Article

Reliable and Faithful Generative Explainers for Graph Neural Networks [†]

Yiqiao Li [1,*], Jianlong Zhou [1], Boyuan Zheng [1], Niusha Shafiabady [2] and Fang Chen [1]

[1] Data Science Institute, University of Technology Sydney, Sydney, NSW 2007, Australia; jianlong.zhou@uts.edu.au (J.Z.); boyuan.zheng@uts.edu.au (B.Z.); fang.chen@uts.edu.au (F.C.)
[2] Department of Information Technology, Australian Catholic University, North Sydney, NSW 2060, Australia; niusha.shafiabady@acu.edu.au
[*] Correspondence: yiqiao.li@uts.edu.au
[†] This paper is an extended version of our paper published in the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, UK, 21–25 October 2023.

Abstract: Graph neural networks (GNNs) have been effectively implemented in a variety of real-world applications, although their underlying work mechanisms remain a mystery. To unveil this mystery and advocate for trustworthy decision-making, many GNN explainers have been proposed. However, existing explainers often face significant challenges, such as the following: (1) explanations being tied to specific instances; (2) limited generalisability to unseen graphs; (3) potential generation of invalid graph structures; and (4) restrictions to particular tasks (e.g., node classification, graph classification). To address these challenges, we propose a novel explainer, GAN-GNNExplainer, which employs a generator to produce explanations and a discriminator to oversee the generation process, enhancing the reliability of the outputs. Despite its advantages, GAN-GNNExplainer still struggles with generating faithful explanations and underperforms on real-world datasets. To overcome these shortcomings, we introduce ACGAN-GNNExplainer, an approach that improves upon GAN-GNNExplainer by using a more robust discriminator that consistently monitors the generation process, thereby producing explanations that are both reliable and faithful. Extensive experiments on both synthetic and real-world graph datasets demonstrate the superiority of our proposed methods over existing GNN explainers.

Keywords: graph neural networks; explanations; generative methods; faithful; reliable

Citation: Li, Y.; Zhou, J.; Zheng, B.; Shafiabady, N.; Chen, F. Reliable and Faithful Generative Explainers for Graph Neural Networks. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 2913–2929. https://doi.org/10.3390/make6040139

Academic Editor: Weiping Ding

Received: 22 October 2024
Revised: 30 November 2024
Accepted: 11 December 2024
Published: 18 December 2024

1. Introduction

Graph neural networks (GNNs) have swiftly progressed as a powerful method for processing graph-structured data, showing outstanding performance across various real-world applications, including crime prediction [1], traffic flow estimation [2], event forecasting [3], and medical diagnosis [4]. GNNs are proficient in capturing intricate node relationships and extracting valuable features from graph data, making them an ideal option for tasks that require graph-based analysis.

Although GNNs demonstrate strong performance, their lack of explainability reduces their trustworthiness in key fields like healthcare and finance. The inherent black-box characteristic of GNNs complicates the comprehension of their decision-making mechanisms, making it challenging to uncover the reasoning behind their predictions and to detect potential biases. These challenges have restricted the wider adoption of GNNs in vital sectors where interpretability and transparency are essential, including healthcare [5], recommendation systems [6], and other areas.

To address this challenge, a multitude of GNN explainers have been proposed to shed light on the decision-making process of GNNs. These methods provide explanations at the node or graph level, helping to identify important graph structures and features that contribute to the model's predictions. Specifically, explaining GNN models is encouraged and even required to increase confidence in the GNN model's predictions, guarantee the security of real-world applications, and promote trustworthy artificial intelligence (AI) [7,8].

The explanation of GNN has attracted substantial scholarly interest, and many explainers [9–13] have been proposed over the past few years. Although these methods provide some useful explanations for complex GNN models, their practical application is hampered by their inherent constraints: (1) the explanation scale is tied to a specific instance; (2) the explanation cannot be easily generalised for unseen graphs; (3) the explanation may not be a valid graph; (4) the explanation may be limited to a specific task (e.g., node classification, graph classification, etc.). In particular, the seminal method GNNExplainer [9] limits itself to local explanation and lacks generalisability. After that, XGNN [13], which trains a graph generator to explain a class by displaying class-specific graph patterns, addressed the limitation of the explanation scale. However, it still lacks generalisability, and worse, it may generate some nonexisting important subgraphs. Recent Gem [12] has mitigated the limitations faced by previous methods, while its precision in explaining different tasks can vary significantly and lacks stability due to the inherent nature of the generation process.

To tackle the existing limitations, this paper introduces two novel GNN explainers, GAN-GNNExplainer and ACGAN-GNNExplainer [14], respectively, which use the generative method to produce explanations for GNNs. Both of our methods consist of a generator and a discriminator. Specifically, for GAN-GNNExplainer, the generator learns to produce explanations for the input graph G, which requires an explanation. Meanwhile, the discriminator distinguishes between "real" and generated explanations. The discriminator provides feedback to the generator, refining the explanation process. Through repeated interactions between the generator and discriminator, the generator eventually produces explanations that closely resemble the desired "real" ones. As a result, the quality of the explanations improves, leading to a significant boost in overall explanation accuracy. GAN-GNNExplainer represents a notable advancement in the accuracy of explanations, successfully addressing some limitations of current popular GNN explainers. However, GAN-GNNExplainer has inadequate reliability on real-world datasets and lacks fidelity.

To address these limitations, we introduce an enhanced method, ACGAN-GNNExplainer, which leverages the Auxiliary Classifier Generative Adversarial Network (ACGAN) [15] as its backbone to generate explanations for GNNs. Specifically, the input graph G, along with its corresponding label $f(G)$, determined by the target GNN model f, is fed into the generator, which then learns to generate explanations. To ensure the validity and accuracy of the generated subgraph, a discriminator is incorporated. The discriminator distinguishes between "real" and generated explanations, assigns a prediction label to each explanation, and provides feedback to the generator, overseeing the entire generation process. Extensive experiments on both synthetic and real-world datasets demonstrate the effectiveness of our method, showcasing its superiority over existing GNN explainers.

Key contributions of this paper include the following:

- We propose a novel explainer called GAN-GNNExplainer, specifically tailored for GNN models. This approach employs a generator to generate explanations and is supervised by a discriminator, ensuring reliable results throughout the procedure.
- Additionally, we introduce ACGAN-GNNExplainer, a more advanced explainer for GNN models. It leverages both a generator and a discriminator, which consistently oversees the procedure, leading to explanations that are both reliable and faithful.
- Our methods are comprehensively evaluated across various graph datasets, spanning both synthetic and real-world data, and across multiple tasks, including node classification and graph classification. The outcomes consistently highlight the advantages of our approach over existing methods.

2. Related Work

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) [16] are composed of two main components: a generator and a discriminator, both of which are trained concurrently in a competitive setup. The generator begins with random noise and learns to create synthetic samples that closely resemble the real data distribution, while the discriminator distinguishes between genuine data and synthetic outputs generated by the generator. During the training process, the generator aims to produce increasingly realistic outputs, making it progressively more difficult for the discriminator to differentiate them accurately. This adversarial mechanism has allowed GANs to make remarkable advancements in multiple fields, including image synthesis, data augmentation, and cross-modal tasks.

Over time, a variety of GAN extensions have been developed, addressing issues such as training stability and mode collapse, while also improving the diversity and fidelity of the generated outputs. These innovations involve alterations to network architectures, loss functions, and optimisation strategies. For instance, conditional GANs (CGANs) [17] introduce a conditioning mechanism, where additional information (such as class labels) is provided to both the generator and discriminator, enabling the generation of class-specific samples. This conditional setup allows for more targeted generation tasks, improving sample diversity and applicability.

In another line of work, models such as InfoGAN [18] explore the disentanglement of latent variables. By optimising the mutual information between a portion of the latent variables and the generated samples, InfoGAN gains the ability to control distinct features of the generated data, thereby improving interpretability. This introduces an additional layer of control over the generation process, making it possible to manipulate distinct attributes of the samples, such as object orientation or style.

Building on the idea of conditioning, the Auxiliary Classifier Generative Adversarial Networks (ACGAN) [15] introduces an auxiliary classification objective to further enhance the generative process. ACGAN incorporates class labels into the generation process, with the discriminator tasked not only with distinguishing between real and synthetic data but also with classifying the samples according to their respective categories. This dual objective improves both the quality of the generated samples and their relevance to the given class labels. As a result, ACGAN has found applications in scenarios requiring fine-grained control over the generation process, such as in medical imaging [19] and other domain-specific tasks [20].

These advancements have significantly expanded the scope and capability of GAN models, making them versatile tools for a variety of practical applications, from creative tasks like art generation to critical areas such as healthcare and security. The continuous evolution of GAN architectures and techniques ensures their relevance in tackling increasingly complex data generation challenges.

2.2. Graph Neural Networks

GNNs represent a robust class of deep learning models crafted to handle graph-structured data, including social networks, citation networks, and molecular structures. In contrast to traditional neural networks, which process information in vector or matrix formats, GNNs work directly on graph data by collecting and integrating information from adjacent nodes and edges. GNNs have demonstrated outstanding results across multiple tasks, such as node classification [21], graph classification [22], and link prediction [23].

Beyond their strong theoretical grounding, GNNs are widely utilised in practical settings. For instance, Wang et al. [24] introduced a homophily-based constraint to refine the optimisation of region graphs for crime prediction. This method encourages neighbouring region nodes in the graph to exhibit similar crime patterns, aligned with the diffusion convolution framework. GNNs are also employed in traffic prediction [24] and medical diagnosis [25], demonstrating their adaptability in real-world applications.

Similar to many other deep learning models, GNNs face a notable limitation: they are frequently regarded as black-box systems, lacking explanations that are comprehensible to humans. Without a thorough understanding and verification of the internal mechanisms of GNNs, their application in critical areas involving fairness, privacy, and safety is hindered. Hence, the development of explainable GNN models has become a crucial research area.

2.3. Graph Neural Networks Explainers

Explaining the reasoning behind GNNs is a critical yet complex task, as it directly contributes to improving the explainability, trustworthiness, and safety of these models. In recent years, a range of methods have emerged to tackle this issue, leveraging the distinctive structural and relational features of graphs to produce meaningful explanations. Below, we outline several key approaches that have significantly advanced this area of research.

GNNExplainer [9] is one of the foundational methods developed for explaining GNNs, focusing on identifying the critical substructures and node features that drive a model's prediction. This technique provides instance-specific explanations, offering insights into how local patterns in the graph influence individual decisions. PGExplainer [11] extends this by generating probabilistic explanations that generalise across multiple instances. Unlike GNNExplainer, it operates at a model-wide level, making it adaptable to diverse scenarios.

Further advancing the field, the authors in [12] introduce a generative approach, Gem, which can offer both local and global explanations. Its inductive nature allows it to function without the need for retraining the GNN, providing greater flexibility in real-time applications. OrphicX [26] builds on this concept by offering causal explanations, concentrating on latent factors to deliver a more profound understanding of the cause-and-effect dynamics influencing GNN predictions. However, despite their promising contributions, both Gem and OrphicX encounter challenges when applied to real-world datasets, particularly in maintaining the accuracy of their explanations. However, in our paper, we seek to overcome these challenges by introducing novel explainers that can provide high-fidelity explanations across both synthetic and real-world datasets.

In addition to these methods, reinforcement learning has also been explored as a tool for explaining GNNs. For instance, XGNN [13] is a model-level explainer that employs a graph generator to discover patterns enhancing the model's predictive capabilities, thereby uncovering important graph structures. Another notable approach, RC-Explainer [27], employs causal analysis combined with a reinforcement learning framework to uncover causal dependencies in GNN predictions. Moreover, RG-Explainer [28] further enhances this by using reinforcement learning to generate explanations that generalise well in inductive settings, showcasing robust performance across diverse applications.

In parallel, another line of research focuses on generating counterfactual explanations, which offer alternative scenarios to explain the model's behaviour. CF-GNNExplainer [29] stands out for producing counterfactual explanations for a majority of GNN instances, thereby highlighting the key features that would change the outcome of a prediction. Similarly, RCExplainer [30] generates robust counterfactual explanations, while ReFine [31] adopts a multi-grained strategy, incorporating pretraining and fine-tuning to improve the precision and detail of its explanations.

3. Method

3.1. Problem Formulation

Interpretation and explanation are crucial for gaining insights into the inner workings of GNNs. While interpretation aims to uncover the model's decision-making process, emphasising the transparency and traceability of decisions, explanation supports GNN predictions by providing a logical and coherent rationale for the observed outcomes.

In this paper, we focus on identifying the subgraphs that significantly influence GNN predictions. A graph is represented as $G = (V, A, X, L)$, where V denotes the set of nodes, $A \in 0, 1$ is the adjacency matrix with $Aij = 1$ indicating an edge between nodes i and j, and $Aij = 0$ otherwise. X is the feature matrix of graph G, and L represents the class label. Let f denote the GNN model, such that $f(G) \to Y$.

We define $E(f(G), G) \to G^s$ as the explanation generated by a GNN explainer. Ideally, when this explanation is provided as input to the GNN model f, it should yield the same prediction Y, implying $f(G) = f(E(f(G), G))$. Furthermore, the explanation $E(f(G), G) \to G^s$ should represent a valid subgraph of the original graph G, meaning $G^s \subseteq G$.

3.2. Obtaining Causal Real Explanations

The objective of this paper is to uncover the underlying rationale behind the predictions made by the target GNN model f. Instead of delving into the inner workings of f, we treat it as a black box, focusing on identifying the subgraphs that significantly affect its predictions. To achieve this, we employ a generative model capable of autonomously generating relevant subgraphs or explanations. For the generative model to produce precise explanations, it requires training with "real" or ground-truth data. However, such data are often unavailable in practice. To address this limitation, we utilise Granger causality [32], a widely adopted method to assess whether one variable exerts a causal influence on another, enabling the generation of meaningful and reliable explanations.

In our experiments, individual edges are selectively masked, and their influence on the target predictions of the GNN model is assessed. By comparing the prediction probabilities of the original and masked graphs, we quantify the impact of each edge on the prediction of the model by assigning weights based on the observed differences. These weights are then used to rank the edges, with the most critical ones representing the most significant explanations (i.e., important subgraphs). However, it is important to note that directly applying Granger causality to explain a GNN model f can be both computationally expensive and limited in terms of generalisation. Our approach addresses this by using a parameterised explainer that identifies shared patterns across similar graphs. Once these patterns are learned, the explainer can be transferred to other graphs, leading to enhancements in both efficiency and scalability.

3.3. GAN-GNNExplainer

In this paper, we introduce GAN-GNNExplainer, a GAN-based explanation method for GNNs that leverages the generative capabilities of GANs. The model comprises two components: a generator ($\mathcal{G}_1$) and a discriminator ($\mathcal{D}_1$), as illustrated in Figure 1.

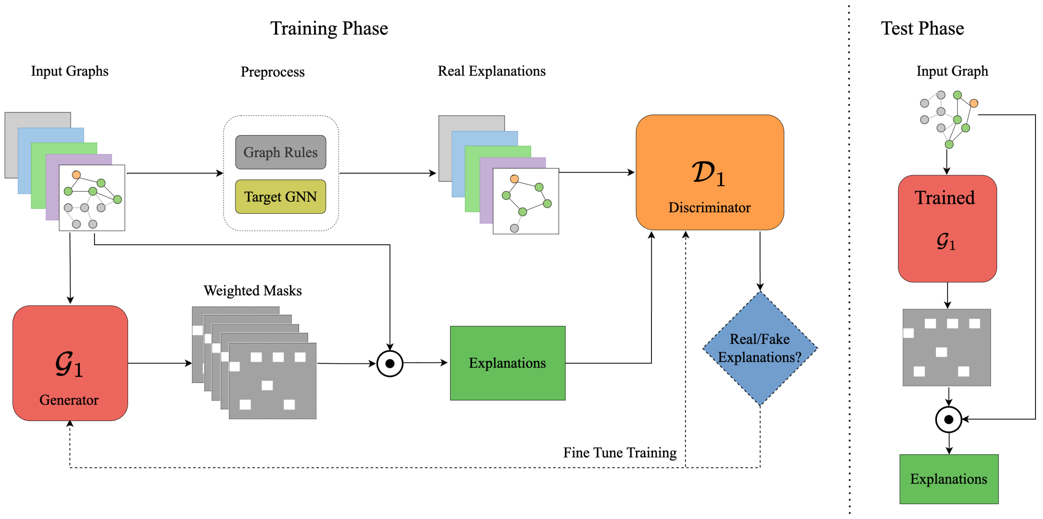

Figure 1. The framework of GAN-GNNExplainer. The $\odot$ symbol represents element-wise multiplication. The framework has two stages: Training and Testing. During the Training Phase, the goal is to optimise the generator and discriminator components of the GAN-GNNExplainer model. After successful training, the Testing Phase uses the trained generator to produce explanations for test data.

Unlike the typical way of training a GAN where random noise z is fed into the generator $\mathcal{G}$, in our model, we feed $\mathcal{G}$ with the original graph G which is the graph we want to explain. Doing so ensures that the generator $\mathcal{G}$ provides a corresponding explanation to the original input graph G. In addition, the generator $\mathcal{G}$ trained under this mechanism can be easily generalised to unseen graphs without significant retraining and thus can save computational cost. For our $\mathcal{G}$, we employ an encoder–decoder network where the encoder projects the original input graph G into a compact hidden representation and then the decoder reconstructs the explanation from the compact hidden representation. In our case, the reconstructed explanation is a mask indicating the significance of each edge. When we conduct experiments on synthetic datasets, we have a six-layer encoder and a two-layer decoder; when we experiment with real-world datasets, we keep the decoder complexity but slightly increase the complexity of the encoder. Thus, we end with a seven-layer decoder.

In principle, $\mathcal{G}_1$ can generate both valid and invalid explanations, which may conflict with the goal of accurately explaining a GNN. To regulate the generation process, a discriminator $\mathcal{D}_1$ is introduced. $\mathcal{D}_1$ acts as a graph classifier, receiving both the "real" and generated explanations generated by the explainer. Its role is to differentiate between the "real" and generated explanations, ensuring that the generator produces reliable outputs.

To train $\mathcal{G}_1$ and $\mathcal{D}_1$, we first need to obtain the "real" explanations. This is performed through a preprocessing step in our framework (Figure 1), where Granger causality generates the "real" explanations as ground truth for training the discriminator. Details of this process can be found in Section 3.2. Once the input graph G and corresponding subgraph are identified, the model is trained to generate a weighted mask that emphasises the important edges and nodes in G that play a key role in the decision-making process of the GNN model f. By applying this weighted mask to the adjacency matrix, we extract the relevant explanations or key subgraphs. These explanations are essential for understanding the reasoning behind the complex predictions made by the GNN model.

In a GAN framework, the generator ($\mathcal{G}$) and discriminator ($\mathcal{D}$) engage in a minimax game, competing against each other. The generator learns to mimic the underlying distribution of training data and generates "fake" samples that deceive the discriminator into treating them as real. The objective of this minimax game is defined in Equation (1):

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \quad \mathbb{E}_{G^{gt} \sim p_{(G^{gt})}}[\log \mathcal{D}(G^{gt})] + \mathbb{E}_{G \sim p_{(G)}}[\log(1 - \mathcal{D}(\mathcal{G}(G)))], \tag{1}$$

where G represents the original graph requiring explanation and G^{gt} refers to its ground-truth explanation (e.g., the significant subgraph).

When we simply adopt Equation (1) as our objective function to train our $\mathcal{G}_1$ and $\mathcal{D}_1$ simultaneously, we empirically observe that the accuracy of the final explanation is not optimistic. We suppose it is because Equation (1) does not explicitly encode the information of the accuracy of the explanation from a target GNN model. To address this issue and improve the precision of the explanation, we then explicitly incorporate the accuracy of the explanation into our objective function and obtain an improved GAN-based loss function defined in Equation (2):

$$\begin{aligned} \min_{\mathcal{G}_1} \max_{\mathcal{D}_1} \quad & \mathbb{E}_{G^{gt} \sim p_{(G^{gt})}}[\log \mathcal{D}_1(G^{gt})] \\ & + \mathbb{E}_{G \sim p_{(G)}}[\log(1 - \mathcal{D}_1(\mathcal{G}_1(G)))] \\ & + \lambda \frac{1}{N} \sum_{i=1}^{N} (f(G) - f(\mathcal{G}_1(G)))^2, \end{aligned} \tag{2}$$

where f denotes a pre-trained target GNN model, N represents the count of node set of G, and G is the input graph we aim to explain, while G^{gt} is its corresponding ground-truth explanation (e.g., the important subgraph). The parameter λ is a trade-off hyperparameter that balances the influence of the GAN model and the explanation accuracy derived from the pre-trained target GNN f. If λ is set to zero, Equation (2) becomes identical to Equation (1).

As highlighted in Section 1, GAN-GNNExplainer represents a notable advancement in the field of GNN explainability, effectively addressing some of the limitations found in existing popular GNN explainers. Nonetheless, there are still several challenges that warrant further exploration, particularly its limited reliability on real-world datasets and insufficient fidelity.

Therefore, we focus on developing an enhanced model in Section 3.4, ACGAN-GNNExplainer, which incorporates the predicted labels from the target GNN into the explanation generation process. This enhancement is designed to improve its performance on real-world datasets, making the explanations both more reliable and more faithful.

3.4. ACGAN-GNNExplainer

To address the limitations of GAN-GNNExplainer, we introduce ACGAN-GNNExplainer, which also comprises a generator ($\mathcal{G}_2$) and a discriminator ($\mathcal{D}_2$). Similarly, $\mathcal{G}_2$ generates explanations, while $\mathcal{D}_2$ oversees the generation process. The detailed framework of ACGAN-GNNExplainer is shown in Figure 2.

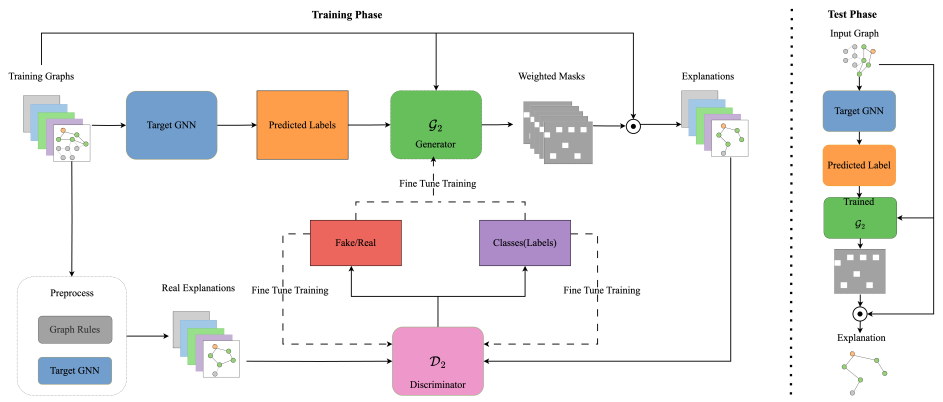

Figure 2. The framework of ACGAN-GNNExplainer. The symbol $\odot$ denotes element-wise multiplication. The framework consists of two phases: the Training Phase and the Testing Phase. During the Training Phase, both the generator and discriminator of the ACGAN-GNNExplainer model are trained. Once training is complete, the Testing Phase uses the trained generator to produce explanations for the test data.

In ACGAN-GNNExplainer, the generator $\mathcal{G}_2$ is provided with both the original graph G and the predicted label Y, which is generated by the target GNN model f. This method ensures that the explanation produced by $\mathcal{G}_2$ is crucial for understanding the predictions made by f, as it directly relates to the input graph G. Moreover, the generator $\mathcal{G}_2$, trained using this method, can generalise to unseen graphs without requiring significant retraining, thereby reducing computational costs.

The generator follows an encoder–decoder architecture, where the encoder compresses the input graph G into a compact hidden representation, and the decoder reconstructs the explanation from this latent space. In this context, the explanation takes the form of a mask matrix that highlights the importance of each edge.

Conceptually, the discriminator $\mathcal{D}_2$ oversees the generation process of $\mathcal{G}_2$. It is provided with both the "real" and generated explanations from $\mathcal{G}_2$, determining whether the explanation is "real" or generated while classifying it. This classification feedback encourages $\mathcal{G}_2$ to improve explanation accuracy and faithfulness. Preprocessing of training graphs is necessary to obtain "real" explanations, guiding $\mathcal{D}_2$ during the training phase in the ACGAN-GNNExplainer framework.

The generator $\mathcal{G}_2$ produces explanations or subgraphs $G^s \subseteq G$ based on two key inputs: the original graph G and the predicted label Y, expressed as $G^s \leftarrow \mathcal{G}_2(G, Y)$. Simultaneously, the discriminator $\mathcal{D}_2$ evaluates both the origin probability $P(S \mid G)$ (whether "real" or generated) and the probability of class classification $P(Y \mid G)$, where Y represents the predicted label of the graph G, denoted as $f(G) \rightarrow Y$. The loss function of the discriminator consists of two components: the likelihood of the correct source $\mathcal{L}_S$, as defined in Equation (3), and the likelihood of the correct class $\mathcal{L}_Y$, as defined in Equation (4):

$$\mathcal{L}_S = \mathbb{E}[\log P(S = \text{ real} \mid G)] + \mathbb{E}[\log P(S = \text{ generated } \mid G^s)], \tag{3}$$

$$\mathcal{L}_Y = \mathbb{E}[\log P(Y = L \mid G)] + \mathbb{E}[\log P(Y = L \mid G^s)]. \tag{4}$$

where G means the original graph that requires an explanation, and L means its class label.

The discriminator $\mathcal{D}_2$ and generator $\mathcal{G}_2$ engage in a minimax game, competing with each other. The primary goal of $\mathcal{D}_2$ is to maximise the probability of correctly distinguishing between "real" and generated graphs ($\mathcal{L}_S$) while also accurately predicting the class label ($\mathcal{L}_Y$) for all graphs. This leads to a combined objective of maximising $(\mathcal{L}_S + \mathcal{L}_Y)$.

Conversely, the generator $\mathcal{G}_2$ seeks to minimise the ability of $\mathcal{D}_2$ to distinguish between "real" and generated graphs while simultaneously maximising its capacity to classify them correctly. This results in a combined objective of maximising $(-\mathcal{L}_S + \mathcal{L}_Y)$. Therefore, based on Equations (3) and (4), the objective functions for $\mathcal{D}_2$ and $\mathcal{G}_2$ are given in Equation (5) and Equation (6), respectively:

$$\begin{aligned}\mathcal{L}_{(\mathcal{D}_2)} = &-\mathbb{E}_{G^{gt}\sim P(G^{gt})}\log \mathcal{D}_2(G^{gt}) \\ &-\mathbb{E}_{G\sim P(G)}\log[1-\mathcal{D}_2(\mathcal{G}_2(G,L))] \\ &-\mathbb{E}_{G^{gt}\sim P(G^{gt})}P(Y \mid G^{gt}) \\ &-\mathbb{E}_{G\sim P(G)}\log(P(Y \mid \mathcal{G}_2(G,L)),\end{aligned} \tag{5}$$

$$\begin{aligned}\mathcal{L}_{(\mathcal{G}_2)} = &-\mathbb{E}_{G\sim P(G)}\log \mathcal{D}_2(\mathcal{G}_2(G,L)) \\ &-\mathbb{E}_{G\sim P(G)}\log P(Y \mid \mathcal{G}_2(G,L)),\end{aligned} \tag{6}$$

where G represents the original graph that requires an explanation, while G^{gt} signifies its corresponding actual explanation (e.g., the "real" important subgraph).

Using the objective functions from Equations (5) and (6) to train $\mathcal{D}_2$ and $\mathcal{G}2$, we observe that the fidelity of the generated explanations is unsatisfactory. This may be because the generator loss $\mathcal{L}(\mathcal{G}_2)$, as defined in Equation (6), does not explicitly consider fidelity information from the target GNN model f. To resolve this and improve both fidelity and accuracy, we incorporate fidelity directly into the generator's objective function. Consequently, we derive an enhanced loss function for $\mathcal{G}_2$, as shown in Equation (7):

$$\begin{aligned}\mathcal{L}_{(\mathcal{G}_2)} = &-\mathbb{E}_{G\sim P(G)}\log \mathcal{D}_2(\mathcal{G}_2(G,L)) \\ &-\mathbb{E}_{G\sim p(G)}\log P(Y \mid \mathcal{G}_2(G,L)) \\ &+\lambda\mathcal{L}_{Fid},\end{aligned} \tag{7}$$

$$\mathcal{L}_{Fid} = \frac{1}{N}\sum_{i=1}^{N}||f(G) - f(\mathcal{G}_2(G))||^2, \tag{8}$$

where $\mathcal{L}_{Fid}$ 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, 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 (7) being precisely equivalent to Equation (6).

4. Experiments

In this section, we thoroughly evaluate the performance of our proposed methods, GAN-GNNExplainer (see Section 4.2) and ACGAN-GNNExplainer (see Section 4.3). We begin by describing the datasets used in our experiments and outlining the implementation details in Section 4.1. Next, we present a comparative analysis of our methods against other state-of-the-art GNN explainers, assessing their effectiveness on both synthetic and real-world datasets.

4.1. Experimental Settings

Datasets. We focus on two commonly used synthetic node classification datasets, BA-Shapes and Tree-Cycles [9], as well as two real-world graph classification datasets, Mutagenicity [33] and NCI1 [34]. Detailed descriptions of datasets are provided in Table 1.

The BA-Shapes dataset comprises a Barabasi–Albert (BA) graph with 300 nodes. It incorporates 80 "house"-structured network motifs randomly attached to nodes within the base graph. Nodes are classified into four categories based on their structural roles: those at the top, middle, and bottom of houses and those not part of any house.

The Tree-Cycles dataset originates from an initial eight-level balanced binary tree. It incorporates 80 six-node cycle motifs attached randomly to nodes within the base graph. Nodes are divided into two classes based on whether they belong to the tree or the cycle.

The Mutagenicity datasets consist of 4337 molecule graphs representing atoms as nodes and chemical bonds as edges. These graphs are categorised into two classes, nonmutagenic and mutagenic, indicating their effects on the Gram-negative bacterium Salmonella Typhimurium. Specifically, carbon rings containing NH_2 or NO_2 groups are known to be mutagenic. However, carbon rings are present in both mutagenic and nonmutagenic graphs, rendering them nondiscriminative.

NCI1 is a curated subset of chemical compounds evaluated for their efficacy against non-small-cell lung cancer. It encompasses over 4000 compounds, each tagged with a class label indicating positive or negative activity. Each compound is depicted as an undirected graph, with nodes representing atoms, edges denoting chemical bonds, and node labels indicating atom types.

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

	Node Classification		**Graph Classification**	
	BA-Shapes	**Tree-Cycles**	**Mutagenicity**	**NCI1**
# of graphs	1	1	4337	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. With the rising adoption of GNNs in various real-world applications, the need for explainability has gained significant attention, as it plays a crucial role in enhancing model transparency and building user trust. In this context, we selected three prominent GNN explanation methods for comparison: GNNExplainer [9], Gem [12], and OrphicX [26]. For these methods, we utilised their official implementations to ensure consistency in evaluation.

Different top edges (K or R). After calculating the importance (or weight) of each edge in the input graph G, selecting an appropriate number of edges for the explanation is crucial. Choosing too few edges may result in incomplete explanations, while selecting too many can introduce noise. To address this, we define a top K for synthetic datasets and a top ratio (R) for real-world datasets to determine the number of edges to include in the explanation. We evaluate the stability of our method by experimenting with different values of K and R. Specifically, we use $K = \{5, 6, 7, 8, 9\}$ 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 the real-world datasets.

Data split. To ensure consistency and fairness in our experiments, we split the data into three subsets: 80% for training, 10% for validation, and 10% for testing. The testing data are kept completely separate and unused until the final evaluation stage.

Evaluation metrics. An effective GNN explainer should produce concise explanations or subgraphs while preserving the model's predictive accuracy when these explanations are input back into the target GNN. Therefore, it is essential to assess the performance of the explainer using multiple evaluation metrics [35]. In our experiments, we evaluate the accuracy of the GAN-GNNExplainer and assess both the accuracy and fidelity of the ACGAN-GNNExplainer.

Specifically, we generate explanations for the test set using GNNExplainer [9], Gem [12], OrphicX [26], GAN-GNNExplainer, and ACGAN-GNNExplainer. These explanations are then fed into the pre-trained target GNN model f to evaluate the accuracy, which is formally defined in Equation (9):

$$ACC_{exp} = \frac{|f(G) = f(G^s)|}{|T|}, \tag{9}$$

where G represents the original graph requiring explanation and G^s refers to its corresponding explanation (such as the significant subgraph). The term $|f(G) = f(G^s)|$ denotes the number of instances where the predictions of the target GNN model f on both G and G^s are identical, while $|T|$ is the total number of instances.

Furthermore, fidelity assesses how accurately the generated explanations capture the key subgraphs of the original input graph. In our experiments, we utilise the metrics $Fidelity^+$ and $Fidelity^-$ [36] to evaluate the fidelity of the explanations.

$Fidelity^+$ measures the change in prediction accuracy when the key input features are excluded, comparing the original predictions with those generated using the modified graph. Conversely, $Fidelity^-$ evaluates the variation in prediction accuracy when the important features are retained and nonessential structures are removed. Together, $Fidelity^+$ and $Fidelity^-$ offer a comprehensive assessment of how well the explanations capture the model's behaviour and the significance of various input features. The mathematical definitions of $Fidelity^+$ and $Fidelity^-$ are provided in Equation (10) and Equation (11), respectively:

$$Fid^+ = \frac{1}{N}\sum_{i=1}^{N}(f(G_i)_{L_i} - f(G_i^{1-s})_{L_i}), \tag{10}$$

$$Fid^- = \frac{1}{N}\sum_{i=1}^{N}(f(G_i)_{L_i} - f(G_i^{s})_{L_i}), \tag{11}$$

where N represents the total number of samples and L_i denotes the class label for instance i. The terms $f(Gi)L_i$ and $f(Gi^{1-s})L_i$ refer to the prediction probabilities for class L_i based on the original graph G_i and the occluded graph G_i^{1-s}, respectively. The occluded graph is created by removing the important features (explanations) identified by the explainers from the original graph. A higher $Fidelity^+$ value is preferred, indicating a more critical explanation. On the other hand, $f(Gi^s)L_i$ refers to the prediction probability for class L_i using the explanation graph G_i^s, which contains the crucial structures identified by the explainers. A lower $Fidelity^-$ value is desirable as it reflects a more complete and sufficient explanation.

In summary, the accuracy of the explanation (ACC_{exp}) evaluates how well the generated explanations reflect the model's predictions, while $Fidelity^+$ and $Fidelity^-$ measure the necessity and sufficiency of these explanations, respectively. By comparing the accuracy and fidelity metrics across different explainers, we can gain meaningful insights into the effectiveness and suitability of each method.

4.2. Evaluation GAN-GNNExplainer

4.2.1. Results on Synthetic Datasets

We commence our experimental analysis by evaluating our proposed method on two well-known synthetic datasets: BA-Shapes and Tree-Cycles [9]. Comprehensive details about these datasets are provided in Section 4.1. To assess the effectiveness of our GAN-GNNExplainer, we compare its performance with that of existing explainers, namely, GNNExplainer and Gem. The accuracy results for different values of K are presented in Table 2 for the BA-Shapes dataset and Table 3 for the Tree-Cycles dataset, respectively.

After analysing the results for the BA-Shapes dataset, as presented in Table 2, we find that our GAN-GNNExplainer consistently obtains the most accurate explanations in most cases. Although GNNExplainer, Gem, and GAN-GNNExplainer demonstrate strong performance on synthetic datasets, GAN-GNNExplainer incorporates several key enhancements that further improve its effectiveness. Moreover, on the Tree-Cycles dataset, as shown in Table 3, GAN-GNNExplainer significantly outperforms the baselines.

It is worth noting that in our experiments on the relatively simple dataset, BA-Shapes, our model GAN-GNNExplainer demonstrates improved accuracy as K increases, effectively leveraging the additional information. This distinction becomes particularly important when applied to more complex datasets. Although our model may require more information to achieve optimal performance in such cases, it has the potential to surpass Gem by delivering higher accuracy.

Table 2. Explanation accuracy on the BA-Shapes dataset. The presented outcomes encompass averages from five runs. Note that the best-performing results have been emphasised in bold.

K (Edges)	**5**	**6**	**7**	**8**	**9**
GNNExplainer	0.7941	0.8824	0.9118	0.9118	0.9118
Gem	**0.9412**	0.9412	0.9412	0.9412	**0.9412**
GAN-GNNExplainer	0.6764	**0.9706**	**0.9706**	**0.9706**	**0.9412**

Table 3. Explanation accuracy on the Tree-Cycles dataset. The presented outcomes encompass averages from five runs. Note that the best-performing results have been emphasised in bold.

K (Edges)	**6**	**7**	**8**	**9**	**10**
GNNExplainer	0.2000	0.5429	0.7143	0.8571	0.9429
Gem	0.7142	0.8285	0.5714	0.8285	0.9428
GAN-GNNExplainer	**0.9429**	**0.9715**	**0.9429**	**1.0000**	**1.0000**

4.2.2. Results on Real-World Datasets

This subsection reports the experimental results with real-world datasets. The quantitative evaluation is shown in Tables 4 and 5. As shown in the table, the reported results successfully demonstrate that the proposed GAN-GNNExplainer can generate explanations with consistently high accuracy across all datasets compared with other explainers.

In the case of the Mutagenicity datasets, our proposed method outperformed Gem only when $R = 0.7$. However, for the NCI1 datasets, our method showed better accuracy compared to Gem across most R values. The results of the real-world datasets align with those of the BA-Shapes dataset, suggesting that when dealing with complex data, additional information is necessary to generate accurate explanations. Furthermore, our findings indicate that our approach has the potential to achieve higher accuracy than Gem when provided with more information.

Table 4. Explanation accuracy on the Mutagenicity dataset. The presented outcomes encompass averages from five runs. Note that the best-performing results have been emphasised in bold.

R (Edge Ratio)	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
GNNExplainer	**0.6175**	0.5968	0.6313	0.6935	0.7811
Gem	0.5737	**0.6014**	0.6590	**0.7235**	**0.7903**
GAN-GNNExplainer	0.5914	0.5956	**0.6929**	0.7215	0.7598

Table 5. Explanation accuracy on the NCI1 dataset. The presented outcomes encompass averages from five runs. Note that the best-performing results have been emphasised in bold.

R (Edge Ratio)	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
GNNExplainer	0.5961	0.6107	0.6788	**0.7616**	0.8127
Gem	0.5645	0.6083	0.6837	0.7518	**0.8321**
GAN-GNNExplainer	**0.6375**	**0.6496**	**0.7105**	**0.7616**	0.7762

4.3. Evaluation ACGAN-GNNExplainer

4.3.1. Results on Synthetic Datasets

Firstly, we conduct experiments on synthetic datasets, including BA-Shapes and Tree-Cycles. We evaluate the performance of explanations provided by GNNExplainer, Gem, OrphicX, and ACGAN-GNNExplainer (our model). The performance of explanations for synthetic datasets with various K settings is detailed in Tables 6 and 7.

Table 6. The results of explanations on BA-Shapes dataset: $ACC_{exp}(\uparrow)$, $Fid^{+}(\uparrow)$, $Fid^{-}(\downarrow)$. The presented outcomes encompass averages from five runs. Note that the best-performing results have been emphasised in bold. In this table, "Our Method" refers to our proposed method ACGAN-GNNExplainer.

K	Metrics	GNNExplainer	Gem	OrphicX	Our Method
5	Acc_{exp}	0.7941	**0.9412**	0.7353	0.7941
	Fid^{+}	0.7059	0.5588	**0.7941**	0.6471
	Fid^{-}	0.1471	**0.000**	0.2059	0.1471
6	Acc_{exp}	0.8824	**0.9706**	0.7353	0.8529
	Fid^{+}	0.6765	0.5588	**0.7941**	0.5882
	Fid^{-}	0.0588	−**0.0294**	0.2059	0.0882
7	Acc_{exp}	0.9118	**0.9706**	0.8529	**0.9706**
	Fid^{+}	0.7059	0.5882	**0.7941**	0.6176
	Fid^{-}	0.0294	−**0.0294**	0.0882	−**0.0294**
8	Acc_{exp}	0.9412	**0.9706**	0.8824	**0.9706**
	Fid^{+}	0.7353	0.5882	**0.7941**	0.6471
	Fid^{-}	0.000	−**0.0294**	0.0588	−**0.0294**
9	Acc_{exp}	0.9118	0.9706	0.8824	**1.000**
	Fid^{+}	0.7353	0.5882	**0.7941**	0.6471
	Fid^{-}	0.0294	−0.0294	0.0588	−**0.0588**

Table 7. The results of explanations on Tree-Cycles dataset: $ACC_{exp}(\uparrow)$, $Fid^{+}(\uparrow)$, $Fid^{-}(\downarrow)$. The presented outcomes encompass averages from five runs. Note that the best-performing results have been emphasised in bold. In this table, "Our Method" refers to our proposed method ACGAN-GNNExplainer.

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

As shown in Table 6, none of the models consistently outperforms the others across all metrics for the BA-Shapes dataset. However, with increasing values of K, ACGAN-GNNExplainer exhibits progressively stronger explanation accuracy (ACC_{exp}) and demonstrates enhanced performance in terms of $Fidelity^{-}$. In contrast, OrphicX consistently achieves higher $Fidelity^{+}$ values across various K, indicating its effectiveness in identifying essential subgraphs. Nevertheless, its lower performance in ACC_{exp} and $Fidelity^{-}$ suggests that it struggles to deliver comprehensive and accurate explanations.

The results in Table 7 show that all methods perform comparably well on the Tree-Cycles dataset across various K values. However, none of them consistently surpasses others across all evaluation metrics, following the observed pattern in the BA-Shapes dataset (see Table 6). Notably, in the range of $K = \{6, 7\}$, our method (ACGAN-GNNExplainer) stands out as the most effective option, exhibiting the highest fidelity across all K values; while OrphicX surpasses ACGAN-GNNExplainer in terms of $Fidelity^{-}$ and explanation accuracy (ACC_{exp}) when K falls within the range of $\{8, 9, 10\}$, ACGAN-GNNExplainer continues to demonstrate solid performance.

Overall, all GNN explainers perform well on synthetic datasets, which is largely attributed to the relative simplicity of these datasets compared to real-world scenarios. Notably, ACGAN-GNNExplainer consistently outperforms other methods in many cases. Even when it does not outperform its competitors, ACGAN-GNNExplainer remains highly competitive. To provide a more comprehensive evaluation, we extend our analysis to real-world datasets in Section 4.3.2 for a more in-depth assessment.

4.3.2. Results on Real-World Datasets

To further validate our method, we conducted experiments on two widely used real-world datasets: Mutagenicity [33] and NCI1 [34]. The results of these experiments are presented in Table 8 for the Mutagenicity dataset and Table 9 for the NCI1 dataset.

Table 8. The results of explanations on Mutagenicity dataset: $ACC_{exp}(\uparrow)$, $Fid^{+}(\uparrow)$, $Fid^{-}(\downarrow)$. The presented outcomes encompass averages from five runs alongside their corresponding standard deviations. Note that the best-performing results have been emphasised in bold. In this table, "Our Method" refers to our proposed method ACGAN-GNNExplainer.

R	Metrics	GNNExplainer	Gem	OrphicX	Our Method
0.5	Acc_{exp}	**0.6175**	0.5737	0.4539	**0.6175**
	Fid^{+}	0.3618	0.3018	0.2419	**0.3963**
	Fid^{-}	**0.2535**	0.2972	0.4171	**0.2535**
0.6	Acc_{exp}	0.5968	0.6014	0.5599	**0.6037**
	Fid^{+}	0.3825	0.3295	0.2949	**0.3828**
	Fid^{-}	0.2742	0.2696	0.3111	**0.2673**
0.7	Acc_{exp}	0.6313	0.659	0.6244	**0.7074**
	Fid^{+}	0.3963	0.2857	0.2995	**0.3986**
	Fid^{-}	0.2396	0.212	0.2465	**0.1636**
0.8	Acc_{exp}	0.6935	0.7235	0.7097	**0.7673**
	Fid^{+}	**0.3641**	0.2581	0.3157	0.3602
	Fid^{-}	0.1774	0.1475	0.1613	**0.1037**
0.9	Acc_{exp}	0.7811	0.7903	**0.8111**	0.7903
	Fid^{+}	0.3641	0.212	0.2949	**0.3871**
	Fid^{-}	0.0899	0.0806	**0.0599**	0.0806

As shown in Table 8, ACGAN-GNNExplainer demonstrates superior performance in both fidelity and explanation accuracy (ACC_{exp}) across most settings where R ranges from 0.5 to 0.8. Although OrphicX slightly surpasses ACGAN-GNNExplainer in explanation accuracy (ACC_{exp}) when $R = 0.9$, it falls behind in terms of fidelity. In practical applications of GNN explanations, maintaining high fidelity without sacrificing accuracy is critical, and from this perspective, our method has a clear advantage. Similarly, Table 9 shows that ACGAN-GNNExplainer consistently outperforms its competitors in both fidelity and accuracy across different values of R.

Table 9. The results of explanations on NCI1 dataset: $ACC_{exp}(\uparrow)$, $Fid^{+}(\uparrow)$, $Fid^{-}(\downarrow)$. The presented outcomes encompass averages from five runs alongside their corresponding standard deviations. Note that the best-performing results have been emphasised in bold. In this table, "Our Method" refers to our proposed method ACGAN-GNNExplainer.

R	Metrics	GNNExplainer	Gem	OrphicX	Our Method
0.5	Acc_{exp}	0.5961	0.5645	0.562	**0.6569**
	Fid^{+}	0.3358	0.3796	0.3114	**0.4015**
	Fid^{-}	0.2749	0.3066	0.309	**0.2141**
0.6	Acc_{exp}	0.6107	0.6083	**0.6496**	**0.6496**
	Fid^{+}	0.3625	0.4307	0.3431	**0.4523**
	Fid^{-}	0.2603	0.2628	0.3236	**0.2214**
0.7	Acc_{exp}	0.6788	0.6837	0.6083	**0.6861**
	Fid^{+}	0.3844	0.4282	0.3382	**0.4453**
	Fid^{-}	0.1922	0.1873	0.2628	**0.1849**
0.8	Acc_{exp}	0.7616	0.7518	0.708	**0.7932**
	Fid^{+}	0.3747	0.4404	0.3698	**0.4672**
	Fid^{-}	0.1095	0.1192	0.163	**0.0779**
0.9	Acc_{exp}	0.8127	0.8321	0.8102	**0.8446**
	Fid^{+}	0.3236	0.3212	0.3139	**0.3942**
	Fid^{-}	0.0584	0.0389	0.0608	**0.0254**

ACGAN-GNNExplainer consistently achieves higher $Fidelity^{+}$ scores, reflecting its ability to capture the most important subgraphs. Moreover, the lower $Fidelity^{-}$ scores, compared to other methods, emphasise the sufficiency of our explanations by capturing the critical information necessary for accurate predictions while minimising irrelevant noise. Furthermore, ACGAN-GNNExplainer consistently outperforms others in explanation accuracy, demonstrating its ability to effectively capture the reasoning behind the predictions of the GNN model. Overall, these results highlight the effectiveness of ACGAN-GNNExplainer in generating faithful and reliable explanations.

5. Discussion

Merits of our methods. Our approaches present several key advantages:

- They effectively capture the underlying patterns in graphs, naturally providing explanations at the desired scale.
- Once trained, they can generate explanations for previously unseen graphs without requiring retraining.
- They consistently produce valid and meaningful subgraphs, facilitated by the continuous supervision of the discriminator.
- They exhibit robust performance across diverse tasks, such as node classification and graph classification.

Limitations of GAN-GNNExplainer. Despite the advancements of GAN-GNNExplainer in enhancing the explainability of GNNs, several limitations persist in GAN-GNNExplainer:

- Effects of reliability of real-world datasets on performance: Real-world graph datasets are often affected by nuisance factors, such as noise in node features and graph structures. This consequently affects the performance of GAN-GNNExplainer.
- Absence of fidelity considerations: Although fidelity is crucial for faithful explanations, the design objective of GAN-GNNExplainer is to balance accuracy and explainability, with fidelity not explicitly optimised as a core criterion.

These challenges can be mitigated by incorporating an enhanced discriminator to ensure reliability and introducing a fidelity-specific loss term during training to improve fidelity. In our ACGAN-GNNExplainer, we have effectively resolved this limitation.

Future work. Although ACGAN-GNNExplainer has mitigated some of these limitations and achieved competitive performance in terms of both accuracy and fidelity, it still has areas that warrant further refinement:

- Preprocessing overhead: The preprocessing step required to distill real explanations for training data imposes significant computational overhead and time constraints.
- High demand for training graphs: The method also requires a substantial number of training graphs to achieve effective performance.

By addressing these limitations, we can enhance the capabilities and applicability of ACGAN-GNNExplainer, contributing to more robust and equitable interpretability solutions for graph-based models across various domains. Future research should focus on refining the model architecture to reduce reliance on ground-truth data, thereby streamlining the preprocessing stage and improving efficiency. Additionally, exploring techniques for efficient data augmentation or semi-supervised learning could help alleviate the need for extensive training data.

6. Conclusions

Understanding the internal mechanisms of GNNs is essential for increasing confidence in their predictions, ensuring the dependability of their use in practical applications, and fostering the development of trustworthy GNN models. To achieve these goals, a variety of methods have been proposed in recent years. Although these approaches exhibit notable effectiveness in certain aspects, many encounter challenges in delivering strong performance on real-world datasets.

To address this limitation, we propose two approaches in this paper. First, we introduce a novel explainer named GAN-GNNExplainer for GNN models. This method employs a generator to produce explanations and a discriminator to oversee the generation process, ensuring the reliability of the explanations. However, GAN-GNNExplainer has limitations in generating faithful explanations and does not perform well on real-world datasets.

To overcome the limitations of GAN-GNNExplainer, we introduce ACGAN-GNNExplainer, an advanced explainer for GNN models. This approach also utilises a generator to create explanations but employs a discriminator that consistently monitors the generation process, resulting in explanations that are both reliable and faithful.

To evaluate the effectiveness of our proposed methods, we conduct comprehensive experiments on both synthetic and real-world graph datasets. We compare the fidelity and accuracy of our approaches against other well-known GNN explainers. The results clearly demonstrate that ACGAN-GNNExplainer excels in producing high-fidelity and accurate explanations, particularly when applied to real-world datasets.

Author Contributions: Conceptualization, Y.L. and J.Z.; methodology, Y.L.; validation, Y.L.; formal analysis, Y.L.; investigation, Y.L.; resources, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L., J.Z., B.Z. and N.S.; visualization, Y.L.; supervision, J.Z. and F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, B.; Zhou, H.; Wang, W.; Chen, L.; Ma, J.; Zheng, Z. HDM-GNN: A Heterogeneous Dynamic Multi-view Graph Neural Network for Crime Prediction. *Acm Trans. Sens. Netw.* **2024**. [CrossRef]
2. Klosa, D.; Büskens, C. Low Cost Evolutionary Neural Architecture Search (LENAS) Applied to Traffic Forecasting. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 830–846. [CrossRef]
3. Deng, S.; de Rijke, M.; Ning, Y. Advances in Human Event Modeling: From Graph Neural Networks to Language Models. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, 25–29 August 2024; ACM: New York, NY, USA, 2024; pp. 6459–6469. [CrossRef]

4. Liu, M.; Srivastava, G.; Ramanujam, J.; Brylinski, M. Insights from Augmented Data Integration and Strong Regularization in Drug Synergy Prediction with SynerGNet. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 1782–1797. [CrossRef]
5. Sun, W.; Xu, J.; Zhang, W.; Li, X.; Zeng, Y.; Zhang, P. Funnel graph neural networks with multi-granularity cascaded fusing for protein-protein interaction prediction. *Expert Syst. Appl.* **2024**, *257*, 125030. [CrossRef]
6. Malhi, U.S.; Zhou, J.; Rasool, A.; Siddeeq, S. Efficient Visual-Aware Fashion Recommendation Using Compressed Node Features and Graph-Based Learning. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 2111–2129. [CrossRef]
7. van Mourik, F.; Jutte, A.; Berendse, S.E.; Bukhsh, F.A.; Ahmed, F. Tertiary Review on Explainable Artificial Intelligence: Where Do We Stand? *Mach. Learn. Knowl. Extr.* **2024**, *6*, 1997–2017. [CrossRef]
8. Rizzo, L.; Verda, D.; Berretta, S.; Longo, L. A Novel Integration of Data-Driven Rule Generation and Computational Argumentation for Enhanced Explainable AI. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 2049. [CrossRef]
9. Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; Leskovec, J. GNNExplainer: Generating Explanations for Graph Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; pp. 9240–9251.
10. Huang, Q.; Yamada, M.; Tian, Y.; Singh, D.; Chang, Y. GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks. *IEEE Trans. Knowl. Data Eng.* **2022**, *35* , 1–6. [CrossRef]
11. Luo, D.; Cheng, W.; Xu, D.; Yu, W.; Zong, B.; Chen, H.; Zhang, X. Parameterized Explainer for Graph Neural Network. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, Virtual, 6–12 December 2020.
12. Lin, W.; Lan, H.; Li, B. Generative Causal Explanations for Graph Neural Networks. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Virtual, 18–24 July 2021; Proceedings of Machine Learning Research—PMLR: New York, NY, USA, 2021; Volume 139, pp. 6666–6679.
13. Yuan, H.; Tang, J.; Hu, X.; Ji, S. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In Proceedings of the KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, 23–27 August 2020; ACM: New York, NY, USA, 2020; pp. 430–438. [CrossRef]
14. Li, Y.; Zhou, J.; Dong, Y.; Shafiabady, N.; Chen, F. ACGAN-GNNExplainer: Auxiliary Conditional Generative Explainer for Graph Neural Networks. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, UK, 21–25 October 2023; ACM: New York, NY, USA, 2023; pp. 1259–1267. [CrossRef]
15. Odena, A.; Olah, C.; Shlens, J. 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—PMLR: New York, NY, USA, 2017; Volume 70, pp. 2642–2651.
16. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.C.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
17. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
18. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, 5–10 December 2016; pp. 2172–2180.
19. Waheed, A.; Goyal, M.; Gupta, D.; Khanna, A.; Al-Turjman, F.M.; Pinheiro, P.R. CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved COVID-19 Detection. *arXiv* **2021**, arXiv:2103.05094. [CrossRef] [PubMed]
20. Ding, H.; Chen, L.; Dong, L.; Fu, Z.; Cui, X. Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection. *Future Gener. Comput. Syst.* **2022**, *131*, 240–254. [CrossRef]
21. Alawad, D.M.; Katebi, A.; Hoque, M.T. Enhanced Graph Representation Convolution: Effective Inferring Gene Regulatory Network Using Graph Convolution Network with Self-Attention Graph Pooling Layer. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 1818–1839. [CrossRef]
22. Fu, C.; Su, Y.; Su, K.; Liu, Y.; Shi, J.; Wu, B.; Liu, C.; Ishi, C.T.; Ishiguro, H. HAM-GNN: A hierarchical attention-based multi-dimensional edge graph neural network for dialogue act classification. *Expert Syst. Appl.* **2024**, 261, 125459. [CrossRef]
23. Zhang, M.; Chen, Y. Link Prediction Based on Graph Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018; pp. 5171–5181.
24. Jiang, W.; Luo, J. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.* **2022**, *207*, 117921. [CrossRef]
25. Chen, D.; Zhao, H.; He, J.; Pan, Q.; Zhao, W. An Causal XAI Diagnostic Model for Breast Cancer Based on Mammography Reports. In Proceedings of the 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Houston, TX, USA, 9–12 December 2021; pp. 3341–3349. [CrossRef]
26. Lin, W.; Lan, H.; Wang, H.; Li, B. 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 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 13719–13728. [CrossRef]
27. Wang, X.; Wu, Y.; Zhang, A.; Feng, F.; He, X.; Chua, T.S. Reinforced Causal Explainer for Graph Neural Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 2297–2309. [CrossRef] [PubMed]

28. Shan, C.; Shen, Y.; Zhang, Y.; Li, X.; Li, D. Reinforcement Learning Enhanced Explainer for Graph Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, Virtual, 6–14 December 2021; pp. 22523–22533.
29. Lucic, A.; ter Hoeve, M.A.; Tolomei, G.; de Rijke, M.; Silvestri, F. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics, AISTATS 2022, Virtual, 28–30 March 2022; Proceedings of Machine Learning Research—PMLR: New York, NY, USA, 2022; Volume 151, pp. 4499–4511.
30. Bajaj, M.; Chu, L.; Xue, Z.Y.; Pei, J.; Wang, L.; Lam, P.C.; Zhang, Y. Robust Counterfactual Explanations on Graph Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, Virtual, 6–14 December 2021; pp. 5644–5655.
31. Wang, X.; Wu, Y.; Zhang, A.; He, X.; Chua, T. Towards Multi-Grained Explainability for Graph Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, Virtual, 6–14 December 2021; pp. 18446–18458.
32. Granger, C. Investigating causal relations by econometric models and cross-spectral methods. In *Essays in Econometrics: Collected Papers of Clive WJ Granger*; Cambridge University Press: Cambridge, UK, 2001; pp. 31–47.
33. Kazius, J.; McGuire, R.; Bursi, R. Derivation and validation of toxicophores for mutagenicity prediction. *J. Med. Chem.* **2005**, *48*, 312–320. [CrossRef] [PubMed]
34. Wale, N.; Watson, I.A.; Karypis, G. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowl. Inf. Syst.* **2008**, *14*, 347–375. [CrossRef]
35. Li, Y.; Zhou, J.; Verma, S.; Chen, F. A Survey of Explainable Graph Neural Networks: Taxonomy and Evaluation Metrics. *arXiv* **2022**, arXiv:2207.12599.
36. Yuan, H.; Yu, H.; Wang, J.; Li, K.; Ji, S. On Explainability of Graph Neural Networks via Subgraph Explorations. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Virtual, 18–24 July 2021; Proceedings of Machine Learning Research—PMLR: New York, NY, USA, 2021; Volume 139, pp. 12241–12252.