ACGAN-GNNExplainer: Auxiliary Conditional Generative Explainer for Graph Neural Networks

Yiqiao Li
Data Science Institute
University of Technology Sydney
Sydney, Australia
Yiqiao.Li-1@student.uts.edu.au

Jianlong Zhou
Data Science Institute
University of Technology Sydney
Sydney, Australia
Jianlong.Zhou@uts.edu.au

Yifei Dong
Data Science Institute
University of Technology Sydney
Sydney, Australia
Yifei.Dong@uts.edu.au

Niusha Shafiabady
Faculty of Science and Technology
Charles Darwin University
Sydney, Australia
Niusha.Shafiabady@cdu.edu.au

Fang Chen
Data Science Institute
University of Technology Sydney
Sydney, Australia
Fang.Chen@uts.edu.au

ABSTRACT

Graph neural networks (GNNs) have proven their efficacy in a variety of real-world applications, but their underlying mechanisms remain a mystery. To address this challenge and enable reliable decision-making, many GNN explainers have been proposed in recent years. However, these methods often encounter limitations, including their dependence on specific instances, lack of generalizability to unseen graphs, producing potentially invalid explanations, and yielding inadequate fidelity. To overcome these limitations, we, in this paper, introduce the Auxiliary Classifier Generative Adversarial Network (ACGAN) into the field of GNN explanation and propose a new GNN explainer dubbed *ACGAN-GNNExplainer*. Our approach leverages a generator to produce explanations for the original input graphs while incorporating a discriminator to oversee the generation process, ensuring explanation fidelity and improving accuracy. Experimental evaluations conducted on both synthetic and real-world graph datasets demonstrate the superiority of our proposed method compared to other existing GNN explainers.

KEYWORDS

graph neural networks; explanations; graph neural network explainer; conditional generative adversarial network

ACM Reference Format:
Yiqiao Li, Jianlong Zhou, Yifei Dong, Niusha Shafiabady, and Fang Chen. 2023. ACGAN-GNNExplainer: Auxiliary Conditional Generative Explainer for Graph Neural Networks. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Graph neural networks (GNNs) have emerged as a powerful tool for modelling graph-structured data and a natural choice for a variety of real-world applications such as crime prediction [22], traffic forecasting [7], and medical diagnosis [2], due to their ability to capture complex relationships between nodes and extract meaningful features from graphs. Notwithstanding its widespread adoption, its internal working mechanism remains a mystery, presenting potential challenges to its credibility and hindering its broader adoption in critical domains where explainability and transparency are essential.

GNN explainers such as GNNExplainer [25], XGNN [26], and PGExplainer [13] have gained increasing attention in the field of explainable artificial intelligence (XAI), which attempts to identify the most important graph structures and/or features that contribute to GNNs' predictions. These methods have provided some elucidation of GNNs; however, substantial work is still required in the following aspects: 1) *Explanation scale* (local or global explanation): whether the explanation is linked to a specific instance or whether it has captured the archetypal patterns shared by the same group; 2) *Generalizability*: whether an explainer can be generalized to unseen graphs without retraining; 3) *Fidelity*: whether the explanations are real important subgraphs; 4) *Versatility*: whether an explainer is able to generate accurate explanations for different tasks such as node classification and graph classification. The pioneering method GNNExplainer [25], for instance, is limited to local explanation and lacks generalizability. Later, XGNN [26] addressed this limitation but still lacks generalizability. Recent Gem [10] has overcome the limitations of its predecessors, but the nature of its generation process makes its precision in explaining various tasks unstable.

In order to address the aforementioned challenges, we, in this paper, propose a new GNN explanation method dubbed ACGAN-GNNExplainer, which uses the auxiliary classifier Generative Adversarial Network (ACGAN) [16] as its backbone to generate explanations for graph neural networks. In particular, it consists of a generator and a discriminator. The generator learns to produce explanations based on these two pieces of information—the original graph $\mathbf{G}$ that requires an explanation and its corresponding label $f(\mathbf{G})$, which is determined by the target GNN model f. In addition, a discriminator is adopted to distinguish whether the generated explanations are "real" or "fake" and to designate a prediction label to each explanation. In this way, the discriminator could provide "feedback" to the generator and further monitor the entire generation process. Through iterative iterations of this interplay learning process between the generator and the discriminator, the generator ultimately is able to produce explanations akin to those deemed "real"; consequently, the quality of the final explanation is enhanced, and the overall explanation accuracy is significantly increased. Although ACGAN has been widely used in various domains (e.g., image processing [17], data augmentation [29], medical image analysis [20], etc.), to the best of our knowledge, this is the first time that ACGAN has been used to explain GNN models. Our method *ACGAN-GNNExplainer* has the following merits: 1) it learns the underlying pattern of graphs, thus naturally providing explanations on a goal scale; 2) after learning the underlying pattern, it can produce explanations for unseen graphs without retraining; 3) it is more likely to generate valid important subgraphs with the consistent monitoring of the discriminator; 4) it is capable of performing well under different tasks, including node classification and graph classification.

Our main contributions to this paper could be summarized as the following points:

- We present a novel explainer, dubbed *ACGAN-GNNExplainer*, for GNN models, which employs a generator to generate explanations and a discriminator to consistently monitor the generation process;
- We empirically evaluate and demonstrate the superiority of our method *ACGAN-GNNExplainer* over other existing methods on various graph datasets, including synthetic and real-world graph datasets, and tasks, including node classification and graph classification.

2 RELATED WORK

2.1 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [5] are composed of two neural networks: a generator and a discriminator, trained in a game-like manner. The generator takes random noise as input and generates samples intended to resemble the training data distribution. On the contrary, the discriminator takes both real and generated samples as input and distinguishes between them. The generator tries to fool the discriminator by generating realistic samples, while the discriminator learns to accurately distinguish between real and fake samples. GANs have demonstrated successful applications across a wide range of tasks, including image generation, style transfer, text-to-image synthesis, and video generation.

Furthermore, the increasing utilization of GANs has led to the proposal of various variations, reflecting ongoing innovation and refinement within the field. These novel approaches introduce new architectural designs, optimization techniques, and training strategies to improve the stability, convergence, and overall quality of GAN models.

Specifically, one strategy for expanding GANs involves incorporating side information. For instance, CGAN [14] proposes providing both the generator and discriminator with class labels to produce class conditional samples. Researchers in [19] demonstrate that class conditional synthesis significantly improves the quality of generated samples. Another avenue for expanding GANs involves tasking the discriminator with reconstructing side information. This is achieved by modifying the discriminator to include an auxiliary decoder network that outputs the class label of the training data or a subset of the latent variables used for sample generation. For example, Chen et al. [3] propose InfoGAN, a GAN-based model that maximizes the mutual information between a subset of latent variables and the observations. It is known that incorporating additional tasks can enhance performance on the original task. In the paper [15], the auxiliary decoder leverage pre-trained discriminators, such as image classifiers, to further improve the quality of synthesized images.

Motivated by the aforementioned variations, Odena et al. [16] introduce the Auxiliary Classifier Generative Adversarial Network (ACGAN), a model combining both strategies to leverage side information. Specifically, the proposed model is class-conditional, incorporating an auxiliary decoder tasked with reconstructing class labels. ACGAN is an extension of CGANs. ACGANs are designed not only to generate samples that are similar to the training data and conditioned on the input information, but also to classify the generated samples into different categories. In ACGANs, both the generator and the discriminator are conditioned on auxiliary information, such as class labels. The generator takes random noise as input and generates samples conditioned on the input information and a set of labels, while the discriminator not only distinguishes between real and fake samples but also classifies them into different categories based on the input information.

ACGANs provide a way to generate diverse samples that are conditioned on the input information and classified into different categories, making them useful tools in many applications, such as image processing, data augmentation, and data balancing. In particular, the authors [17] propose a semi-supervised image classifier based on ACGAN. Waheed et al. [20] apply ACGAN in medical image analysis. Furthermore, in [29], authors augment the data by applying ACGAN in the electrocardiogram classification system. Ding et al. [4] propose a tabular data sampling method that integrates the Knearest neighbour method and tabular ACGAN to balance normal and attack samples.

2.2 Graph Neural Network Explainers

Explaining the decision-making process of graph neural networks (GNNs) is a challenging and important research topic, as it could greatly benefit users by improving safety and promoting trust in these models. To achieve this goal, several popular approaches have emerged in recent years that aim to explain GNN models by leveraging the unique properties of graph features and structures. In this regard, we briefly review several representative GNN explainers below.

GNNExplainer [25] is a seminal method in the field of explaining GNN models. By identifying the most relevant features and subgraphs that are essential to the predictions made by the GNN model, GNNExplainer is able to provide local-scale explanations for GNN models. *PGExplainer* [13] generates explanations for GNN models by using a probabilistic graph. Compared to GNNExplainer, it provides model-level explanations for each instance and has

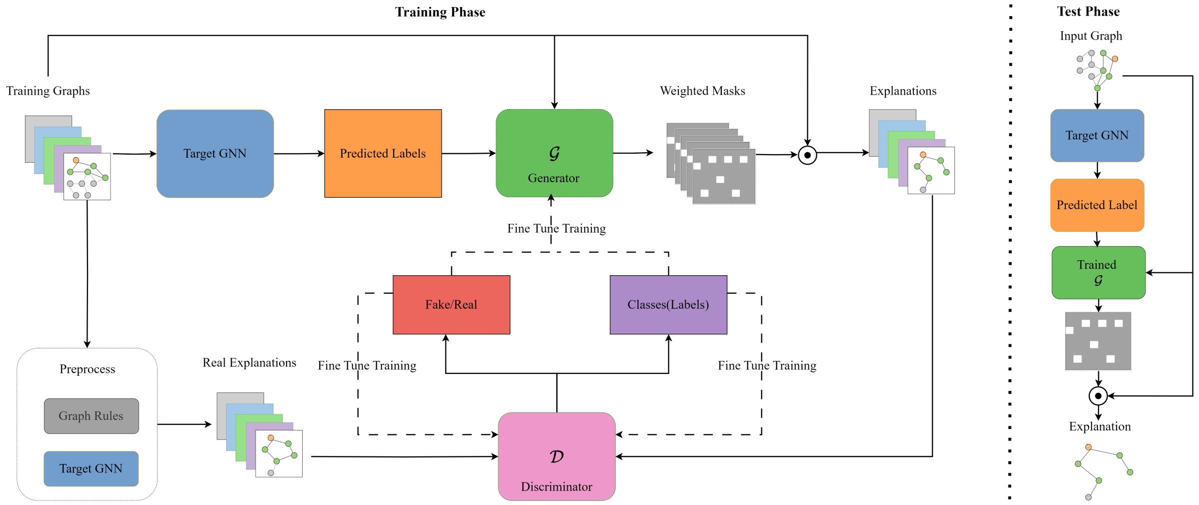

Figure 1: The framework of ACGAN-GNNExplainer. The ⊙ means element-wise multiplication. This figure includes two phases: the training phase and the test phase. During the Training Phase, the objective is to train the generator and discriminator of the ACGAN-GNNExplainer model. After successful training, the Test Phase then utilizes the trained generator to generate explanations for the testing data.

strong generalizability. Recent Gem [10] offers local and global explanations and operates in an inductive configuration, allowing it to explain GNN models without retraining. Lin et al. [11] later propose OrphicX, which uses latent causal factors to generate causal explanations for GNN models. However, Gem [10] and OrphicX [11] face difficulties in achieving consistently accurate explanations on real-world datasets. Therefore, in this paper, we attempt to develop a GNN explainer that is capable of generating explanations with high fidelity and precision for both synthetic and real-world datasets.

Furthermore, reinforcement learning is another prevalent technique employed for explicating GNN models. For example, Yuan et al. [27] propose XGNN, a model-level explainer that trains a graph generator to generate graph patterns to maximize a specific prediction of the model. Wang et al. [23] introduce RC-Explainer, which generates causal explanations for GNNs by combining the causal screening process with a Markov Decision Process in reinforcement learning. Furthermore, Shan et al. [18] propose RG-Explainer, a reinforcement learning enhanced explainer that can be applied in the inductive setting, demonstrating its better generalization ability.

In addition to the works we have mentioned above, there is another line of work that is working on generating counterfactual explanations. For example, the CF-GNNExplainer [12] generates counterfactual explanations for the majority of instances of GNN explanations. Furthermore, Bajaj et al. [1] propose an RCExplainer that generates robust counterfactual explanations, and Wang et al. [24] propose ReFine, which pursues multi-grained explainability by pre-training and fine tuning.

3 METHOD

3.1 Problem Formulation

The notions of "interpretation" and "explanation" are crucial in unravelling the underlying working mechanisms of GNNs. Interpretation entails understanding the decision-making process of the model, prioritizing transparency, and the ability to trace the trajectory of decisions. In contrast, an explanation furnishes a rationale or justification for the predictions of GNNs, striving to present a coherent and succinct reasoning for the outcomes.

In this paper, we attempt to identify the subgraphs that significantly impact the predictions of GNNs. We represent a graph as $\mathbf{G} = (\mathbf{V}, \mathbf{A}, \mathbf{X})$, where $\mathbf{V}$ is the set of nodes, $\mathbf{A} \in \{0, 1\}$ denotes the adjacency matrix that $\mathbf{A}_{ij} = 1$ if there is an edge between node i and node j, otherwise $\mathbf{A}_{ij} = 0$, and $\mathbf{X}$ indicates the feature matrix of the graph $\mathbf{G}$. We also have a GNN model f and $\hat{l}$ denotes its prediction, $f(\mathbf{G}) \rightarrow \hat{l}$. We further define $E(f(\mathbf{G}), \mathbf{G})) \rightarrow \mathbf{G}^s$ as the explanation of a GNN explainer. Ideally, when feeding the explanation into the GNN model f, it would produce the exact same prediction $\hat{l}$, which means that $f(\mathbf{G})$ equals $f(E(f(\mathbf{G}), \mathbf{G}))$. We also expect that the explanation $E(f(\mathbf{G}), \mathbf{G})) \rightarrow \mathbf{G}^s$ should be a subgraph of the original input graph $\mathbf{G}$, which means that $\mathbf{G}^s \in \mathbf{G}$, so that the explained graph is a valid subgraph.

3.2 Obtaining Causal Real Explanations

Our objective in this paper is to elucidate the reasoning behind the predictions made by the target GNN model f. To achieve this, we regard the target GNN model f as a black box and refrain from investigating its internal mechanisms. Instead, we attempt to identify the subgraphs that significantly affect the predictions of the target GNN model f. In particular, we employ a generative model to autonomously generate these subgraphs/explanations. In order for the generative model to generate faithful explanations, it must first be trained under the supervision of "real" explanations (ground truth). However, these ground truths are typically unavailable in real-world applications. In this paper, we employ Granger causality [6], which is commonly used to test whether a specific variable has a causal effect on another variable, to circumvent this difficulty.

Specifically, in our experiments, we mask an edge and then observe its effect on the prediction of the target GNN model f. We then calculate the difference between the prediction probability of the original graph and the masked graph and set this difference as an edge weight to indicate its effect on the prediction of the target GNN model f. After that, we sort all edges of the graph according to the weight values we have obtained and save the resulting weighted graph. Therefore, edges with the highest weights correspond to actual explanations (important subgraphs). However, it should also be noted that using Granger causality [6] directly to explain a target GNN model f is computationally intensive and has limited generalizability. Our method, on the other hand, could naturally overcome this challenge, as our parameterized explainer could capture the fundamental patterns shared by the same group and is adaptable and transferable across different graphs once the shared patterns have been comprehensively learned.

3.3 ACGAN-GNNExplainer

Using the generating capacity of ACGAN, in this paper, we propose an ACGAN-based explanation method for GNN models, which is termed ACGAN-GNNExplainer. It consists of a generator ($\mathcal{G}$) and a discriminator ($\mathcal{D}$). The generator $\mathcal{G}$ is used to generate the explanations, while the discriminator $\mathcal{D}$ is used to monitor the generation process. The detailed framework of our method ACGAN-GNNExplainer is depicted in Figure 1.

In contrast to the conventional strategy of training an ACGAN, in which random noise $\mathbf{z}$ is fed into the generator $\mathcal{G}$, our model feeds the generator $\mathcal{G}$ with the original graph $\mathbf{G}$, which is the graph we want to explain, and the label L, which is predicted by the target GNN model f. Employing this strategy, we ensure that the explanation produced by the generator $\mathcal{G}$, which plays a crucial role in determining the predictions of the GNN model f, corresponds to the original input graph $\mathbf{G}$. In addition, the generator $\mathcal{G}$ trained under this mechanism can be easily generalized to unseen graphs without significant retraining, thus saving computational costs. For the generator $\mathcal{G}$, we employ an encoder-decoder network where the encoder would project the original input graph $\mathbf{G}$ into a compact hidden representation, and the decoder would then reconstruct the explanation from the compact hidden representation. In our case, the reconstructed explanation is a mask matrix that indicates the significance of each edge.

Conceptually, the generator $\mathcal{G}$ is capable of generating any explanation (valid or invalid) if it is sufficiently complex, which contradicts the objective of *explaining* a GNN. Inspired by ACGAN, we adopt a discriminator $\mathcal{D}$ to monitor the generating process of $\mathcal{G}$. Specifically, our discriminator $\mathcal{D}$ is a graph classifier with five convolutional layers. It is fed with the real explanation and the explanation generated by our generator $\mathcal{G}$. It attempts to identify whether the explanation is "real" or "fake" and, at the same time classify the explanation, which serves as "feedback" to the generator $\mathcal{G}$ and further encourages the generator $\mathcal{G}$ to produce faithful explanations.

In addition, in order to train our generator $\mathcal{G}$ and discriminator $\mathcal{D}$, we need to obtain the "real" explanations first. To achieve this goal, we incorporate pre-processing in our framework (Fig. 1), which uses the Granger causality [6] to acquire the "real" explanations. The details can be found in Section in 3.2. Once the input graph $\mathbf{G}$, its corresponding real subgraphs (ground truth), and the labels have been acquired. We can train our ACGAN-GNNExplainer to produce a weighted mask that effectively highlights the edges and nodes in the original input graph $\mathbf{G}$ that significantly contribute to the decision-making process of the given GNN model f. Then, by multiplying the mask by the original adjacency matrix of the input graph, we obtain the corresponding explanations/important subgraphs. These explanations are particularly useful for comprehending the reasoning behind the complex GNN model.

3.4 Improved Loss Function

The generator $\mathcal{G}$ generates explanations/subgraphs $\mathbf{G}^s \in \mathbf{G}$ based on two essential inputs: the original graph $\mathbf{G}$ and its associated label l, as expressed by $\mathbf{G}^s \leftarrow \mathcal{G}(\mathbf{G}, l)$. Concurrently, the discriminator $\mathcal{D}$ assesses the probabilities of origins ($S = \{"real", "fake"\}$) denoted as $P(S \mid \mathbf{G})$, and the probabilities of class classification (class labels $L = \{l_1, \cdots, l_n\}$) denoted as $P(L \mid \mathbf{G})$. Consequently, the loss function of the discriminator comprises two components: the log-likelihood of the correct source, $\mathcal{L}_S$, defined in Equation 1, and the log-likelihood of the correct class, $\mathcal{L}_L$, defined in Equation 2.

$$\begin{aligned}\mathcal{L}_S =& \mathbb{E}\left[\log P\left(S = \text{ real } \mid \mathbf{G}\right)\right] + \\ & \mathbb{E}\left[\log P\left(S = \text{ fake } \mid \mathbf{G}^{\mathrm{s}}\right)\right]\end{aligned} \tag{1}$$

$$\begin{aligned}\mathcal{L}_L =& \mathbb{E}\left[\log P\left(L = l \mid \mathbf{G}\right)\right] + \\ & \mathbb{E}\left[\log P\left(L = l \mid \mathbf{G}^{\mathrm{s}}\right)\right]\end{aligned} \tag{2}$$

The discriminator $\mathcal{D}$ and the generator $\mathcal{G}$ play a minimax game, engaging in competitive interactions. The primary objective of the discriminator $\mathcal{D}$ is to maximize the probability of accurately classifying real and fake graphs ($\mathcal{L}_S$), as well as correctly predicting the class label ($\mathcal{L}_L$) for all graphs, resulting in a combined objective such as maximizing ($\mathcal{L}_S + \mathcal{L}_L$).

Conversely, the generator $\mathcal{G}$ aims to minimize the discriminator's capacity to identify real and fake graphs while simultaneously maximizing the discriminator's ability to classify them, as indicated by a combined objective like maximizing ($-\mathcal{L}_S + \mathcal{L}_L$). Thus, based on the Equation 1 and Equation 2, the objective function of the $\mathcal{D}$ and $\mathcal{G}$ are formulated in Equation 3 and Equation 4, respectively.

$$\begin{aligned}\mathcal{L}_{(\mathcal{D})} =& -\mathbb{E}_{\mathbf{G}^{gt} \sim P(\mathbf{G}^{gt})} \log \mathcal{D}(\mathbf{G}^{gt}) \\ & -\mathbb{E}_{\mathbf{G} \sim P(\mathbf{G})} \log[1 - \mathcal{D}(\mathcal{G}(\mathbf{G}, l))] \\ & -\mathbb{E}_{\mathbf{G}^{gt} \sim P(\mathbf{G}^{gt})} P(L \mid \mathbf{G}^{gt}) \\ & -\mathbb{E}_{\mathbf{G} \sim P(\mathbf{G})} \log(P(L \mid \mathcal{G}(\mathbf{G}, l))\end{aligned} \tag{3}$$

$$\begin{aligned}\mathcal{L}_{(\mathcal{G})} =& -\mathbb{E}_{\mathbf{G} \sim P(\mathbf{G})} \log \mathcal{D}(\mathcal{G}(\mathbf{G}, l)) \\ & -\mathbb{E}_{\mathbf{G} \sim P(\mathbf{G})} \log P(L \mid \mathcal{G}(\mathbf{G}, l))\end{aligned} \tag{4}$$

Here, **G** represents the original graph that requires explanation, while $\mathbf{G}^{gt}$ signifies its corresponding actual explanation (e.g., the real important subgraph).

Using the objective functions as detailed in Equation 3 and Equation 4 for training the discriminator $\mathcal{D}$ and generator $\mathcal{G}$, it is observed that fidelity of the generated explanation is not satisfactory. This could be attributed to the fact that $\mathcal{L}_{(\mathcal{G})}$ as in Equation 4 does not explicitly incorporate the fidelity information from a target GNN model f. To overcome this limitation and enhance both the fidelity and accuracy of the explanation, we intentionally integrate the fidelity of the explanation into our objective function. Finally, we derive an enhanced generator ($\mathcal{G}$) loss function, as defined in Equation 5.

$$\begin{aligned}\mathcal{L}_{(\mathcal{G})} = &- \mathbb{E}_{\mathbf{G}\sim P(\mathbf{G})} \log \mathcal{D}(\mathcal{G}(\mathbf{G}, l)) \\ &- \mathbb{E}_{\mathbf{G}\sim p(\mathbf{G})} \log P(L \mid \mathcal{G}(\mathbf{G}, l)) \\ &+ \lambda \mathcal{L}_{Fid}\end{aligned} \tag{5}$$

$$\mathcal{L}_{Fid} = \frac{1}{N} \sum_{i=1}^{N} ||f(\mathbf{G}) - f(\mathcal{G}(\mathbf{G}))||^2 \tag{6}$$

In this context, $\mathcal{L}_{Fid}$ represents the loss function component associated with fidelity. f symbolizes a pre-trained target GNN model, N signifies the count of node set of **G**, and **G** represents the original graph intended for explanation. Correspondingly, $\mathbf{G}^{gt}$ stands for the explanation ground truth associated with it (e.g., the real important subgraph). Within this framework, λ is a trade-off hyperparameter responsible for adjusting the relative significance of the ACGAN model and the explanation accuracy obtained from the pre-trained target GNN f. Setting λ to zero results in Equation 5 being precisely equivalent to Equation 4. Notably, for our experiments, we selected $\lambda = 2.0$ for synthetic graph datasets, $\lambda = 4.0$ for the Mutagenicity dataset, and $\lambda = 4.5$ for the NCI1 dataset.

3.5 Pseudocode of ACGAN-GNNExplainer

In Sections 3.3 and 3.4, we have described the framework and loss functions of our method in detail. To further elucidate our method, we provide its pseudocode in Algorithm 1.

4 EXPERIMENTS

In this section, we undertake a comprehensive evaluation of the performance of our proposed method, ACGAN-GNNExplainer. We first introduce the datasets we used in our experiments, as well as the implementation details in Section 4.1. After that, we show the quantitative evaluation of our method in comparison with other representative GNN explainers on synthetic datasets (see Section 4.2) and real-world datasets (see Section 4.3). Finally, we also provide a qualitative analysis and visualize several explanation samples generated by our method, as well as other representative GNN explainers in Section 4.4.

4.1 Implementation Details

Datasets. We focus on two widely used synthetic node classification datasets, including BA-Shapes and Tree-Cycles [25], and two real-world graph classification datasets, Mutagenicity [8] and NCI1 [21]. Details of these datasets are shown in Table 1.

Algorithm 1: Training a ACGAN-GNNExplainer

```
Input: Graph data $G = \{g_1, \cdots, g_n\}$ with labels
       $L = \{l_1, \cdots, l_n\}$, real explanations for graph data
       $G^{gt} = \{g_1^{gt}, \cdots, g_n^{gt}\}$ (obtained in preprocessing
       phase), a pre-trained GNN model $f$
Output: A well-trained Generator $\mathcal{G}$, a well-trained
        discriminator $\mathcal{D}$
Initialize the Generator $\mathcal{G}$ and the Discriminator $\mathcal{D}$ with
  random weights
for epoch in epochs do
    Sample a minibatch of $m$ real data samples
      $\{g^{(1)}, \cdots, g^{(m)}\}$ and real labels $\{l^{(1)}, \cdots, l^{(m)}\}$
    Generate fake data samples
      $\{g^{s(1)}, \cdots, g^{s(m)}\} \leftarrow \mathcal{G}(g^{(1)}, \cdots, g^{(m)})$ and obtain
      their labels $\{l^{s(1)}, \cdots, l^{s(m)}\}$
    Update $\mathcal{D}$ with the gradient:
        $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\mathcal{L}_{(\mathcal{D})}]$
    Update $\mathcal{G}$ with the gradient:
        $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [\mathcal{L}_{(\mathcal{G})}]$
end
```

Table 1: Details of synthetic and real-world datasets.

	Node Classification		Graph Classification	
	BA-Shapes	Tree-Cycles	Mutagenicity	NCI1
# of Graphs	1	1	4,337	4110
# of Edges	4110	1950	266,894	132,753
# of Nodes	700	871	131,488	122,747
# of Labels	4	2	2	2

Baseline Approaches. Due to the growing prevalence of GNN in a variety of real-world applications, an increasing number of research studies seek to explain GNN, thereby enhancing its credibility and nurturing trust. Among them, we identify three representative GNN explainers as our competitors: *GNNExplainer* [25], *OrphicX* [11] and *Gem* [10]. For these competitors, we adopt their respective official implementations.

Different Top Edges (K or R). After obtaining the weight (importance) of each edge for the input graph **G**, it is also important to select the right number of edges to serve as the explanations as selecting too few edges may lead to an incomplete explanation/subgraph while selecting too many edges may introduce a lot of noisy information into our explanation. To overcome this uncertainty, we specifically define a top K (for synthetic datasets) and a top R (for real-world datasets) to indicate the number of edges we would like to select. We test different K and R to show the stability of our method. To be specific, we set $K = \{5, 6, 7, 8, 9\}$

Table 2: The fidelity and accuracy of explanations on BA-Shapes dataset: $Fid^{+}(\uparrow)$, $Fid^{-}(\downarrow)$, $ACC_{exp}(\uparrow)$.

K	5			6			7			8			9		
(top edges)	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}
GNNExplainer	0.7059	0.1471	0.7941	0.6765	0.0588	0.8824	0.7059	0.0294	0.9118	0.7353	0.0000	0.9412	0.7353	0.0294	0.9118
Gem	0.5588	**0.0000**	**0.9412**	0.5588	**-0.0294**	**0.9706**	0.5882	**-0.0294**	**0.9706**	0.5882	**-0.0294**	**0.9706**	0.5882	-0.0294	0.9706
OrphicX	**0.7941**	0.2059	0.7353	**0.7941**	0.2059	0.7353	**0.7941**	0.0882	0.8529	**0.7941**	0.0588	0.8824	**0.7941**	0.0588	0.8824
Our Method	0.6471	0.1471	0.7941	0.5882	0.0882	0.8529	0.6176	**-0.0294**	**0.9706**	0.6471	**-0.0294**	**0.9706**	0.6471	**-0.0588**	**1.0000**

Table 3: The fidelity and accuracy of explanations on Tree-Cycles dataset: $Fid^{+}(\uparrow)$, $Fid^{-}(\downarrow)$, $ACC_{exp}(\uparrow)$.

K	6			7			8			9			10		
(top edges)	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}
GNNExplainer	0.9143	0.8000	0.1714	0.9429	0.4571	0.5143	**0.9714**	0.1714	0.8000	**0.9714**	0.0571	0.9143	**0.9714**	0.0571	0.9143
Gem	**0.9714**	0.2571	0.7143	**0.9714**	0.1429	0.8286	**0.9714**	0.2571	0.7143	**0.9714**	0.1143	0.8571	**0.9714**	0.0857	0.8857
OrphicX	0.9429	**0.0000**	**0.9714**	0.9429	0.0000	0.9714	0.9429	**-0.0286**	**1.0000**	0.9429	**-0.0286**	**1.0000**	0.9429	**-0.0286**	**1.0000**
Our Method	**0.9714**	**0.0000**	**0.9714**	**0.9714**	**-0.0286**	**1.0000**	**0.9714**	0.0286	0.9429	**0.9714**	0.0571	0.9143	**0.9714**	0.0000	0.9714

for the BA-Shapes dataset, $K = \{6, 7, 8, 9, 10\}$ for the Tree-Cycles dataset, and $R = \{0.5, 0.6, 0.7, 0.8, 0.9\}$ for real-world datasets.

Data Split. To maintain consistency and fairness in our experiments, we divide the data into three sets: 80% for training, 10% for validation, and 10% for testing. Testing data remain untouched throughout the experiments.

Evaluation Metrics. A good GNN explainer should be able to generate concise explanations/subgraphs while maintaining high prediction accuracy when these explanations are fed into the target GNN. Therefore, it is desirable to evaluate the method with different metrics [9]. In our experiments, we use the accuracy and fidelity of the explanation as our performance metrics.

In particular, we generate explanations for the test set using *GNNExplainer* [25], *Gem* [10], *OrphicX* [11], and ACGAN-GNNExplainer (our method), respectively. We then feed these generated explanations to the pre-trained target GNN f to compute the accuracy, which can be formally defined as Equation 7:

$$ACC_{exp} = \frac{|f(\mathbf{G}) = f(\mathbf{G}^{s})|}{|T|} \tag{7}$$

where $\mathbf{G}$ signifies the initial graph necessitating explanation, and $\mathbf{G}^{s}$ denotes its associated explanation (e.g. the significant subgraph); $|f(\mathbf{G}) = f(\mathbf{G}^{s})|$ represents the count of accurately classified instances in which the predictions of f on $\mathbf{G}$ and $\mathbf{G}^{s}$ are exactly the same, and $|T|$ is the total number of instances.

In addition, fidelity is a measure of how faithfully the explanations capture the important subgraphs of the input original graph. In our experiments, we employ the $Fidelity^{+}$ and $Fidelity^{-}$ [28] to evaluate the fidelity of the explanations.

$Fidelity^{+}$ quantifies the variation in the predicted accuracy between the original predictions and the new predictions generated by excluding the important input features. On the contrary, $Fidelity^{-}$ denotes the changes in prediction accuracy when significant input features are retained while non-essential structures are removed. Evaluation of both $Fidelity^{+}$ and $Fidelity^{-}$ provides a comprehensive insight into the precision of the explanations to capture the behaviour of the model and the importance of different input features. $Fidelity^{+}$ and $Fidelity^{-}$ are mathematically described in Equation 8 and Equation 9, respectively.

$$Fid^{+} = \frac{1}{N}\sum_{i=1}^{N}(f(\mathbf{G}_i)_{l_i} - f(\mathbf{G}_i^{1-s})_{l_i}) \tag{8}$$

$$Fid^{-} = \frac{1}{N}\sum_{i=1}^{N}(f(\mathbf{G}_i)_{l_i} - f(\mathbf{G}_i^{s})_{l_i}) \tag{9}$$

In these equations, N denotes the total number of samples, and l_i represents the class label for instance i. $f(\mathbf{G}_i)_{l_i}$ and $f(\mathbf{G}_i^{1-s})_{l_i}$ correspond to the prediction probabilities for class l_i using the original graph $\mathbf{G}_i$ and the occluded graph $\mathbf{G}_i^{1-s}$, respectively. The occluded graph is derived by removing the significant features identified by the explainers from the original graph. A higher value of $Fidelity^{+}$ is preferable, indicating a more essential explanation. In contrast, $f(\mathbf{G}_i^{s})_{l_i}$ represents the prediction probability for class l_i using the explanation graph $\mathbf{G}_i^{s}$, which encompasses the crucial structures identified by explainers. A lower $Fidelity^{-}$ value is desirable, signifying a more sufficient explanation.

In general, the accuracy of the explanation (ACC_{exp}) assesses the accuracy of the explanations, while $Fidelity^{+}$ and $Fidelity^{-}$ assess their necessity and sufficiency, respectively. A higher $Fidelity^{+}$ suggests a more essential explanation, while a lower $Fidelity^{-}$ implies a more sufficient one. Through comparison of accuracy and fidelity across different explainers, we can derive valuable insights into the performance and suitability of each approach.

Table 4: The fidelity and accuracy of explanations on Mutagenicity dataset: $Fid^{+}(\uparrow)$, $Fid^{-}(\downarrow)$, $ACC_{exp}(\uparrow)$.

R	0.5			0.6			0.7			0.8			0.9		
(edge ratio)	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}
GNNExplainer	0.3618	**0.2535**	**0.6175**	0.3825	0.2742	0.5968	0.3963	0.2396	0.6313	0.3641	0.1774	0.6935	0.3641	0.0899	0.7811
Gem	0.3018	0.2972	0.5737	0.3295	0.2696	0.6014	0.2857	0.2120	0.6590	0.2581	0.1475	0.7235	0.2120	0.0806	0.7903
OrphicX	0.2419	0.4171	0.4539	0.2949	0.3111	0.5599	0.2995	0.2465	0.6244	0.3157	0.1613	0.7097	0.2949	**0.0599**	**0.8111**
Our Method	**0.3963**	**0.2535**	**0.6175**	**0.3828**	**0.2673**	**0.6037**	**0.3986**	**0.1636**	**0.7074**	**0.3602**	**0.1037**	**0.7673**	**0.3871**	0.0806	0.7903

Table 5: The fidelity and accuracy of explanations on NCI1 dataset: $Fid^{+}(\uparrow)$, $Fid^{-}(\downarrow)$, $ACC_{exp}(\uparrow)$.

R	0.5			0.6			0.7			0.8			0.9		
(edge ratio)	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}	Fid^{+}	Fid^{-}	ACC_{exp}
GNNExplainer	0.3358	0.2749	0.5961	0.3625	0.2603	0.6107	0.3844	0.1922	0.6788	0.3747	0.1095	0.7616	0.3236	0.0584	0.8127
Gem	0.3796	0.3066	0.5645	0.4307	0.2628	0.6083	0.4282	0.1873	0.6837	0.4404	0.1192	0.7518	0.3212	0.0389	0.8321
OrphicX	0.3114	0.3090	0.5620	0.3431	0.3236	0.5474	0.3382	0.2628	0.6083	0.3698	0.1630	0.7080	0.3139	0.0608	0.8102
Our Method	**0.4015**	**0.2141**	**0.6569**	**0.4523**	**0.2214**	**0.6496**	**0.4453**	**0.1849**	**0.6861**	**0.4672**	**0.0779**	**0.7932**	**0.3942**	**0.0254**	**0.8446**

4.2 Experiments on Synthetic Datasets

We first conduct experiments on two common synthetic datasets including BA-Shapes and Tree-Cycles [25], of which the details can be found in Section 4.1. We assess the fidelity and accuracy of the explanations generated by GNNExplainer, Gem, OrphicX, and our proposed ACGAN-GNNExplainer (our method). Table 2 and Table 3 present the fidelity and accuracy of explanations for BA-Shapes and Tree-Cycles datasets across different K, respectively.

When examining the results for the BA-Shapes, as shown in Table 2, it is evident that no single model consistently surpasses the others across all metrics. However, as the value of K increases, ACGAN-GNNExplainer progressively achieves competitive explanation accuracy ACC_{exp} and better performance of $Fidelity^{-}$. On the contrary, OrphicX consistently exhibits higher $Fidelity^{+}$ values for various K, highlighting its proficiency in capturing essential subgraphs. However, its performance in terms of explanation accuracy ACC_{exp} and $Fidelity^{-}$ lags behind, indicating that it struggles to provide comprehensive and precise explanations. Upon analyzing the results presented in Table 3, it is evident that all methods demonstrate a commendable performance on the Tree-Cycles datasets with different K values. However, no single method consistently outperforms the others in all evaluation metrics, which shows a trend similar to the results in the BA shapes (see Table 2). Notably, within the range of $K = \{6, 7\}$, ACGAN-GNNExplainer emerges as the superior choice among all the alternatives. It maintains the highest fidelity compared to the other methods on all K values. Although outperformed by Orphicx in terms of $Fidelity^{-}$ and accuracy ACC_{exp} when K is in the range of $\{8, 9, 10\}$, ACGAN-GNNExplainer still shows competitive performance.

In summary, all GNN explainers manifest robust performance in synthetic datasets, largely attributed to their intrinsic simplicity in contrast to real-world datasets. Notably, ACGAN-GNNExplainer consistently outperforms alternative methods in several scenarios. Moreover, even in situations where ACGAN-GNNExplainer does not outshine its counterparts, it maintains competitive levels of performance. To offer a comprehensive evaluation of ACGAN-GNNExplainer, we extend our exploration to real-world datasets in the forthcoming Section 4.3, facilitating a thorough analysis.

4.3 Experiments on Real-world Datasets

Here we further experiment with our method with two popular real-world datasets including Mutagenicity [8] and NCI1 [21]. The experimental results for Mutagenicity and NCI1 are shown in Table 4 and Table 5, respectively.

From Table 4, it can be seen that ACGAN-GNNExplainer demonstrates superior performance in both fidelity ($Fidelity^{+}$, $Fidelity^{-}$) and accuracy ACC_{exp} in most settings where R ranges from 0.5 to 0.8. While OrphicX marginally outperforms ACGAN-GNNExplainer in terms of explanation accuracy ACC_{exp} when $R = 0.9$, its fidelity lags behind. However, maintaining high fidelity without sacrificing accuracy is crucial when explaining GNNs in practice. From this perspective, our method shows an obvious advantage over others. Similarly, from Table 5, one can observe that ACGAN-GNNExplainer consistently outperforms other competitors in terms of fidelity and accuracy in different R.

Our method consistently yields higher $Fidelity^{+}$ scores, suggesting that our generated explanations have successfully covered the important subgraphs. On the other hand, our method achieved lower $Fidelity^{-}$ scores compared to other methods. This highlights the sufficiency of our explanations, as they effectively conveyed the necessary information for accurate predictions while mitigating inconsequential noise. Furthermore, in terms of accuracy, our method consistently yields higher explanation accuracy compared with other methods, underscoring its proficiency in effectively capturing the underlying rationale of the GNN model. In general, these results

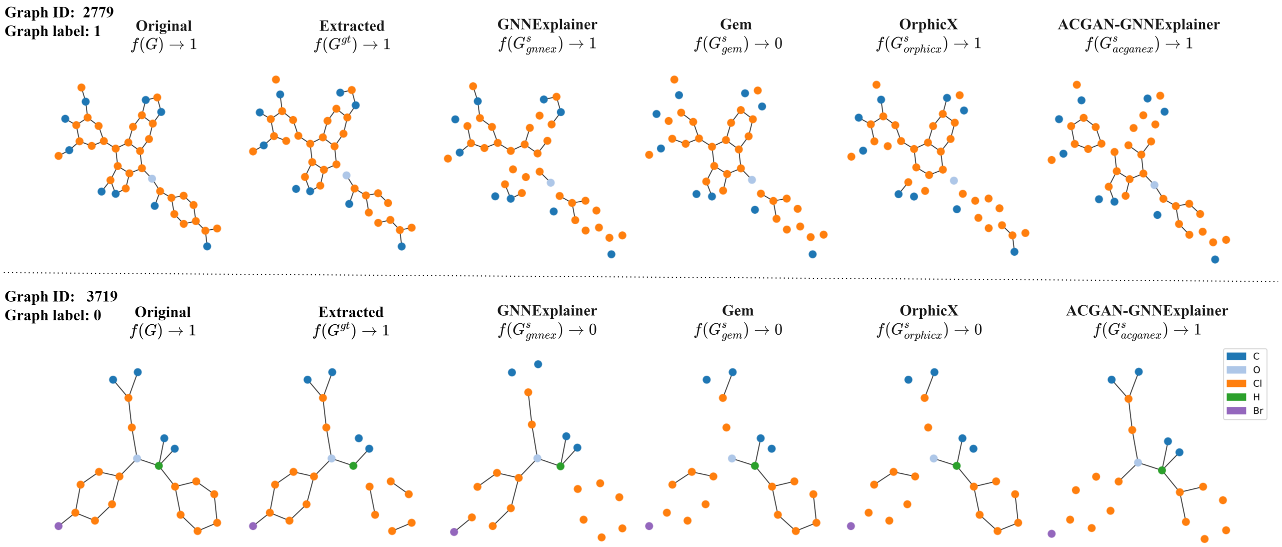
Graph ID: 2779
Graph label: 1
Original
$f(G) \rightarrow 1$
Extracted
$f(G^{gt}) \rightarrow 1$
GNNExplainer
$f(G^s_{gnnex}) \rightarrow 1$
Gem
$f(G^s_{gem}) \rightarrow 0$
OrphicX
$f(G^s_{orphicx}) \rightarrow 1$
ACGAN-GNNExplainer
$f(G^s_{acganex}) \rightarrow 1$
Graph ID: 3719
Graph label: 0
Original
$f(G) \rightarrow 1$
Extracted
$f(G^{gt}) \rightarrow 1$
GNNExplainer
$f(G^s_{gnnex}) \rightarrow 0$
Gem
$f(G^s_{gem}) \rightarrow 0$
OrphicX
$f(G^s_{orphicx}) \rightarrow 0$
ACGAN-GNNExplainer
$f(G^s_{acganex}) \rightarrow 1$
C
O
Cl
H
Br

Figure 2: The explanation visualization on NCI1 when $R = 0.5$. $f(\cdot) \rightarrow \{0, 1\}$ means predictions made by the target GNN model f. The 1^{st} column contains the initial graph. The 2^{nd} column showcases the real explanation that we obtained during the preprocessing stage. The 3^{rd} to 5^{th} columns are the explanations produced by GNNExplainer, Gem, OrphicX and ACGAN-GNNExplainer, respectively. On analyzing the first row, we observe that GNNExplainer, OrphicX, and ACGAN-GNNExplainer successfully obtain the explanations that are successfully classified by the target GNN model f. However, upon examining the visualization of the explanation subgraph, it is obvious that the explanation produced by ACGAN-GNNExplainer exhibits the closest resemblance to the real explanations. Moving on to the second row, we find that ACGAN-GNNExplainer tends to select molecules other than the Cl circle as part of the explanation subgraph. In contrast, other competitors have a tendency to include the Cl molecule circle as part of the explanation subgraph.

highlight the effectiveness of our proposed method in producing faithful explanations.

4.4 Qualitative Analysis

Qualitative evaluation is another effective way to compare explanations generated by different explainers. Here, we present visualizations of the explanations on NCI1 with $R = 0.5$ and visualize two examples of explanations—the target GNN model f successfully classifies one example but fails to classify the other one. We try to investigate the factors that affect the predictions of the target GNN model f—resulting in a correct prediction or causing a wrong prediction. Specifically, when the target GNN model f yields a correct prediction (e.g., the first-row visualization example in Figure 2), our objective is to provide an explanation that would highlight the key elements that lead to the correct prediction. Conversely, when the target GNN model f produces an incorrect prediction (e.g., the second-row visualization example in Figure 2), we hope to offer an explanation that elucidates the factors contributing to the incorrect prediction.

Therefore, our goal is to ensure that the explanation generated by our proposed method aligns well with the prediction made by the target GNN model f. In particular, when the target GNN model f accurately predicts the label for a given graph, we expect our explanation to yield the same prediction. As illustrated in the first row of Figure 2, we observe that GNNExplainer, Orphicx, and ACGAN-GNNExplainer provide correct explanations for the graph that the GNN correctly predicts. However, it is worth noting that the explanation subgraph generated by ACGAN-GNNExplainer exhibits the closest resemblance to the real explanation subgraph extracted in the preprocessing phase. Furthermore, when examining another graph for which the target GNN model f makes an incorrect prediction, we find that only ACGAN-GNNExplainer is capable of producing a correct explanation. Notably, the ACGAN-GNNExplainer model demonstrates a tendency to select other molecules as part of the explanation subgraph rather than the Cl circles. In contrast, other methods we have compared tend to include the Cl molecule circle as part of the explanation subgraph.

Visually, our method demonstrates a higher degree of visual similarity to the actual explanation in comparison to other competing methods. This observation provides additional evidence supporting the efficacy of our method in producing faithful explanations.

5 CONCLUSION

Unboxing the intrinsic operational mechanisms of a GNN is of paramount importance in bolstering trust in model predictions, ensuring the reliability of real-world applications, and advancing the establishment of trustworthy GNNs. In pursuit of these objectives, many methods have emerged in recent years. Although they demonstrate commendable functionality in certain aspects, most of them struggle in obtaining good performance on real-world datasets.

To address this limitation, we, in this paper, propose an ACGAN-based explainer, dubbed ACGAN-GNNExplainer, for graph neural networks. This framework comprises a generator and a discriminator, where the generator is used to generate the corresponding explanations for the original input graphs and the discriminator is used to monitor the generation process and signal feedback to the generator to ensure the fidelity and reliability of the generated explanations. To assess the effectiveness of our proposed method, we conducted comprehensive experiments on synthetic and real-world graph datasets. We performed fidelity and accuracy comparisons with other representative GNN explainers. The experimental findings decisively establish the superior performance of our proposed ACGAN-GNNExplainer in terms of its ability to generate explanations with high fidelity and accuracy for GNN models, especially on real-world datasets.

REFERENCES

[1] Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. 2021. Robust Counterfactual Explanations on Graph Neural Networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 5644–5655.

[2] Dehua Chen, Hongjin Zhao, Jianrong He, Qiao Pan, and Weiliang Zhao. 2021. An Causal XAI Diagnostic Model for Breast Cancer Based on Mammography Reports. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 3341–3349. https://doi.org/10.1109/BIBM52615.2021.9669648

[3] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 2172–2180.

[4] Hongwei Ding, Leiyang Chen, Liang Dong, Zhongwang Fu, and Xiaohui Cui. 2022. Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection. *Future Gener. Comput. Syst.* 131 (2022), 240–254. https://doi.org/10.1016/j.future.2022.01.026

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. 2672–2680.

[6] CWJ Granger. 2001. Investigating causal relations by econometric models and cross-spectral methods. In *Essays in econometrics: collected papers of Clive WJ Granger*. 31–47.

[7] Weiwei Jiang and Jiayun Luo. 2022. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications* 207 (Nov. 2022), 117921. https://doi.org/10.1016/j.eswa.2022.117921

[8] Jeroen Kazius, Ross McGuire, and Roberta Bursi. 2005. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of Medicinal Chemistry* 48, 1 (Jan. 2005), 312–320. https://doi.org/10.1021/jm040835a

[9] Yiqiao Li, Jianlong Zhou, Sunny Verma, and Fang Chen. 2022. A Survey of Explainable Graph Neural Networks: Taxonomy and Evaluation Metrics. *CoRR* abs/2207.12599 (2022). https://doi.org/10.48550/arXiv.2207.12599 arXiv:2207.12599

[10] Wanyu Lin, Hao Lan, and Baochun Li. 2021. Generative Causal Explanations for Graph Neural Networks. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 6666–6679.

[11] Wanyu Lin, Hao Lan, Hao Wang, and Baochun Li. 2022. OrphicX: A Causality-Inspired Latent Variable Model for Interpreting Graph Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 13719–13728. https://doi.org/10.1109/CVPR52688.2022.01336

[12] Ana Lucic, Maartje A. ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. 2022. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event (Proceedings of Machine Learning Research, Vol. 151)*. PMLR, 4499–4511.

[13] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized Explainer for Graph Neural Network. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, 19620–19631.

[14] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *CoRR* abs/1411.1784 (2014). arXiv:1411.1784

[15] Anh Mai Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 3387–3395.

[16] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional Image Synthesis with Auxiliary Classifier GANs. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 2642–2651.

[17] Subhankar Roy, Enver Sangineto, Nicu Sebe, and Begüm Demir. 2018. Semantic-Fusion Gans for Semi-Supervised Satellite Image Classification. In *2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018*. IEEE, 684–688.

[18] Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. 2021. Reinforcement Learning Enhanced Explainer for Graph Neural Networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 22523–22533.

[19] Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. 2016. Conditional Image Generation with PixelCNN Decoders. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 4790–4798.

[20] Abdul Waheed, Muskan Goyal, Deepak Gupta, Ashish Khanna, Fadi M. Al-Turjman, and Plácido Rogério Pinheiro. 2021. CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection. *CoRR* abs/2103.05094 (2021). arXiv:2103.05094

[21] Nikil Wale, Ian A. Watson, and George Karypis. 2008. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowl. Inf. Syst.* 14, 3 (2008), 347–375. https://doi.org/10.1007/s10115-007-0103-5

[22] Chenyu Wang, Zongyu Lin, Xiaochen Yang, Jiao Sun, Mingxuan Yue, and Cyrus Shahabi. 2022. HAGEN: Homophily-Aware Graph Convolutional Recurrent Network for Crime Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 4, 4193–4200. https://doi.org/10.1609/aaai.v36i4.20338

[23] Xiang Wang, Ying-Xin Wu, An Zhang, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2022. Reinforced Causal Explainer for Graph Neural Networks. *CoRR* abs/2204.11028 (2022). https://doi.org/10.48550/arXiv.2204.11028 arXiv:2204.11028

[24] Xiang Wang, Ying-Xin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. 2021. Towards Multi-Grained Explainability for Graph Neural Networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 18446–18458.

[25] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNN Explainer: A Tool for Post-hoc Explanation of Graph Neural Networks. *CoRR* abs/1903.03894 (2019). arXiv:1903.03894

[26] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, Virtual Event CA USA, 430–438. https://doi.org/10.1145/3394486.3403085

[27] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. ACM, 430–438. https://doi.org/10.1145/3394486.3403085

[28] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On Explainability of Graph Neural Networks via Subgraph Explorations. arXiv:2102.05152 [cs.LG]

[29] Zhanhong Zhou, Xiaolong Zhai, and Chung Tin. 2021. Fully automatic electrocardiogram classification system based on generative adversarial network with auxiliary classifier. *Expert Syst. Appl.* 174 (2021), 114809. https://doi.org/10.1016/j.eswa.2021.114809