.0881	0.7816 ± 0.0079	0.6622 ± 0.0045	0.6970 ± 0.0089	0.6133 ± 0.0239
	CaA	**0.4831 ± 0.0571**	**0.5793 ± 0.0284**	**0.5128 ± 0.0482**	**0.7857 ± 0.0043**	**0.6664 ± 0.0040**	0.6949 ± 0.0072	**0.6212 ± 0.0102**
	CaA w/o cl	0.4682 ± 0.0783	0.5461 ± 0.0641	0.4521 ± 0.0739	0.7498 ± 0.0139	0.6558 ± 0.0057	0.6637 ± 0.0048	0.6085 ± 0.0146

Table 7. Structural Generalizability of FedGR. Each rationalization method in FedGR is highlighted with a gray background.

		MolHIV	MolToxCast	MolBBBP	MolSIDER
GIN is the backbone	DisC	0.7212	0.6274	0.6561	0.5869
	DisC+FedGR	0.7313 (↑1.01%)	0.6301 (↑0.27%)	0.6618 (↑0.57%)	0.5942 (↑0.73%)
	RGDA	0.7246	0.6235	0.6605	0.5906
	RGDA+FedGR	0.7344 (↑0.98%)	0.6326 (↑0.91%)	0.6673 (↑0.68%)	0.6008 (↑1.02%)
	GSAT	0.7149	0.6255	0.6555	0.5952
	GSAT+FedGR	0.7267 (↑1.18%)	0.6293 (↑0.38%)	0.6628 (↑0.73%)	0.5980 (↑0.28%)
	InterRAT	0.7026	0.6095	0.6426	0.5842
	InterRAT+FedGR	0.7193 (↑1.67%)	0.6245 (↑1.50%)	0.6587 (↑1.61%)	0.5927 (↑0.85%)
	DARE	0.7220	0.6289	0.6621	0.5886
	DARE+FedGR	0.7291 (↑0.71%)	0.6331 (↑0.42%)	0.6686 (↑0.65%)	0.5945 (↑0.59%)
GCN is the backbone	DisC	0.7342	0.6171	0.6406	0.5894
	DisC+FedGR	0.7467 (↑1.25%)	0.6203 (↑0.32%)	0.6488 (↑0.82%)	0.5965 (↑0.71%)
	RGDA	0.7293	0.6197	0.6456	0.5958
	RGDA+FedGR	0.7381 (↑0.88%)	0.6254 (↑0.57%)	0.6568 (↑1.12%)	0.6032 (↑ 0.74%)
	GSAT	0.7457	0.6125	0.6536	0.6004
	GSAT+FedGR	0.7419 (↓ 0.38%)	0.6234 (↑1.09%)	0.6635 (↑0.99%)	0.6077 (↑0.73%)
	InterRAT	0.7180	0.6079	0.6398	0.5827
	InterRAT+FedGR	0.7346 (↑1.66%)	0.6183 (↑1.04%)	0.6578 (↑ 1.80%)	0.5967 (↑1.40%)
	DARE	0.7424	0.6094	0.6416	0.5968
	DARE+FedGR	0.7491 (↑0.67%)	0.6172 (↑0.78%)	0.6587 (↑1.71%)	0.6043 (↑0.75%)

E.2. Performance of Complement-aware augmenter in centralized scenarios

In section 4.3, we validate the effectiveness of the complement-aware augmenter (referred to as CaA) through ablation experiments. In this section, we delve further into the performance of CaA. Specifically, in section 3, we mention that CaA can be naturally applied to centralized scenarios. To investigate this, we conduct experiments on centralized scenarios, and the results are presented in Table 6.

Upon observation, we find that our approach consistently outperforms the baselines in the centralized scenarios. This result demonstrates the effectiveness of our complement-aware augmenter in both the centralized and federated learning (FL) scenarios. Furthermore, we conduct experiments by removing the contrastive constraint (denoted as CaA w/o cl). From the table, we can observe that CaA w/o cl exhibits a significant decrease in performance compared to the complement-aware augmenter, performing similarly to the GSAT and DisC baselines. This finding highlights the importance of incorporating contrastive learning constraints to satisfy the principles of sufficiency and independence in rationalization methods.

E.3. Structural Generalizability of FedGR

To investigate that FedGR can contribute to the performance improvements in existing rationale-based methods, we conducte experiments on the OGB dataset by replacing the complement-aware augmenter in FedGR with DisC, RGDA, GSAT, InterRAT, and DARE. The results are presented in Table 7.

Upon analyzing the table, we observe a consistent improvement in performance across all rationalization methods when our difference-aware augmenter is employed in the FedGR framework. This finding highlights the generalizability of our FedGR approach and its ability to effectively enhance the performance of other rationale-based methods in FL scenarios.

E.4. Training Process of FedGR

In this section, we delve into the training process of FedGR and Vanilla GR, as depicted in Figure 7. The experiments are conducted using GIN as the backbone, and we present the changes in AUC for the global and local models of FedGR and Vanilla GR on the MolSIDER test set across communication rounds. It is important to note that the global models of FedGR and Vanilla GR are the main models evaluated in Table 1 and Table 6.

Analyzing the figure, we observe that the global model of Vanilla GR outperforms all of its local models. This finding aligns with Assumption 3.3 that local models tend to exhibit relatively higher bias compared to the global model. Furthermore, we notice that both the local and global models of FedGR surpass Vanilla GR in terms of performance. This observation highlights the efficacy of the data augmentations utilized in FedGR, which can isolate shortcuts and compose faithful rationales, thereby enabling effective predictions.

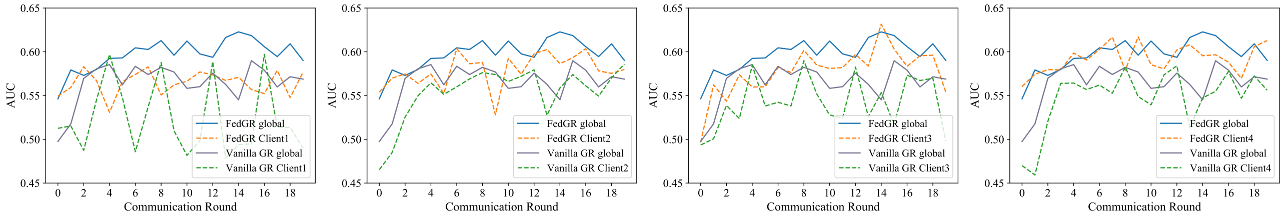


Figure 7. Training process of FedGR and Vanilla GR on Mol-SIDER, where the test set is considered as an unbiased test set.

E.5. Why the performance of DIR in the centralized scenario is lower than in the FL scenario?

Although DIR trained by federated dataset (DIR-fed) outperforms DIR trained by centralized dataset (DIR-center) on the OGB dataset, we also find that on the Spurious-Motif dataset, DIR-center performs higher than DIR-fed. Based on the above observation, we argue that one possible reason is due to the dataset itself. To this end, we additionally add experiments on other datasets, such as Graph-SST2 (Wu et al., 2022) and MNIST-75sp (Knyazev et al., 2019) datasets, where the training set is partitioned into 4 clients with $\gamma = 4$.

Table 8. The performance of DIR in both Graph-SST2 and MNIST-75sp datasets under the centralized and federated scenarios, where DIR is implemented with GIN.

GIN is the backbone	Graph-SST2(ACC)	MNIST-75sp(ACC)
DIR-fed	0.7877 ± 0.0282	0.1384 ± 0.0394
DIR-center	0.8083 ± 0.0115	0.1893 ± 0.0458

From Table 8, we observe that DIR-center performs higher than DIR-fed on both Graph-SST2 and MNIST-75sp datasets. Therefore, we can conclude the problem of DIR's results may be caused by the dataset itself.

Besides, we also implement our FedGR on these two datasets.

Table 9. Performance on the Graph-SST2 and MNIST-75sp datasets in FL scenarios.

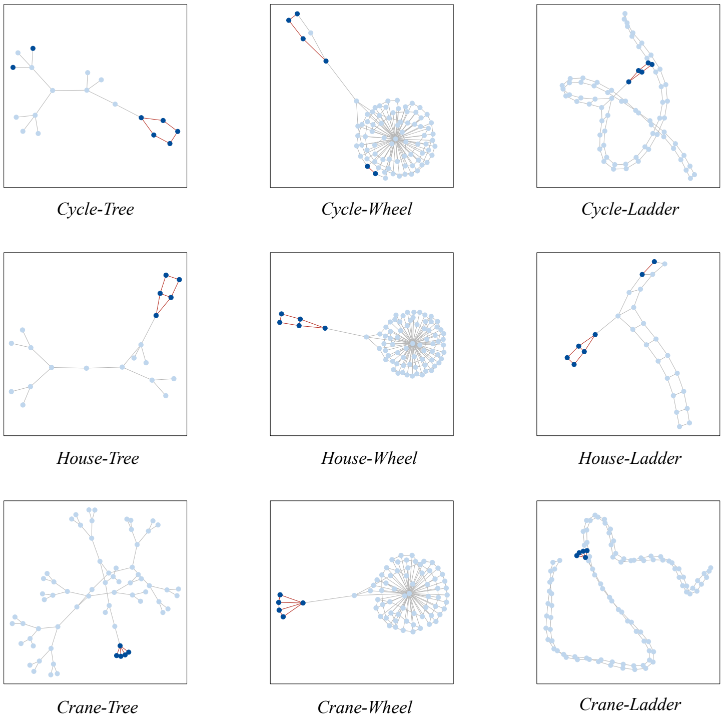
GIN is the backbone	Graph-SST2(ACC)	MNIST-75sp(ACC)
GIN	0.7921 ± 0.0028	0.1123 ± 0.0039
VanillaGR	0.7783 ± 0.0129	0.1134 ± 0.0068
DIR	0.7877 ± 0.0282	0.1384 ± 0.0394
DisC	0.8124 ± 0.0086	0.1308 ± 0.0184
CAL	0.8173 ± 0.0138	0.1347 ± 0.0038
GSAT	0.8283 ± 0.0080	0.1239 ± 0.0128
DARE	0.8219 ± 0.0238	0.1402 ± 0.0238
InterRAT	0.8192 ± 0.0048	0.1303 ± 0.0093
RGDA	0.8247 ± 0.0057	0.1455 ± 0.0129
FedGR	**0.8313 ± 0.0069**	**0.1683 ± 0.0135**

E.6. Case Study

In this section, we first present the visualization of FedGR, which is trained in Spurious-Motif (bias=0.9) on the test set. Specifically, Figure 8 shows several rationale subgraphs extracted by FedGR (GIN serves as the backbone). Among them, each graph consists of a motif type (*Cycle*, *House* and *Crane*) and a base (*Tree*, *Wheel* and *Ladder*). The highlighted navy blue nodes represent selected rationale nodes[1]. Meanwhile, we assume that if there is an edge between the two identified nodes, we visualize this edge as the red lines. From the figure, we can observe that FedGR successfully extracts more

[1] In this paper, when the probability of predicting a node as part of rationales $\tilde{m}_i$ to be greater than 0.55, we take the node as part of rationales.

accurate rationales for prediction. These visualizations highlight the effectiveness of the FedGR in composing accurate and faithful rationales from graph data, thereby enhancing the model's explainability and overall performance.



Cycle-Tree *Cycle-Wheel* *Cycle-Ladder*

House-Tree *House-Wheel* *House-Ladder*

Crane-Tree *Crane-Wheel* *Crane-Ladder*

Figure 8. Visualization of FedGR rationale subgraphs.