Are Graph Neural Network Explainers Robust to Graph Noises?

Yiqiao Li[1](✉), Sunny Verma[1], Shuiqiao Yang[2], Jianlong Zhou[1], and Fang Chen[1]

[1] University of Technology Sydney, Sydney, Australia
Yiqiao.Li-1@student.uts.edu.au,
{sunny.verma,jianlong.zhou,fang.chen}@uts.edu.au
[2] University of New South Wales, Sydney, Australia
shuiqiao.yang@unsw.edu.au

Abstract. With the rapid deployment of graph neural networks (GNNs) based techniques in a wide range of applications such as link prediction, community detection, and node classification, the explainability of GNNs become an indispensable component for predictive and trustworthy decision making. To achieve this goal, some recent works focus on designing explainable GNN models such as GNNExplainer, PGExplainer, and Gem. These GNN explainers have shown remarkable performance in explaining the predictive results from GNNs. Despite their success, the robustness of these explainers is less explored in terms of vulnerabilities of GNN explainers. Graph perturbations such as adversarial attacks can lead to inaccurate explanations and consequently cause catastrophes. Thus, in this paper, we take the first step and strive to explore the robustness of GNN explainers. To be specific, we first define two adversarial attack scenarios—*aggressive adversary* and *conservative adversary* to contaminate graph structures. We then investigate the impacts of the poisoned graphs on the explainability of three prevalent GNN explainers with three standard evaluation metrics: $Fidelity^+$, $Fidelity^-$, and $Sparsity$. We conduct experiments on synthetic and real-world datasets and focus on two popular graph mining tasks: node classification and graph classification. Our empirical results suggest that GNN explainers are generally not robust to the adversarial attacks caused by graph structural noises.

Keywords: Graph neural networks · GNN explainers · Adversarial attacks · Robustness

1 Introduction

Generally, a computation graph $\boldsymbol{G}$ can be represented as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{A}, \boldsymbol{X})$, where $\boldsymbol{V}$ is the node set, $\boldsymbol{A} \in \{0, 1\}$ denotes the adjacency matrix that $\boldsymbol{A}_{ij} = 1$ if there is an edge between node i and node j, otherwise $\boldsymbol{A}_{ij} = 0$, and $\boldsymbol{X}$ indicates the feature matrix of the graph $\boldsymbol{G}$. It is an ideal data structure for

H. Aziz et al. (Eds.): AI 2022, LNAI 13728, pp. 161–174, 2022.
https://doi.org/10.1007/978-3-031-22695-3_12

a variety of real-world datasets, such as chemical compounds [3], social circles [21], and road networks [15]. Graph neural networks (GNNs) [5,26,29,33], with the resurgence of deep learning, have become a powerful tool to model these graph datasets and achieved impressive performance. However, a GNN model is typically very complicated and how it makes predictions is unclear; while unboxing the working mechanism of a GNN model is crucial in many practical applications (e.g., criminal associations predicting [24], traffic forecasting [11], and medical diagnosis [1,23]).

Recently, several explainers [19,20,30] have been proposed to tackle the problem of explaining GNN models. These attempts can be categorized into *local* and *global* explainers according to their interpretation scales. In particular, if the method provides an explanation only for a specific instance, it is a *local explainer*. In contrast, if the method explains the whole model, then it is a *global explainer*. Alternatively, GNN explainers can also be classified as either *transductive* or *inductive* explainers based on their capacity to generalize to extra unexplained nodes. We investigate a flurry of recent GNN explainers and decide to use three most representative GNN explainers—GNNExplainer [30], PGExplainer [20], and Gem [19]—in our experiments. GNNExplainer is challenging to be applied into inductive settings as its explanations are limited to a single instance and it merely provides local explanations; while a trained PGExplainer which constructs global explanations and Gem which generates both local and global explanations can be used in inductive scenarios to infer explanations for unexplained instances without the need of retraining the explanation models. Table 1 summarizes the characteristics of these methods.

Table 1. The characteristics of GNN explainers.

	GNNExplainer	PGExplainer	Gem
Interpretation scale	Local explainer	Global explainer	Local & global explainer
Transduction/ Induction	Transductive explainer	Inductive explainer	Inductive explainer
Applications	Node classification Graph classification Link prediction	Node classification Graph classification	Node classification Graph classification

On the other hand, *robustness* is also an important topic in the community of deep learning and has gained significant attention over years. Recently, there are a large number of research studies focusing on the robustness of image classification including adversarial robustness [27] and non-adversarial robustness [10,16]. In addition, researchers start to explore the robustness of GNN models in recent years, having gained several crucial observations and insights [2,34]. Nevertheless, the robustness of GNN explainers is still under exploration. While in real world, graph datasets are never ideal and often contaminated by various nuisance factors such as noises in node features and/or in graph structures. Therefore, one natural question one might ask: *are current GNN explainers robust against these nuisance factors?*

To answer this question, we in this paper take the first step to examine the robustness of GNN explainers. To be specific, we explore two adversary scenarios to contaminate graph datasets:

- *Aggressive adversary.* We introduce noises to graph structures without considering the characteristics of nodes–whether it is an important node or a redundant node. To be more specific, we may pollute any nodes to have edges with others regardless of the impact on the GNN models.
- *Conservative adversary.* In contrast to *aggressive adversary*, we introduce noises to graph datasets in a more cautious way such that we hope the injected noises would not affect the GNN model itself. To achieve this goal, we have to take the characteristics of graph dataset itself into account (e.g., whether the node is an important node or an unimportant node). We then only alter the graph structure by adding edges among unimportant nodes. By doing so, the underlying essential subgraph, which determines the prediction of GNN models, is untouched.

We first use the aforementioned adversary scenarios to contaminate the graph datasets. We then use these generated noisy graph datasets to evaluate the robustness of the GNN explainers. For the baseline, we refer to the performance of the GNN explainers on original (clean) graph datasets. Thus, we track and compare the difference in the performance of GNN explainers between original and polluted graph datasets. Our contributions can be summarized as followings:

- For the sake of comprehensive evaluations, we propose to generate noisy graph data under two scenarios—*aggressive* adversary and *conservative* adversary.
- We empirically investigate the robustness of GNN explainers against these perturbations through two different applications including node classification and graph classification.
- We find that GNN explainers in general are not robust to these perturbations, implying that robustness is another essential factor one should take into account when evaluating GNN explainers.

2 Related Work

2.1 GNNs and the Robustness of GNNs

Graph neural networks (GNNs) have shown their effectiveness and obtained the state-of-the-art performance on many different graph tasks, such as node classification, graph classification, and link prediction. Since graph data widely exist in different real-world applications, such as social networks [25], chemistry [8], and biology [6], GNNs are becoming increasingly important and useful.

Despite their great performance, GNNs share the same drawback as other deep learning models; that is, they are usually treated as black-boxes and lack human-intelligible explanations. Without understanding and verifying the inner working mechanisms, GNNs cannot be fully trusted, which prevents their use in critical applications pertaining to fairness, privacy, and safety [4].

On the other hand, the robustness evaluation for GNNs has received a great deal of attention recently. In recent years, some adversarial attacks and backdoor attacks against GNNs are proposed [7,9,28,34]. Specially, in [28], Yang et al. propose a transferable trigger to launch backdoor attack against different GNNs. In [34], authors propose an efficient algorithm NETTACK exploiting incremental computations. They concentrate on adversarial perturbations that target the node's characteristics and the graph structure, therefore taking into account the interdependencies between instances. In addition, they ensure that the perturbations are undetectable by keeping essential data features. Ghorbani et al. [9] demonstrate how to generate adversarial perturbations that produce perceptively indistinguishable inputs that are assigned the same predicted label, yet have very different interpretations. They prove that systematic perturbations can result in drastically different interpretations without modifying the label. Fox et al. [7] investigate that GNNs are not robust to structural noise. They focus on inserting addition of random edges as noise in the node classification without distinguish important and unimportant nodes. On the contrast, we focus on injecting conservative structure noise into unimportant nodes/subgraphs. Overall, in our research, we propose to infuse aggressive and conservative structure noise individually into graph data in order to examine the robustness of GNN explainers.

2.2 GNN Explainers

GNNs incorporate both graph structure and feature information, which results in complex non-linear models, rendering explaining its prediction remain a challenging task. Besides, models explanations could bring a lot of benefits to users (e.g., improving safety and promoting fairness). Thus, some popular works has emerged in recent years focusing on the explanation of GNN models by leveraging the properties of graph features and structures. There are some popular GNN explainers developing explaining strategies based on graph intrinsic structures and features. We will briefly review three different GNN explainers: GNNExplainer, PGExplainer, and Gem.

GNNExplainer [30] is a seminal method in the field of explaining GNN models. It provides local explanations for GNNs by identifying the most relevant features and subgraphs, which are essential in the prediction of a GNN. *PGExplainer* [20] introduces explanations for GNNs with the use of a probabilistic graph. It provides model-level explanations for each instance and possesses strong generalizability. *Gem* [19] is able to provide both local and global explanations and it is also operated in an inductive setting. Thus, it can explain GNN models without retraining. Particularly, it adopts a parameterized graph auto-encoder with Graph Convolutional Network(GCN) [14] layers to generate explanations.

3 Method

In this paper, we examine the robustness of GNN explainers under two adversary scenarios—*aggressive adversary* and *conservative adversary*. In this section, we provide the details of our method. Particularly, we first introduce how we inject noises into graph data and construct noisy graph data (see Sect. 3.1), and we then depict our evaluation flow (see Sect. 3.2).

3.1 Adversary Generation

Without loss of generality, we consider generating aggressive and conservative adversaries in a graph classification task. For a graph $\boldsymbol{G}_i = (\boldsymbol{V}_i, \boldsymbol{A}_i, \boldsymbol{X}_i)$ with label $\boldsymbol{L}_i$, we have the prediction $f(\boldsymbol{G}_i)$ of a GNN model, and the explanation $E(f(\boldsymbol{G}_i), \boldsymbol{G}_i)$ from a GNN explainer.

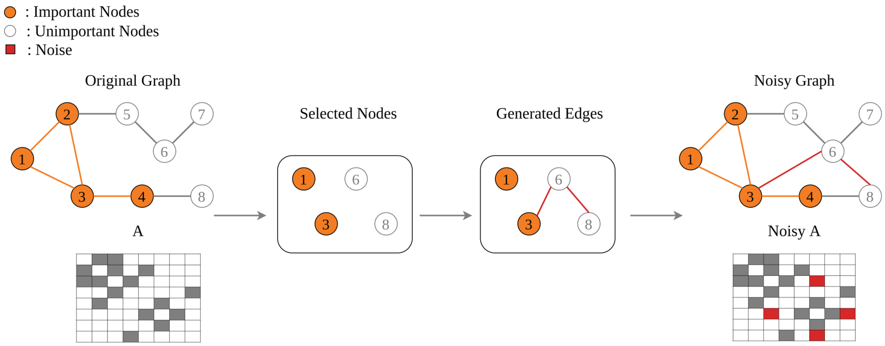

Fig. 1. The instance of generating aggressive structure noise. The orange nodes denote important nodes, while the rest means unimportant nodes in the graph. In this scenario, we do not take the node property into account and we randomly select nodes. (Color figure online)

Aggressive Adversary Generation. The *aggressive adversary* disregards the role of nodes and radically incorporates structure noises into nodes without considering their impacts on the GNN models. For a particular graph $\boldsymbol{G}_i$, we randomly choose $\varepsilon = \{10\%, 30\%, 50\%, 80\%\}$ nodes from the set $\boldsymbol{V}_i$, then generate edges among these selected nodes by using random graph generation model with generating edges probability 0.1, meaning that the number of edges is equal to 10% of the number of selected nodes. Figure 1 shows a toy example of *aggressive adversary* generation. After generating aggressive structure noises, we obtain a new noisy graph $\widehat{\boldsymbol{G}_i} = (\boldsymbol{V}_i, \widehat{\boldsymbol{A}_i}, \boldsymbol{X}_i)$ with label $\boldsymbol{L}_i$, and further obtain the GNN prediction $f(\widehat{\boldsymbol{G}_i})$ on this new noisy graph as well as its the explanation $E(f(\widehat{\boldsymbol{G}_i}), \widehat{\boldsymbol{G}_i})$. As we have aggressively changed the structure of the graph, the probability of $f(\widehat{\boldsymbol{G}_i})$ is expected to be lower, implying that the aggressive structure noises also affect the performance of the GNN models. Furthermore, predictions of GNN model is another input to GNN explainers, which is another factor to influence explanations of GNN explainers.

Conservative Adversary Generation. The *conservative adversary* selectively appends structure noise into unimportant nodes. Particularly, in *conservative adversary*, we build a structure noise which would not alter the prediction of GNN models. For a particular graph $\boldsymbol{G}_i$, we obtain the unimportant nodes set $\boldsymbol{N}_i$ with the similar ratio of $\varepsilon = \{10\%, 30\%, 50\%, 80\%\}$ we used in the setting of *aggressive adversary*. Then, we use random graph generation model to generate edges among $\boldsymbol{N}_i$ with the generating edges probability 0.1. Similarly, Fig. 2 shows a toy example of *conservative adversary* generation. After developing conservative structure noise, we get a noisy graph $\boldsymbol{G}_i^{'} = (\boldsymbol{V}_i, \boldsymbol{A}_i^{'}, \boldsymbol{X}_i)$ with label $\boldsymbol{L}_i$. Therefore, we are able to obtain the GNN prediction $f(\boldsymbol{G}_i^{'})$ and the explanation $E(f(\boldsymbol{G}_i^{'}), \boldsymbol{G}_i^{'})$. In *conservative adversary*, since the significant subgraph that determines the prediction of GNN models is unmodified, there is a high possibility that $f(\boldsymbol{G}_i^{'})$ would make the *correct* predictions. Thus, the prediction of GNN as a parameter in GNN explainers inputs keeps stable and unchanged. Therefore, one should expect that the GNN explainers would be more robust against conservative adversary than aggressive adversary.

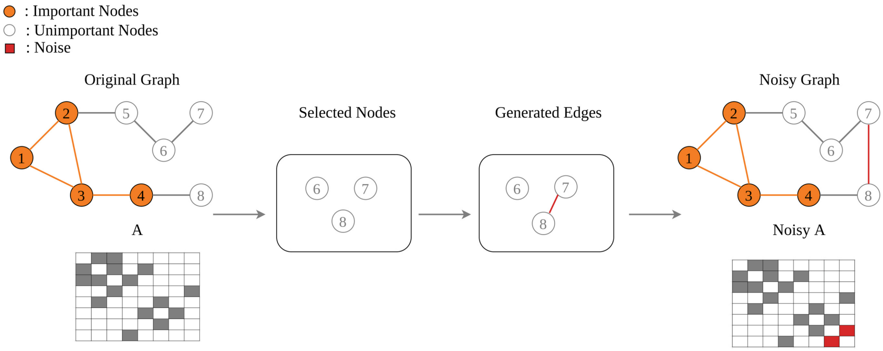

Fig. 2. The instance of generating conservative structure noise. The orange nodes denote important nodes, while the rest are unimportant nodes in the graph. We only select unimportant nodes. (Color figure online)

3.2 Robustness Evaluation Framework

For a GNN model, GNN explainers are used to unveil why the GNN model makes its predictions. Thus, it is intriguing to explore whether these explanations really make sense, especially when the graph data is not clean and polluted by noises, which is often the case in real-world datasets. The contamination can occur in many ways such as during the process of data collection, the defects of sensors, data transmission through network, and many others. In this paper, we insert noises into the original clean graph data to examine whether the explanation of GNN explainers would be affected.

Specifically, in our experiments, we target to investigate the robustness of the GNN explainer to structure noises. We introduce two types of structure noises to graph datasets, of which the detailed information can be found in Sect. 3.1. After obtaining noisy graph dataset, we feed it into a pre-trained GNN that is trained by the original clean graph dataset and get its corresponding predictions. Then a GNN explainer conducts its explanations and we obtain its explanation performance and further conduct comparisons with the explanations on the original graph dataset. The pipeline of our robustness evaluation method is shown in Fig. 3. We further show an example of our experimental flow under the *conservative adversary* in Fig. 4.

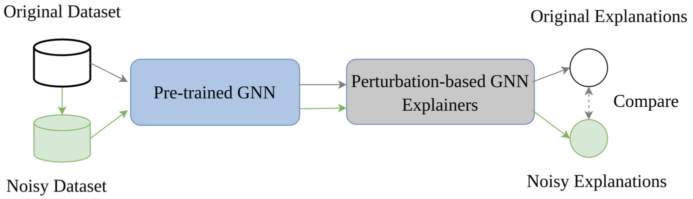

Fig. 3. In this diagram, different lines denote distinct flows. The black lines denote initial flow that generates explanations for the original dataset. The green lines denote flow that generates a noisy graph data from the original graph data as well as its explanations. Finally, we can compare "noisy" explanations with "original" explanations. (Color figure online)

Furthermore, we use accuracy to quantitatively measure the influence of structure noises to the GNN model. We assume that the performance of GNN model would rarely be affected if the prediction accuracy on the noisy graph dataset is roughly the same as the accuracy on the original clean graph dataset. We further assume that if the GNN model itself is not confused by the injected noises, then the GNN explainers would yield similar explanations between original clean graph data and noisy graph data.

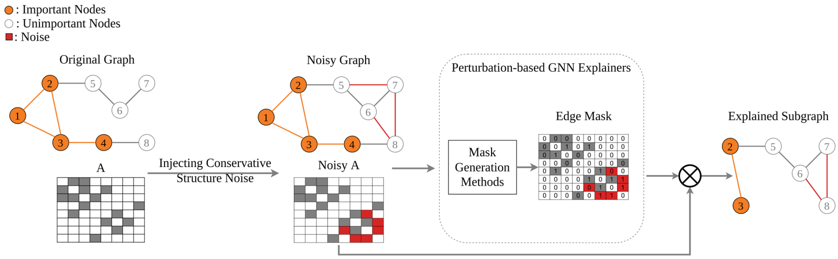

Fig. 4. The instance of generating explanation for noisy graph with *conservative adversary*. The orange nodes denote important nodes, while the rest means unimportant nodes in the graph. The orange nodes and edges are expected to be as an explanation from GNN explainers. However, after injecting structure noise which is highlighted in red colour, the GNN explainers can not get the true important subgraph, which demonstrates that the GNN explainers are not robust to structure noises. (Color figure online)

4 Experiments

In this section, we conduct experiments to inspect the robustness of GNN explainers against structure noises. We first describe the details of the implementation, datasets, and metrics we used in Sect. 4.1. After that, we present and analyze the experimental results for *aggressive adversary scenario* and *conservative adversary scenario* in Sect. 4.2 and Sect. 4.3, respectively.

4.1 Implementation Details, Datasets, and Metrics

Implementation Details. In this paper we choose GCN as the classification classifier. For GNN explainers, we choose GNNExplainer [30], PGExplainer [20], and Gem [19]. In order to obtain the pre-trained GCN models, we split the datasets into percentages of 80/10/10 as the training, validation, and test set, respectively. We follow the experimental settings in Gem [19]. Specifically, we firstly train a three-layer GCN model based on BA-Shapes dataset, Tree-Cycles dataset, and Mutagenicity dataset, respectively. We choose Adam [13] as the optimizer. After that, we utilize the pre-trained GCN models and the explainers to obtain the explanations for both the original clean graph datasets and the noisy graph datasets. Furthermore, by analyzing the experiment settings and results in [19], we note that explainers obtain different levels of accuracy when selecting different top-important edges as explaining edges. Therefore, one should choose an appropriate number of top important edges when evaluating explainers. In our paper, we select top 6 edges for synthetic datasets (BA-Shapes and Tree-Cycles) and top 15 edges for Mutagenicity dataset.

Datasets. We focus on two widely used node classification datasets, including BA-Shapes and Tree-Cycles [18,31], and one graph classification dataset, Mutagenicity [12]. Statistics of these datasets are shown in Table 2. For BA-Shapes and Tree-Cycles datasets the nodes which define a motif structure such as a house or cycle are considered as important nodes. For Mutagenicity datasets, Carbon rings with chemical groups NH_2 or NO_2 are known to be mutagenic. Carbon rings however exist in both mutagen and nonmutagenic graphs, which are not discriminative. Thus, we simply treat carbon rings as the shared base graphs and NH_2, NO_2 as important subgraphs for the mutagen graphs.

Table 2. Dataset information.

	Node classification		Graph classification
	BA-shapes	Tree-cycles	Mutagenicity
# of Graphs	1	1	4,337
# of Edges	4110	1950	266,894
# of Nodes	700	871	131,488
# of Labels	4	2	2

In addition, explainers—GNNExplainer, PGExplainer, and Gem—can obtain higher accuracy when used to explain only important nodes or subgraphs. While in our experiments, we may alter the nodes as well as the subgraph structures, thus we have to explain all nodes or subgraphs (important or unimportant), which may lead to suboptimal accuracy. However, this is not a major issue for us as our goal in this paper is to compare the performance change of GNN explainers on graph datasets before and after adding noises.

Noisy Datasets. Following the noise generation pipeline described in Sect. 3, we inject *aggressive* and *conservative* structure noises into these graph datastes to generate *aggressive* and *conservative* noisy datasets, respectively. For *conservative* structure noisy datasets, we only inject noises into unimportant nodes to minimize the affection of structure noise on GNN prediction. By doing so, we attempt to maintain GNN predictions on *conservative* structure noise datasets.

Metrics. Good metrics should evaluate whether the explanations are faithful to the model. After comparing the characteristic of each quantitative metric [17,32], we chose *Fidelity*$^+$ [31], *Fidelity*$^-$ [31], and *Sparsity* [22] as our evaluation metrics. The *Fidelity*$^+$ metric indicates the difference of predicted probability between the original predictions and the new prediction after removing important input features. In contrast, the metric *Fidelity*$^-$ represents prediction changes by keeping important input features and removing unimportant structures. Besides, *Sparsity* measures the fraction of features selected as important by explanation methods. The *Fidelity*$^+$, *Fidelity*$^-$, and *Sparsity* can be defined as:

$$Fidelity^{+} = \frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{G}_i)_{y_i} - f(\boldsymbol{G}_i^{1-m_i})_{y_i}), \tag{1}$$

$$Fidelity^{-} = \frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{G}_i)_{y_i} - f(\boldsymbol{G}_i^{m_i})_{y_i}), \tag{2}$$

$$Sparsity = \frac{1}{N}\sum_{i=1}^{N}(1 - \frac{|\boldsymbol{s}_i|}{|\boldsymbol{S}_i|_{total}}), \tag{3}$$

where N is the total number of samples and y_i is the class label. $f(\boldsymbol{G}_i)_{y_i}$ and $f(\boldsymbol{G}_i^{1-m_i})_{y_i}$ are the prediction probabilities of y_i when using the original graph $\boldsymbol{G}_i$ and the occluded graph $\boldsymbol{G}_i^{1-m_i}$, which is gained by occluding important features found by explainers from the original graph. Thus, a *higher Fidelity*$^+$ (↑) is desired. $f(\boldsymbol{G}_i^{m_i})_{y_i}$ is the prediction probabilities of y_i when using the explanation graph $\boldsymbol{G}_i^{m_i}$, which is obtained by important structures found by explainable methods. Thus a *lower Fidelity*$^-$ (↓) is desired. Furthermore, the $|S_i|_{total}$ represents the total number of features (e.g., nodes, nodes features, or edges) in the original graph model; while $|s_i|$ is the size of important features/nodes found by the explainable methods and it is a subset of $|S_i|$. Note that higher sparsity values indicate that explanations are sparser and likely to capture only the most essential input information. Hence, a *higher Sparsity* (↑) is desired.

4.2 Vulnerable to Aggressive Adversary

To measure the robustness of GNN explainers against aggressive structure noises, we estimate the differences in performance of GNN explainers between original and aggressive noisy datasets. We first obtain the explanation performance of each explainers on original clean graph datasets, which serves as our baseline. We then obtain the corresponding explanation performance of each explainers on noisy graph datasets with aggressive adversary. For reference, we also report the GCN accuracy.

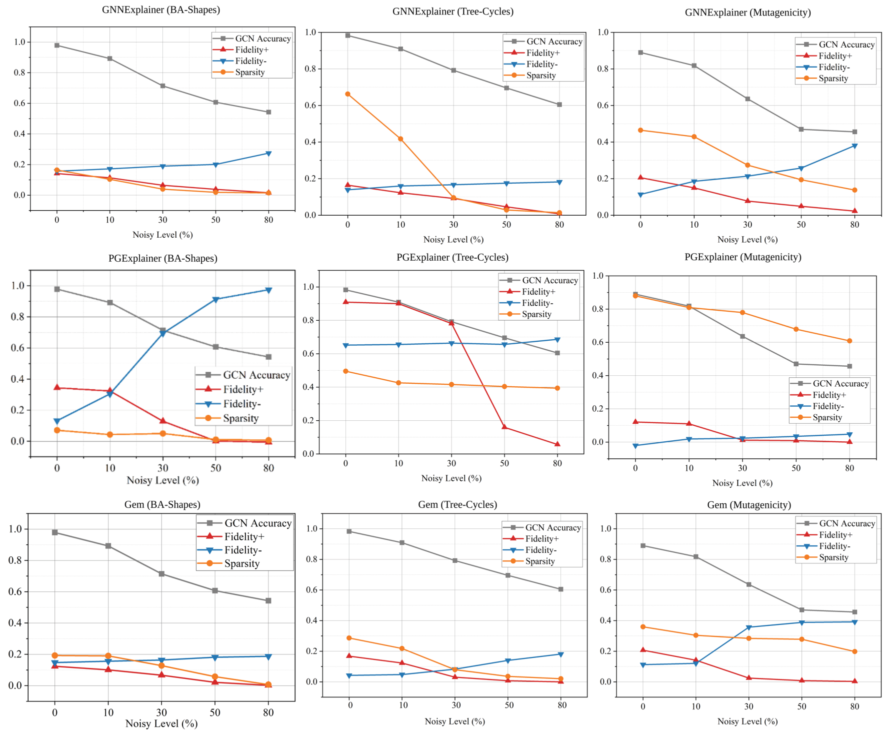

Fig. 5. The results of aggressive adversary in terms of $Fidelity^{+}$, $Fidelity^{-}$, and $Sparsity$.

GNN Explainers Are Not Robust to Aggressive Adversary. Figure 5 shows the results of the robustness of GNN explainers against aggressive noise. One can observe that: 1) As the noise level increases, all explanation performance metrics including $Fidelity^{+}$, $Fidelity^{-}$, and $Sparsity$ consistently become worse, implying that *aggressive* noises do have negative impacts on the GNN explainers; 2) The accuracy of GCN keeps decreasing as the noise level increases, implying that the aggressively injected noises also affect the performance of GCN itself, which is consistent with the findings in [7,34]; 3) The findings mentioned above are consistent across different datasets and different tasks, suggesting the generality of our findings.

4.3 Vulnerable to Conservative Adversary

Now, we start to explore how conservative adversary affect the GNN explainers. We follow the exact pipeline in Sect. 4.2 expect that we here inject noises in a more cautious way. We believe this conservative adversary would yield negligible impacts on the GCN itself while it may still negatively affect the explanation quality of GNN explainers (see Sect. 3 for more details).

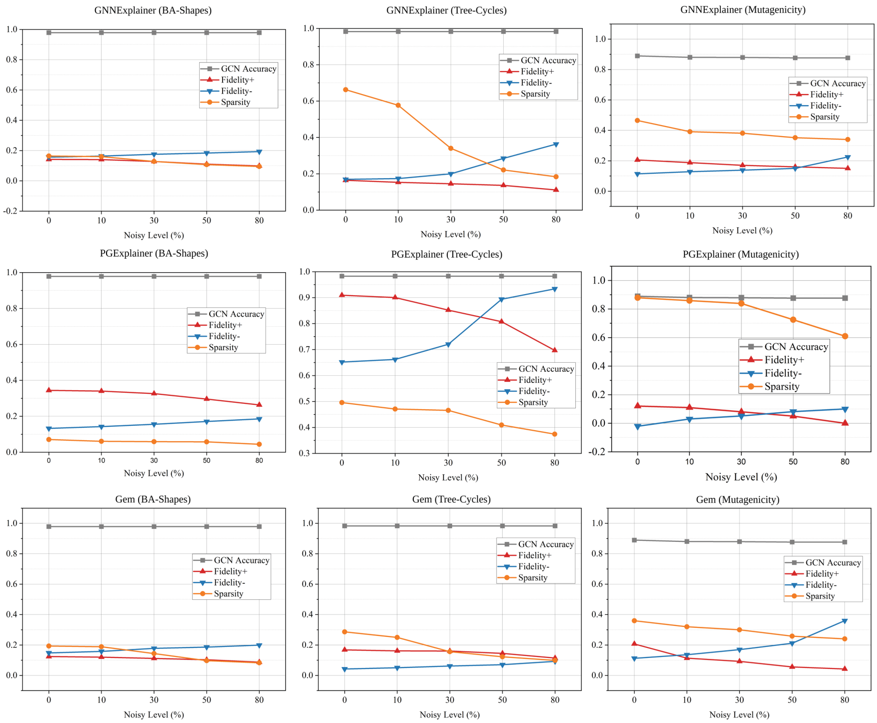

Fig. 6. The results of conservative adversary in terms of $Fidelity^{+}$, $Fidelity^{-}$, and $Sparsity$.

GNN Explainers Are Not Robust to Conservative Adversary. Figure 6 shows the experimental results for the setting of conservative adversary. As expected, the accuracy of the GNN is quite stable and does not change much even when the noise level increases, implying that the noises injected in this way do not alter the essential structures of graph datasets. However, in term of $Fidelity^{+}$, $Fidelity^{-}$, and $Sparsity$, we see a similar trend as the aggressive adversary (Sect. 4.2) although the impacts here are much benign, which further demonstrates the fragility of GNN explainers to graph noises.

5 Conclusion

In this paper, we attempt to identify the robustness issue of GNN explainers. We propose two types of structure noises—*aggressive adversary* and *conservative adversary*—to construct noisy graphs. We evaluate three recent representative GNN explainers including GNNExplainer, PGExplainer, and Gem, which vary in terms of interpretation scales and generality. We conduct experiments on two different tasks—node classification with BA-Shapes and Tree-Cycles datasets and graph classification with Mutagenicity dataset. Through experiments, we find that the current GNN explainers are fragile to adversarial attacks as the quality of their explanations is significantly decreased across different severity of noises. Our findings suggest that robustness is a practical issue one should take into account when developing and deploying GNN explainers in real-world applications. In our future work, we would develop algorithms and models to improve the robustness of GNN explainers against these adversaries.

References

1. Chen, D., Zhao, H., He, J., Pan, Q., Zhao, W.: An causal XAI diagnostic model for breast cancer based on mammography reports. In: 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 3341–3349, December 2021. https://doi.org/10.1109/BIBM52615.2021.9669648
2. Dai, H., et al.: Adversarial attack on graph structured data. In: Proceedings of the 35th International Conference on Machine Learning, pp. 1115–1124. PMLR, July 2018
3. Debnath, A.K., Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. J. Med. Chem. **34(**2), 786–797 (1991). https://doi.org/10.1021/jm00106a046
4. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning (2017). https://doi.org/10.48550/ARXIV.1702.08608
5. Duan, W., Xuan, J., Qiao, M., Lu, J.: Learning from the dark: boosting graph convolutional neural networks with diverse negative samples. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 6, pp. 6550–6558 (2022). https://doi.org/10.1609/aaai.v36i6.20608
6. Fout, A., Byrd, J., Shariat, B., Ben-Hur, A.: Protein interface prediction using graph convolutional networks, pp. 6533–6542, December 2017
7. Fox, J., Rajamanickam, S.: How robust are graph neural networks to structural noise? (2019). https://doi.org/10.48550/ARXIV.1912.10206
8. Fung, V., Zhang, J., Juarez, E., Sumpter, B.G.: Benchmarking graph neural networks for materials chemistry. npj Comput. Mater. **7**(1), 1–8 (2021). https://doi.org/10.1038/s41524-021-00554-0
9. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile, vol. 33, pp. 3681–3688 (2019). https://doi.org/10.1609/aaai.v33i01.33013681
10. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: Proceedings of the International Conference on Learning Representations (2019)

11. Jiang, W., Luo, J.: Graph neural network for traffic forecasting: a survey. Expert Syst. Appl. **207**, 117921 (2022). https://doi.org/10.1016/j.eswa.2022.117921
12. Kazius, J., McGuire, R., Bursi, R.: Derivation and validation of toxicophores for mutagenicity prediction. J. Med. Chem. **48**(1), 312–320 (2005). https://doi.org/10.1021/jm040835a
13. Kingma, D.P., Ba, L.J.: Amsterdam machine learning lab (IVI, FNWI): adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR). arXiv.org (2015)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2017)
15. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. Internet Math. **6**(1), 29–123 (2009). https://doi.org/10.1080/15427951.2009.10129177
16. Li, T., Mehta, R., Qian, Z., Sun, J.: Rethink autoencoders: robust manifold learning. In: ICML Workshop on Uncertainty and Robustness in Deep Learning (2020)
17. Li, Y., Zhou, J., Verma, S., Chen, F.: A survey of explainable graph neural networks: taxonomy and evaluation metrics (2022). https://doi.org/10.48550/ARXIV.2207.12599
18. Lin, C., Sun, G.J., Bulusu, K.C., Dry, J.R., Hernandez, M.: Graph neural networks including sparse interpretability (2020). https://doi.org/10.48550/ARXIV.2007.00119
19. Lin, W., Lan, H., Li, B.: Generative causal explanations for graph neural networks. In: Proceedings of the 38th International Conference on Machine Learning, pp. 6666–6679. PMLR, July 2021
20. Luo, D., et al.: Parameterized explainer for graph neural network. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS 2020, pp. 19620–19631. Curran Associates Inc., Red Hook, December 2020
21. McAuley, J., Leskovec, J.: Learning to discover social circles in ego networks, pp. 539–547, December 2012
22. Pope, P.E., Kolouri, S., Rostami, M., Martin, C.E., Hoffmann, H.: Explainability methods for graph convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
23. Singh, A., Sengupta, S., Lakshminarayanan, V.: Explainable deep learning models in medical image analysis. J. Imaging **6**(6), 52 (2020). https://doi.org/10.3390/jimaging6060052
24. Wang, C., Lin, Z., Yang, X., Sun, J., Yue, M., Shahabi, C.: HAGEN: homophily-aware graph convolutional recurrent network for crime forecasting, vol. 36, pp. 4193–4200, June 2022. https://doi.org/10.1609/aaai.v36i4.20338
25. Wu, Y., Lian, D., Xu, Y., Wu, L., Chen, E.: Graph convolutional networks with Markov random field reasoning for social spammer detection, vol. 34, pp. 1054–1061, April 2020. https://doi.org/10.1609/aaai.v34i01.5455
26. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2021). https://doi.org/10.1109/TNNLS.2020.2978386
27. Xu, H., et al.: Adversarial attacks and defenses in images, graphs and text: a review. Int. J. Autom. Comput., 1–28 (2019). https://doi.org/10.1007/s11633-019-1211-x
28. Yang, S., et al.: Transferable graph backdoor attack. arXiv preprint arXiv:2207.00425 (2022)

29. Yang, S., et al.: Variational co-embedding learning for attributed network clustering. CoRR abs/2104.07295 (2021)
30. Ying, R., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: GNN explainer: a tool for post-hoc explanation of graph neural networks. CoRR abs/1903.03894 (2019)
31. Yuan, H., Yu, H., Gui, S., Ji, S.: Explainability in graph neural networks: a taxonomic survey. IEEE Trans. Pattern Anal. Mach. Intell., 1–19 (2022). https://doi.org/10.1109/TPAMI.2022.3204236
32. Zhou, J., Gandomi, A.H., Chen, F., Holzinger, A.: Evaluating the quality of machine learning explanations: a survey on methods and metrics. Electronics **10**(5), 593 (2021). https://doi.org/10.3390/electronics10050593
33. Zhou, J., et al.: Graph neural networks: a review of methods and applications. AI Open **1**, 57–81 (2020). https://doi.org/10.1016/j.aiopen.2021.01.001
34. Zügner, D., Akbarnejad, A., Günnemann, S.: Adversarial attacks on neural networks for graph data. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, pp. 2847–2856. Association for Computing Machinery, New York, July 2018. https://doi.org/10.1145/3219819.3220078