FACExplainer: Generating Model-faithful Explanations for Graph Neural Networks Guided by Spatial Information

Hua Yang
Sch. of Computer Science&Engineering
South China University of Technology
Guangzhou, China
cs_yanghua@mail.scut.edu.cn

C.L. Philip Chen
Sch. of Computer Science&Engineering
South China University of Technology
Pazhou Lab
Guangzhou, China
philip.chen@ieee.org

Bianna Chen
Sch. of Computer Science&Engineering
South China University of Technology
Guangzhou, China
csbianna@mail.scut.edu.cn

Tong Zhang*
Sch. of Computer Science&Engineering
South China University of Technology
Pazhou Lab
Guangzhou, China
tony@scut.edu.cn

***Abstract*—Graph neural networks (GNNs) have been widely applied in various decision-crucial fields, where accurate predictions with high interpretability are desired. Thus, numerous post-hoc explainers for GNNs have been proposed. However, some prioritize human-intelligible explanations through graph rules, such as the connection rule, which undermines the explanation's faithfulness to the model. This paper proposes an innovative method, FACExplainer, that re-examines the role of spatial information within GNNs for generating model-faithful explanations. FACExplainer employs activation maps from the last graph convolution to narrow down a compact search space. Our approach further identifies the subgraph that maximizes mutual information as the explanation, eliminating the need for domain-specific knowledge about the downstream task. Empirical analysis of FACExplainer on seven benchmark datasets with three classical GNNs reveals significantly improved explanation quality while consuming less time when compared to leading explainers.**

***Index Terms*—Interpretability, Graph Neural Network, Post-hoc Explainer**

I. Introduction

Graph neural networks (GNNs) are broadly used in the scientific field, including traffic forecasting [1], biology [2], and chemistry [3], to learn from graph-structured data. However, most GNNs are deployed without explanation, functioning as black boxes. The absence of explanations undermines the reliability of GNN predictions and reduces their applicability in decision-critical areas. For instance, if a deployed GNN makes a biased decision, it can potentially result in a significant financial loss. This risk can be mitigated by verifying GNN explanations against domain knowledge, i.e., ground truth. GNN explanations can also be helpful for model debugging and error analysis, i.e., if a precise explanation does not match the domain knowledge of the input data, the target GNN may process the data in an unexpected way.

Unlike text and images, graphs are irregular and contain complex structural information. Therefore, generating explanations for GNNs is time-consuming and requires substantial effort. Some recent studies [4]–[10] have attempted to explain GNNs, some of which focus on graph structure and propose graph rules to expedite the explanation process [6], [7]. For example, the connection rule (one of the most commonly used graph rules) underpins the notion that connected subgraph explanations are more intuitive and human-intelligible. Nonetheless, we argue that graph rules are unreliable due to suboptimal training results in practice, as we elaborate on in Section III-B. While characteristics derived from datasets are not appropriate for generating explanations, we shift our focus to another aspect of the prediction-generating process: the information in the GNN model's parameters. We propose to utilize the spatial information of GNNs, akin to the CAM-based approaches [11]–[13] utilized for images or text.

In this paper, we present Fast Adaptive Class Activation Mapping Explainer (FACExplainer). This novel GNNs explainer efficiently generates explanations by utilizing the spatial information of pre-trained GNNs. FACExplainer is built upon the successful CAM-based explainer in the image and text fields which utilizes semantic and spatial information in CNN's internal processing parameters. In the graph domain, GNN differs from CNN for its recursive neighborhood aggregation to handle the structural information. This difference makes directly applying CAM-based explainers [10] leads to incorrect explanations [7]. Therefore, we analyze the feature of spatial information of pre-trained GNNs and find an effective

*Tong Zhang is the corresponding author.

way to generate explanations by conducting a targeted search. Specifically, FACExplainer formulates an explanation-similar subgraph set using the activation maps of pre-trained GNNs. Then FACExplainer searches the subgraph set to provide an explanation that maximizes the mutual information (MI) with GNN's prediction. We conduct extensive experiments on synthetic and real-world benchmark datasets, showing FACExplainer to be significantly faster and more accurate than strong baselines. In summary, our contributions are:

- Identifies the limitation of graph rules that may cause model-unfaithful explanations.
- Re-examines the potential uses of the spatial information available in GNNs, as characteristics derived from datasets are unusable.
- Propose a new GNN explainer FACExplainer and demonstrate the superiority of FACExplainer over strong baselines for explaining GNNs on synthetic, social networks, chemical, and text graphs.

II. Preliminaries

A. Graph neural networks

We denote a graph as $G = (V, E)$, where V denotes the node set, and E denotes the edge set. The set of neighbors of a node v is denoted as $\mathcal{N}(v) = \{u \in V | (v, u) \in E\}$. According to the theoretical analysis conducted in [14], [15], GNNs follow a recursive neighborhood aggregation scheme and has a very close connection to Weisfeiler-Lehman (WL) graph isomorphism test [16]. In essence, GNNs make predictions on G by constructing multi-hop structure-aware representations of each node. To mathematically formalize this scheme, we define the representation of node v at k-th iteration as:

$$h_v^{(k)} = \phi^{(k)}(h_v^{(k-1)}, \psi^{(k)}(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\})), \quad (1)$$

where $\phi^{(k)}(\cdot)$ denotes the combination operation, e.g. summation, and $\psi^{(k)}(\cdot)$ denotes the aggregation operation.

B. Graph Rule

Providing an explanation G_e for G in graph learning tasks is a daunting challenge. Unlike images and texts, graph data contain complex structural information. Take MUTAG as an example. -NO_2 and -NH_2 serve as the ground truth for mutagenic graphs [5]. As if -NO_2 and -NH_2 exist, the molecule is often mutagenic. However, if we intend to quantify the importance of node -N directly, we will not obtain precise quantitative results as it neglects the interactions between nodes and edges [17]. Shapley values are a principled method of capturing complex interactions between nodes [7], [9], [18], [19]. The Shapley value is a concept from cooperative game theory used to assign total game gains to players fairly. However, the Shapley value needs to compute the interaction of node v with all $2^{|V|-1}$ subsets, thereby leading to computational difficulties.

Graph rules are derived from the characteristics of the data to guide the explanation process. Several graph rules have been introduced to speed up the explanation-generating process. For example, SubgraphX [7] believes that the explanation for GNN must be connected, which helps to ignore the non-connected subgraphs and accelerates the generation process. Similarly, GEM [6] incorporates several graph rules to aid the explanation process. As an example, for the MUTAG dataset, GEM assumes explanatory graphs should be connected.

C. Spatial Information and Grad-CAM

Spatial information in the last convolutional layers provides high-level semantics indicating the most important neurons for a given decision. For CNN layers, convolutional activation maps naturally retain spatial information, which is lost in fully-connected layers. Hence, the last convolutional layer in a CNN is generally considered to have the ideal balance between high-level semantics and detailed spatial information. Gradient-weighted Class Activation Mapping (Grad-CAM) [12] is a prominent explainer that uses the spatial information of CNN layers. However, Grad-CAM is reported as unsuitable for GNNs [7], [20] due to the difference between the GNN and CNN layers. We will further discuss this issue in Section III-C.

III. GNN explanation via subgraph importance scoring

A. Problem formalization

Following the idea in [7], [9], [19], [21], we formalize a GNNs explanation problem through feature importance scoring, where features may refer to nodes, edges, subgraphs of graphs, pixels of an image, or part words of text. The target pre-trained GNN model is denoted as $f(\cdot)$, and an input graph as $G = (V, E, X)$. $0 < \gamma < 1$ denotes a sparsity constraint to enforce a concise explanation. The common goal of explaining the GNN model is to find a subgraph that maximizes a given evaluation metric $\mathrm{EVAL}(\cdot, \cdot, \cdot)$, which measures the faithfulness of explanation G_e to input graph G regarding making predictions:

$$G_e^* = \underset{G_e \in G, |G_e| \leq \gamma |G|}{argmax} \mathrm{EVAL}(f(\cdot), G, G_e) \quad (2)$$

$[f(G)]_c$ denotes the probability on class c of model $f(\cdot)$ with input G. In practice, the objective is often relaxed to finding a set of important nodes or edges first and then inducing the important subgraph [4], [5], [9]. Therefore, a more tractable objective of finding the optimal set of nodes $S^* \in V$ is defined as:

$$S^* = \underset{S \subseteq V, |S| \leq \gamma |V|}{argmax} \mathrm{EVAL}(f(\cdot), G, S) \quad (3)$$

Many explainers for GNN can be expressed as in (3) with different $\mathrm{EVAL}(\cdot, \cdot, \cdot)$ function [7], [9].

B. The issue of using graph rule

Various graph principles are incorporated based on input data characteristics to expedite the process of generating explanations [6], [7]. However, the deployment of such explainers raises two significant issues. Firstly, not all datasets of interest have simple ground truth to utilize. For instance, a previous work [6] reported that statistics of MUTAG show that some non-mutagenic molecules contain -NO_2 and -NH_2 motifs.

Therefore, the accurate scientific ground truth of MUTAG is more intricate and challenging to use. Commonly, the datasets in biology or chemistry do not have accurate ground truth. Secondly, the ground truth of the dataset may not be suitable for pre-trained GNN due to suboptimal training [22]. Before producing explanations, the target GNN is trained to maximize the MI between model predictions Y (i.e., $f_\theta(G)$) and the labels $\mathcal{Y}$:

$$\theta^* \triangleq \underset{\theta}{argmax}\, \mathrm{MI}(f_\theta(G); \mathcal{Y}), \tag{4}$$

where θ denotes the parameters of the targeted GNN. The post-hoc explainer endeavors to explain the prediction Y by finding the optimal set of nodes that affects the prediction most, as shown in (3). However, the optimization process in (4) is usually suboptimal in practice [9], [23]. The suboptimal result indicates that using knowledge about the downstream task $\mathcal{Y}$ to guide the explanation-generating process about prediction Y can lead to inaccurate explanations. One of the fundamental uses of post-hoc explainers is facilitating model debugging and error analysis, which is highly compromised by assuming the target GNN is trained as expected. Furthermore, the explanation-generating process still needs an unacceptable amount of time, even with the graph rules above. Therefore, we naturally focus on a different aspect of the target prediction-generation process: the spatial information provided in the GNN's parameters. This is because that the spatial information of the pre-trained model reflects the process of the model's prediciton [12], which does not rely on the domain knowledge of downstream task.

C. Finding Explanation Guided by GNN's Spatial Information

If we take a global view of the explanation-generating process, it is unnecessary to compute the contribution of every node in the explanation-generating process. Instead, we only require a concise subgraph G_e generated by the optimal set of nodes S^* that maximizes the objective function in (3). Therefore, the explanation-generating process is a straightforward combinatorial problem.

a) The combination method.: The explainer should not assume any graph rules, including connectivity. Therefore, the explanation graph can be any subgraph of G. The most straightforward way to find G_e is to test every subgraph of G using (3). We can avoid dealing with complex interdependencies by directly searching for subgraphs that have the most significant impact on model decisions. However, this simple way is still time-consuming, and the time complexity is $\mathcal{O}(|V|^2)$, where $|V|$ is the number of nodes of the instance to be explained. We aim to find a faster solution to this problem by utilizing the spatial information of GNNs, such as activation and prediction probabilities. We begin by defining a type of node and then analyzing the features of the spatial information of GNNs. The convergence node, $v_{\pi i} \in V_{crg}$, is the top node ranked by the importance distribution provided by the spatial information. We formally define the set of convergence nodes, V_{crg}, for GNNs as follows:

$$V_{crg} = \{v_{\pi 1}, v_{\pi 2}, \cdots, v_{\pi k}\} \tag{5}$$

$$k = \lfloor \gamma |V| \rfloor \tag{6}$$

$$v_{\pi i} = \underset{\substack{v_{\pi i} \in V \\ v_{\pi i} \notin \{v_{\pi 1}, v_{\pi 2}, \cdots, v_{\pi (i-1)}\}}}{argmax} \mathcal{L}^c_{\text{GraphGrad-CAM}}(v_{\pi i}) \tag{7}$$

where the $\mathcal{L}^c_{\text{GraphGrad-CAM}}$ represents importance distribution of each node in G, as defined in (11). Then $\mathcal{L}^c_{\text{GraphGrad-CAM}}(v_{\pi i})$ is the importance of node $v_{\pi i}$. We first sort nodes in descending order according to their importance. Then we use the top $\lfloor \gamma |V| \rfloor$ nodes as the convergence nodes.

b) The Spatial Information of GNNs.: A convolutional layer's spatial information (activation map) differs between GNNs and CNNs. Previous work [10] applied Grad-CAM directly to GNNs. However, it failed to incorporate the unique properties of graph data and thus did not perform well in GNNs. We posit that this poor performance arises due to the information shift caused by the recursive neighborhood aggregation operation in (1).

We use v_e to denote the node in explanation G_e, and we assume that the target model has M graph convolutional layers. The prediction-important information in v_e can 'spread' to its M-hop neighbors through (1). Considering numerous nodes will have the prediction-important information, it is highly unlikely that spatial information will lead us directly to v_e. To illustrate this phenomenon, we use a chemical graph example extracted from MUTAG as shown in Fig. 1 (a). The ground truth is that this molecule is classified as mutagenic because of functional groups -NO_2 (marked with blue rectangles). Therefore, the spatial information of GNNs will guide our attention toward the convergence nodes between these functional groups rather than the groups themselves. We marked a possible convergence node with a red circle to show this process. This convergence node is located between three -NO_2 functional groups, and after performing the operation defined in (1) for five iterations, this node may obtain the prediction-important information from all three -NO_2 groups. Any nodes between those functional groups may act as the convergence nodes. Hence the spatial information is highly unreliable.

IV. FACExplainer: Fast Adaptive Class Activation Mapping Explainer

A. The potential use of the spatial information of GNNs

Although spatial information cannot be easily utilized for GNNs, it informs us of the location of the convergence nodes. The nodes that carry important prediction information are located in the M-hop neighbors of the convergence nodes. Based on the observation of spatial information of GNNs, the most straightforward approach is to test every induced subgraph using a subset of convergence nodes and their M-hop neighbors. Nevertheless, the time complexity needs to be lowered to be practical. Thus, we suggest using a more stringent set of subgraphs to find the explanations. As each

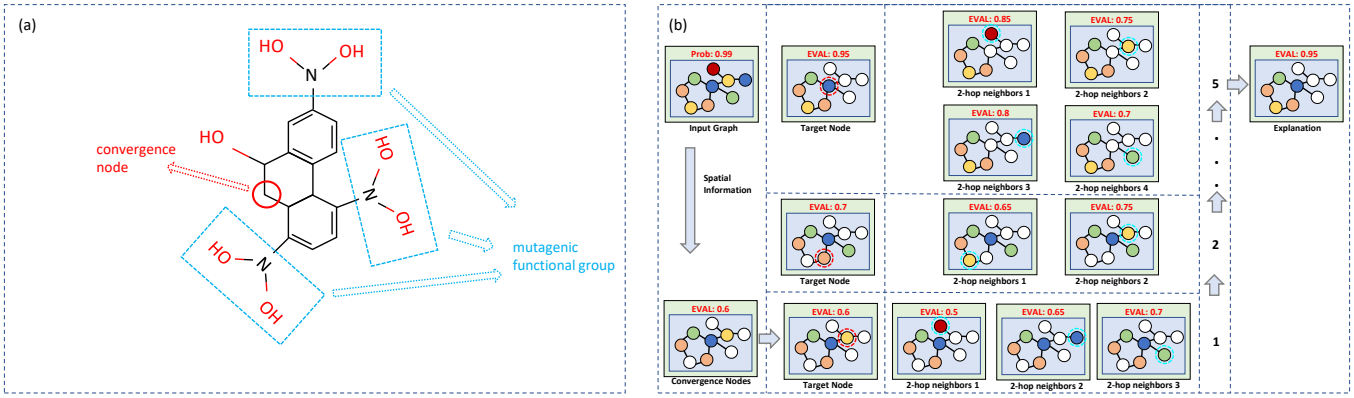

Fig. 1. (a) Chemical graph (right): A real chemical graph containing mutagenic functional groups -NO_2 selected from the MUTAG dataset (dataset description in Section V-A). The chemical graph contains three -NO_2 functional groups (marked with blue rectangles) with a possible convergence node (marked with a red circle). (b) Illustration of FACExplainer for explaining GNNs on a graph classification task. Initially, FACExplainer employs the activation output and gradient of the last graph convolutional layer to identify the convergence nodes in G. Subsequently, FACExplainer replaces each convergence node with each one of its M-hop neighbors ($M = 2$ for illustration) to evaluate if the convergence nodes induced subgraph scores larger EVAL. The nodes inside the red dashed circle represent target nodes waiting to be replaced, while the nodes inside the blue dashed circle represent 2-hop neighbors of the target nodes. Once all convergence nodes have been assessed, FACExplainer presents the final convergence nodes induced subgraph as the explanation subgraph.

convergence node acquires significant predictive information from its M-hop neighbors, we replace these nodes with each of their M-hop neighbors to evaluate the extent of the explanation. Due to the complexity of GNNs, the explanation we find by searching the tighter set is usually a local optimum; however, experimentally, we found that this simple procedure often leads to high-quality explanations effectively.

B. Adjusted Objective Function.

Here, we present our MI-inspired objective function. Unlike the function defined with Shapley value to evaluate the contribution of each node [7], [9], we do not need to define such a function as we only search for a compact subgraph as an explanation. Drawing upon the established approach in prior research [4], [5], [22], we consider the MI between explanation subgraphs and the model's prediction as the optimal objective. Formally, the MI-inspired scoring function $\text{EVAL}(\cdot,\cdot,\cdot)$ can be expressed as follows:

$$\begin{aligned}\text{EVAL}(f(\cdot), G, S) &= MI(f(G), G[S]) \\ &= H(Y) - H(Y|G = G[S]),\end{aligned} \tag{8}$$

where $G[S]$ denotes the induced subgraph of node set S in G. Here we follow the idea in [4], MI quantifies the change in the probability of GNN's prediction when the input graph is limited to explanation subgraph $G[S]$. Consequently, optimizing (8) is equivalent to maximizing the probability of GNN's prediction when the input graph is limited to $G[S]$, which can be expressed as follows:

$$G_e = G[S^*] \triangleq \underset{G[S]}{argmax}[f(G[S])]_c \tag{9}$$

However, there may be multiple sufficient explanations for the exact prediction [24], so we add another constraint to ensure the explanation is necessary. The adjusted objective function is:

$$G[S^*] \triangleq \underset{G[S]}{argmax}[(f(G[S])]_c - [f(G[V \setminus S])]_c, \tag{10}$$

where the $V \setminus S$ represents the node set obtained by eliminating the explanation node set from the input graph's node set. The term $[f(G[V \setminus S])]_c$ evaluates if the complement subgraph of the explanation subgraph is adequate for the target prediction. Thus, maximizing (10) guarantees that the explanations are both necessary and sufficient for the target prediction.

C. The FACExplainer Algorithm

In this section, we present our FACExplainer algorithm. Fig. 1 (b) depicts our proposed FACExplainer. FACExplainer formulates the GNN explanation task as a subgraph scoring problem, where subgraphs are evaluated to find the optimal node-induced subgraph to maximize (10). To incorporate the spatial information of GNNs, we use the GraphGrad-CAM:

$$\mathcal{L}^c_{\text{GraphGrad-CAM}} = Relu(\sum_{t=0}^{k} \alpha_t^c \mathcal{A}^t) \tag{11}$$

$$\alpha_t^c = \frac{1}{|V|} \sum_{i=0}^{|V|} \frac{\partial Y^c}{\partial \mathcal{A}_i^t} \tag{12}$$

where $\mathcal{L}^c_{\text{GraphGrad-CAM}}$ denotes the importance distribution for prediction Y^c and c is the target class. In addition, $\mathcal{A}^t$ denotes t-th activation map of the last graph convolutional layer, and $\mathcal{A}_i$ denotes the i-th node's value in $\mathcal{A}$. α_t^c captures the 'importance' of the feature map $\mathcal{A}^t$ for a specific target class c. After identifying the top $\lfloor \gamma |V| \rfloor$ nodes in $\mathcal{L}^c_{\text{GraphGrad-CAM}}$ as the convergence nodes, we explore the explanation-possible subgraphs by replacing each node in convergence nodes with one of its M-hop neighbors. The objective is to determine if the subgraph induced from this replacement increases the value of (10). We use the subgraph that maximizes (10) as an explanation. Notice that FACExplainer only uses the information in a model's parameters, resulting in an explanation that is highly faithful to the model. We provide the whole FACExplainer algorithm in Algorithm 1.

Algorithm 1: Procedure of the FACExplainer

Input: GNN model $f(\cdot)$ that have M layers; the prediction of target class Y^c; input graph $G = (V, E)$; the t-th activation of the last graph convolutional layer $\mathcal{A}^t$; a sparsity constraint γ

Output: explanation: subgraph $G_e = G[S^*]$

```
Compute the adjusted spatial information: L^c_{GraphGrad-CAM} following (11) and (12)
Find the convergence nodes: V_crg following (5), (6), and (7)
Initialize the explanation: S = V_crg;
Compute scores: s_now = [(f(G[S])]_c − [f(G[V \ S])]_c
for node τ ∈ V_crg do
    Find τ's M-hop neighbors set: N^M(v)
    for node ξ ∈ N^M(v) ∩ ξ ∉ S do
        replace τ ∈ V_crg with ξ: S' = S ⋃ {ξ} \ {τ}
        Compute new score:
          s_new = [(f(G[S'])]_c − [f(G[V \ S'])]_c
        if s_now ≤ s_new then
            s_now = s_new
              S = S'
        end
    end
end
return S* = S
```

V. Experiments

A. Experiment settings

We provide an overview of our datasets, baseline methods, and experimental setups. We train all models to convergence before applying these explainers to explain GNN predictions.

a) Datasets.: We experimented extensively with datasets from various domains, including synthetic, social networks, chemical, and text datasets. nosep, leftmargin=9pt

- **Synthetic dataset.** BA2Motifs [5] consists of 800 graphs that use Barabasi-Albert graphs as base graphs and attach with "house" motifs or five-node cycle motifs. Both motifs are illustrated in the leftmost column of Fig. 3. The graphs are labeled with one of two classes based on the type of the attached motif.
- **Social networks datasets.** Redditbinary [25] contains 2000 graphs, each representing an online discussion thread, while the nodes represent users. An edge indicates at least one interaction or comment among the users. On average, there are 429.6 nodes in each graph. Each graph is labeled according to whether it belongs to a community or subreddit.
- **Text sentiment graph datasets.** GraphSST2 and Twitter [20] contains graphs constructed from words and phrases with understandable, semantic meanings. Each node represents a word, while the edges reflect the relationships between different words. The graphs are labeled as having positive or negative sentiments.
- **Chemical property prediction datasets.** MUTAG [26] and BBBP [27] contains real chemical molecule graphs for graph classification. With atoms as nodes and bonds as edges, the label of each chemical graph is determined by the molecule's mutagenic effect in MUTAG. In BBBP, each graph contains binary labels for its permeability property.

b) Explanation baselines.: We compare our approach with six strong baselines representing the state-of-the-art (SOTA) methods for GNN explanation: GNNExplainer [4], PGExplainer [5], OrphicX [8], Grad-CAM [10], GStarX [9], and SubgraphX [7].

c) Evaluation metrics.: It is important to have precise evaluation metrics due to the absence of accurate ground truth and suboptimal training of GNNs. We follow previous studies [7], [9], [20] to employ Fidelity, Inverse Fidelity (Inv-Fidelity), and Sparsity as evaluation metrics. Fidelity and Inv-Fidelity measure whether the explanation is faithful to the model's prediction by either removing the important structures or keeping only the important ones. Sparsity measures whether the explanations are sparse enough by computing the fraction of explanations. Note that explanations with different Sparsity levels are not directly comparable. Ideal explanations should have high Fidelity and low Inv-Fidelity under the same level of Sparsity. The formulas for these metrics are given below:

$$\text{Fidelity}(\mathrm{G}, \mathrm{G_e}) = [f(G)]_c - [f(G \setminus G_e)]_c \tag{13}$$

$$\text{Inv-Fidelity}(\mathrm{G}, \mathrm{G_e}) = [f(G)]_c - [f(G_e)]_c \tag{14}$$

$$\text{Sparsity}(\mathrm{G}, \mathrm{G_e}) = 1 - |G_e|/|G| \tag{15}$$

Fidelity and Inv-Fidelity are complementary and are both critical for a good explanation. Since they are analogous to precision and recall, we follow the idea in [9], [24] to use a single-scalar-metric "harmonic fidelity" (H-Fidelity), where we normalize them by Sparsity and take their harmonic mean. The better explanation should have a higher H-Fidelity. We show formulas for normalized fidelity (N-Fidelity), normalized inverse fidelity (N-Inv-Fidelity), and harmonic fidelity (H-Fidelity) in (16). Please refer to [9] Appendix A.3 for more details about H-Fidelity. Here, we set $m_0 = \text{Fidelity}(\mathrm{G}, \mathrm{G_e}) \times (1 - \frac{|G_e|}{G})$ and $m_1 = \text{Inv-Fidelity}(\mathrm{G}, \mathrm{G_e}) \times (1 - \frac{|G_e|}{G})$.

$$\text{H-Fidelity}(\mathrm{G}, \mathrm{G_e}) = \frac{(1 + m_0) \times (1 – m_1)}{(2 + m_0 – m_1)} \tag{16}$$

B. Evaluation results

a) Qualitative studies.: We give the explanations of graphs in GraphSST2 in Fig. 2 and BA2Motifs in Fig. 3 and compare them qualitatively [7], [9].

For GraphSST2, we present explanations for both a positive (lower) and a negative (upper) graph with high and comparable sparsity. For both examples, FACExplainer captures all relevant words indicated by humans without including any extraneous ones. In contrast, the baseline methods capture some but not all sentiment words. SubgraphX outperforms

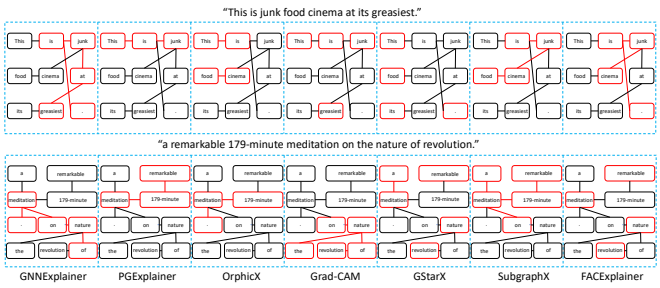

Fig. 2. Explanations for graphs selected from GraphSST2. We show the explanations of one positive sentence (lower) and one negative sentence (upper). Red outlines indicate the selected nodes/edges as the explanation. FACExplainer identifies the sentiment words more accurately compared to baselines.

other baselines, capturing words such as "remarkable" and "junk," but also includes some irrelevant words. Due to graph rules, SubgraphX can only select a connected subgraph as the explanation, leading to the inclusion of extraneous words like "179-minute" and "a."

For BA2Motifs, we visualize explanations selected with high and comparable Sparsity of examples from "house" and five-node cycle labeled graphs in Fig. 3. We chose to visualize BA2Motifs as it is a synthetic dataset and has ground truth about its data. Although the ground truth for datasets is not the same as for trained GNN models due to suboptimal training, the explanation is expected to be close to it. In both cases, FACExplainer precisely captures the label-determining motifs, while other baselines are unsuccessful.

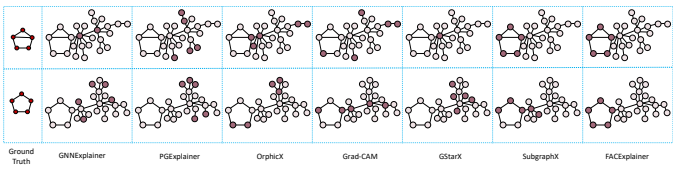

Fig. 3. Explanations for synthetic graphs selected from the BA2Motifs dataset. The leftmost column includes drawings of the "house" motif and the five-node cycle motif. We present the explanation for one graph labeled with "house" (upper) and one labeled with a five-node cycle (lower). The explanations are drawn in deep color. FACExplainer outperforms the baselines by more accurately identifying the label-determining motifs.

b) Quantitative studies.: Visualization results are only partially trustable due to the lack of ground truths [20] and suboptimal training. So, as shown in Table I we present the best H-Fidelity for each explainer from eight different runs, with sparsity ranging from 0.5-0.85 in increments of 0.05. Notice that the sparsity cannot be precisely guaranteed. FACExplainer outperforms others on 6/7 datasets and has the highest average. Following [7], [9], we also plot the curves of Fidelity scores, 1-Inv-Fidelity, and H-Fidelity with respect to the Sparsity scores in Fig. 4. FACExplainer outperforms others on 15/21 settings for eight different sparsities.

C. Ablation study and analysis

In this section, we evaluate the model-agnostic and computational efficiency of our proposed method by following [9]. Additionally, we present additional ablation studies to demonstrate the significant impact of GraphGrad-CAM on guiding the explanation generation process.

a) Model-agnostic explanation.: FACExplainer assumes only that the target GNN uses the aggregation operation, making it applicable to various GNN backbones. To further evaluate its performance, we conducted experiments on two additional popular GNN models: GIN [14] and GAT [28]. We adopted the training protocol of [7], [9] to train GIN on MUTAG and GAT on GraphSST2, and the experimental results are presented in Table II. Our findings indicate that FACExplainer outperforms the baselines for GAT on GraphSST2. As for GIN on MUTAG, although SubgraphX exhibits slightly better performance than FACExplainer, our method still delivers competitive results.

b) Efficiency study.: The FACExplainer algorithm scales in $\mathcal{O}(|V|)$. Following [7], [9], we study the empirical efficiency of our method and baselines by explaining the whole test set from BBBP and Redditbinary separately. Redditbinary, a real social networking dataset, has an average of 429.6 nodes in each graph, making it suitable for evaluating the explainer's efficiency on a large dataset. The average run times are reported in Table III. Results on BBBP for the baselines are similar to those in [7], [9]. For the BBBP dataset, FACExplainer is about 94 times faster than GStarX and 192 times faster than SubgraphX, respectively. For Redditbinary with large graphs, FACExplainer is about 5 times faster than GStarX and 110 times faster than SubgraphX, respectively. Although FACExplainer is not the fastest method, considering FACExplainer generates higher quality explanations than the baselines, the time complexity of our method is acceptable compared to other faster baselines.

c) Effectiveness study.: We conducted ablation studies to demonstrate the importance of employing GraphGrad-CAM as a guidance for the proposed FACExplainer framework. As shown in Table IV, we conducted experiments by removing the GraphGrad-CAM guidance and directly selecting specific nodes in the front as V_{crg}. For all datasets with GCN, FACExplainer was observed to be more powerful than the modified version without GraphGrad-CAM guidance, denoted as FACExplainer w/o GraphGrad-CAM. This result indicates that the significant influence of GraphGrad-CAM.

VI. Conclusion and Future Work

This paper revisits the process of generating graph explanations from the standpoint of the spatial information flowing within GNNs and proposes FACExplainer, a novel post-hoc explainer for GNNs. We first identify the model-unfaithful shortcomings of leading GNN explainers using graph rules. Then, we propose FACExplainer to search for a compact subgraph that is necessary and sufficient for the model's prediction. In FACExplainer, spatial information of graph convolutional layers is correctly utilized to find a much

TABLE I
THE BEST H-FIDELITY (HIGHER IS BETTER) OF EIGHT DIFFERENT SPARSITY FOR EACH DATASET. FACEXPLAINER SHOWS HIGHER H-FIDELITY ON AVERAGE AND ON 6/7 DATASETS.

Dataset	GNNExplainer	PGExplainer	Grad-CAM	OrphicX	GStarX	SubgraphX	FACExplainer
BA2Motifs	0.5005	0.4934	0.5201	0.5190	0.5566	0.5905	**0.6172**
BACE	0.4968	0.5292	0.4932	0.4978	0.5314	0.5596	**0.5979**
BBBP	0.4795	0.4838	0.4944	0.4854	0.5593	0.5647	**0.5722**
GraphSST2	0.4852	0.4935	0.4898	0.4962	0.544	0.5505	**0.5616**
MUTAG	0.4964	0.5034	0.5073	0.4985	**0.5496**	0.5387	0.5373
Twitter	0.4813	0.4987	0.4999	0.4917	0.5664	0.5566	**0.5687**
Redditbinary	0.5351	0.5145	0.5478	–	0.5689	0.5724	**0.5938**
Average	0.4964	0.5024	0.5075	0.4981	0.5537	0.5619	**0.5784**

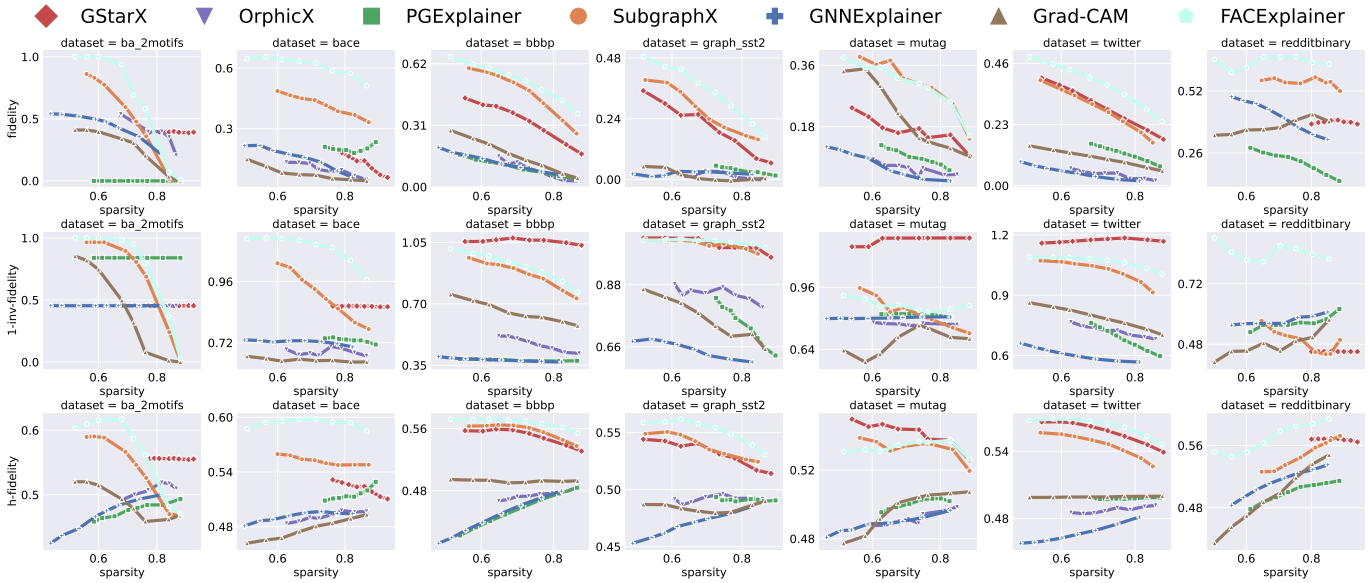

Fig. 4. Fidelity (row1), 1−Inv-Fidelity (row2), and H-Fidelity (row3) vs. Sparsity on all datasets corresponding to the results shown in Table I. All three metrics are the higher the better.

TABLE II
THE BEST H-FIDELITY OF EACH METHOD FOR GAT ON GRAPHSST2 AND GIN ON MUTAG.

Dataset	GNNExplainer	PGExplainer	Grad-CAM	OrphicX	GStarX	SubgraphX	FACExplainer
GraphSST2	0.4921	0.4958	0.4867	0.5007	0.5469	0.5533	**0.5680**
MUTAG	0.4955	0.4965	0.5106	0.5003	0.5033	**0.5558**	0.5428

tighter explanation searching space. FACExplainer is superior to strong baselines on synthetic, social networks, chemical, and text graphs classification tasks. However, similar to other CAM-based methods, the FACExplainer may result in local optima for its approximation strategy. In the future, we will explore ways to achieve better local and even global optima within an acceptable time complexity.

ACKNOWLEDGMENT

This work was funded in part by the National Key Research and Development Program of China under number 2019YFA0706200, in part by the National Natural Science Foundation of China grant under number 62076102, 62222603, and 92267203, in part by the Guangdong Natural Science Funds for Distinguished Young Scholar under number 2020B1515020041, and in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (2019ZT08X214).

REFERENCES

[1] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting," in *International Conference on Machine Learning*. PMLR, 2022, pp. 11 906–11 917.

[2] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec, "Graph convolutional policy network for goal-directed molecular graph generation," *Advances in neural information processing systems*, vol. 31, 2018.

[3] S. Bang, J. H. Jhee, and H. Shin, "Polypharmacy side-effect prediction with enhanced interpretability based on graph feature attention network," *Bioinformatics*, vol. 37, no. 18, pp. 2955–2962, 2021.

[4] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

TABLE III
AVERAGE RUNNING TIME ON TEST SET OF BBBP OR REDDITBINARY, RESPECTIVELY. THE TRAINING TIME THAT PGEXPLAINER AND ORPHICX NEEDS ARE REPORTED IN BRACKETS.

Dataset	GNNExplainer	PGExplainer	Grad-CAM	OrphicX	GStarX	SubgraphX	FACExplainer
BBBP	16.27	0.03(329.5)	0.02	0.02(350)	38.62	78.90	0.41
Redditbinary	16.29	0.03(357.6)	0.07	–	485.29	11367.67	103.11

TABLE IV
THE BEST H-FIDELITY (HIGHER IS BETTER) OF 8 DIFFERENT SPARSITY FOR FACEXPLAINER AND FACEXPLAINER W/O GRAPHGRAD-CAM

Dataset	FACExplainer w/o GraphGrad-CAM	FACExplainer
BA2Motifs	0.598	**0.6172**
BACE	0.5667	**0.5979**
BBBP	0.5654	**0.5722**
GraphSST2	0.5447	**0.5616**
MUTAG	0.5361	**0.5373**
Twitter	0.5451	**0.5687**
Redditbinary	0.5870	**0.5938**
Average	0.5633	**0.5784**

[5] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, "Parameterized explainer for graph neural network," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 19 620–19 631.

[6] W. Lin, H. Lan, and B. Li, "Generative causal explanations for graph neural networks," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 6666–6679.

[7] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, "On explainability of graph neural networks via subgraph explorations," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 12 241–12 252.

[8] W. Lin, H. Lan, H. Wang, and B. Li, "Orphicx: A causality-inspired latent variable model for interpreting graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 13 729–13 738.

[9] S. Zhang, Y. Liu, N. Shah, and Y. Sun, "Gstarx: Explaining graph neural networks with structure-aware cooperative games," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 19 810–19 823.

[10] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability methods for graph convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 772–10 781.

[11] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[12] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[13] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 24–25.

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

[15] A. Wijesinghe and Q. Wang, "A new perspective on" how graph neural networks go beyond weisfeiler-lehman?"," in *International Conference on Learning Representations*, 2022.

[16] A. Leman and B. Weisfeiler, "A reduction of a graph to a canonical form and an algebra arising during this reduction," *Nauchno-Technicheskaya Informatsiya*, vol. 2, no. 9, pp. 12–16, 1968.

[17] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan, "L-shapley and c-shapley: Efficient model interpretation for structured data," *arXiv preprint arXiv:1808.02610*, 2018.

[18] H. W. Kuhn and A. W. Tucker, *Contributions to the Theory of Games*. Princeton University Press, 1953, no. 28.

[19] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[20] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," *arXiv preprint arXiv:2012.15445*, 2020.

[21] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[22] S. Miao, M. Liu, and P. Li, "Interpretable and generalizable graph learning via stochastic attention mechanism," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 15 524–15 543.

[23] S. Suresh, P. Li, C. Hao, and J. Neville, "Adversarial graph augmentation to improve graph contrastive learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 920–15 933, 2021.

[24] K. Amara, R. Ying, Z. Zhang, Z. Han, Y. Shan, U. Brandes, S. Schemm, and C. Zhang, "Graphframex: Towards systematic evaluation of explainability methods for graph neural networks," *arXiv preprint arXiv:2206.09677*, 2022.

[25] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1365–1374.

[26] A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity," *Journal of medicinal chemistry*, vol. 34, no. 2, pp. 786–797, 1991.

[27] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "Moleculenet: a benchmark for molecular machine learning," *Chemical science*, vol. 9, no. 2, pp. 513–530, 2018.

[28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.