C02029: Doctor of Philosophy

PhD Thesis: Analytics

April 2024

Faithful and Fair Generative Explainers for Graph Neural Networks

Yiqiao Li

Data Science Institute

School of Computer Science

Faculty of Engineering & IT

University of Technology Sydney

NSW - 2007, Australia

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Yiqiao Li, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy in the School of Computer Science, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: _____

[Yiqiao Li]

DATE: 5th April, 2024

PLACE: Sydney, Australia

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. The explanation of GNNs, which is essential to understanding the fundamental working mechanism of complex GNNs, guaranteeing the safety of their applications and promoting the reliability of GNNs, has attracted significant attention in recent years. These active research works could be categorized into two mainstreams–*factual explanations (FE)* and *counterfactual explanations (CFE)*. FE aims to answer the question: *why GNNs make that particular decision by finding the most important subgraphs/features*. CFE, on the contrary, attempts to answer the question: *how to modify the original graphs so that GNNs could make the desired predetermined prediction*. CFE typically generates a new graph conditioned by the desired predetermined prediction. This thesis is dedicated to exploring both CF and CFE explainers for GNN, with a focus on addressing three special research questions.

In particular, for generating FE explanations, we propose the primary research question, "*how to generate FE for GNNs?*" To solve this research question, we propose a novel GNN explainer called *GAN-GNNExplainer*, which is a Generative Adversarial Network (GAN) – based explanation method. Specifically, for GAN-GNNExplainer, the generator learns to produce explanations for the input graph $\mathbf{G}$, 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 demonstrates a notable advancement in the accuracy of explanations, successfully addressing some limitations of current popular GNN explainers. However, it has inadequate reliability on real-world datasets and lacks fidelity.

To overcome these constraints, the second research question posited is, "*how to generate faithful FE for GNNs?*" For this research question, we aim to improve the fidelity of explanations on real-world datasets. Specifically, we introduce an enhanced method on top of the GAN-GNNExplainer, dubbed *ACGAN-GNNExplainer*, which leverages the Auxiliary Classifier Generative Adversarial Network (ACGAN) as its backbone to generate explanations for GNNs. To be specific, the input graph $\mathbf{G}$, along with its corresponding label $\mathscr{F}(\mathbf{G})$ determined by the target GNN model $\mathscr{F}$, 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.

On the other hand, the CFE for GNNs plays a crucial role in explaining GNNs from the perspective of "how to minimally modify the input graphs so that the GNNs are able to make predictions as predetermined". Several recent works have emerged to generate CFE using different strategies. However, these methods typically require a large amount of training data, which might not be practical when such training data are

not available, and worse, they have no control over ensuring the generated explanations are unbiased. Hence, we present the third research question, "*how to generate fair CFE for GNNs?*" To address this research question, we propose *fairCFE*, which uses a deep decoder as our generative model and is conditioned by predetermined predictions. We jointly optimize the input seed and the network parameters of the deep decoder for a given graph, requiring no additional training data sets. In addition, we introduce a novel *fair* loss to guide the entire generation process so that the generated counterfactual explanations are guaranteed to be unbiased. To verify the effectiveness of our method, we have conducted experiments on both synthetic and real-world datasets and compared with state-of-the-art baselines. The experimental results demonstrate the superiority of our method over other models.

ACKNOWLEDGMENTS

I am delighted to have had the opportunity to pursue my doctorate degree at the University of Technology Sydney (UTS), where I have gained invaluable knowledge and experienced significant academic growth over the past few years. I am particularly grateful for the excellent learning environment and facilities provided by the university. It is truly an honor to be a doctoral student at UTS.

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Professor Jianlong Zhou, and co-supervisor, Professor Fang Chen, for providing me with the opportunity to study at UTS and for their generous scholarship support, which has enabled me to embark on my research journey smoothly. I am deeply grateful for their guidance, assistance, and support throughout my doctoral studies. I would like to extend a special thank you to Professor Jianlong Zhou for his meticulous guidance, unwavering support, and caring presence throughout my research journey.

I would also like to express my gratitude to my family for their love, understanding, and steadfast support, which has been my guiding force during my doctoral journey.

I am also very grateful to all my friends and peers who have supported me both academically and personally during my time at UTS. Thank you for your assistance and encouragement.

Overall, I am profoundly grateful to everyone who has contributed to my academic and personal growth during my doctoral experience at UTS. Your support has been invaluable, and I am truly fortunate to have had such a supportive network.

List of Publications

Related to the Thesis :

1. **Li, Y.**, Zhou, J., Wu, M., & Chen, F. (2024). fairCFE: Fair Counterfactual Explanations for Graph Neural Networks. *WWW 2025* (under review).

2. **Li, Y.**, Zhou, J., Zheng, B., Shafiabady, N., & Chen, F. (2024). Reliable and Faithful Generative Explainers for Graph Neural Networks. *Machine Learning and Knowledge Extraction* (Accepted).

3. **Li, Y.**, Zhou, J., Dong, Y., Shafiabady, N., & Chen, F. (2023, October). ACGAN-GNNExplainer: Auxiliary Conditional Generative Explainer for Graph Neural Networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (pp. 1259-1267).

4. **Li, Y.**, Zhou, J., Verma, S., & Chen, F. (2022). A survey of explainable graph neural networks: Taxonomy and evaluation metrics. *arXiv preprint arXiv:2207.12599*.

Others:

5. Zheng, B., Zhou, J., Liu, C., **Li, Y.**, & Chen, F. (2024). Explaining Imitation Learning Through Frames. *IEEE Intelligent Systems*.

6. Wang, B., Zhou, J., **Li, Y.**, & Chen, F. (2023, November). Impact of Fidelity and Robustness of Machine Learning Explanations on User Trust. In *Australasian*

Joint Conference on Artificial Intelligence (pp. 209-220). Singapore: Springer Nature Singapore.

7. **Li, Y.**, Verma, S., Yang, S., Zhou, J., & Chen, F. (2022, December). Are Graph Neural Network Explainers Robust to Graph Noises?. In *Australasian Joint Conference on Artificial Intelligence* (pp. 161-174). Cham: Springer International Publishing.

Table of Contents

List of Figures

Figure **Page**

LIST OF TABLES

TABLE **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

A graph **G** can be viewed as a representation of a certain relationship formed by a set of nodes **N** and edges **E**. The graph is an ideal data structure that can be used to model a variety of real-world datasets (e.g., social networks [33], transaction networks [54], and molecules [57], see Figure 1.1). With the resurgence of deep learning, Graph Neural Networks (GNNs) have been a powerful tool to model graph data and have achieved impressive performance in many domains and applications, such as recommendation systems, credit estimation, drug design and development, to name a few [19, 61, 71]. 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.

The explanation for GNNs has attracted significant attention in recent years. These active research works could be categorized into two mainstreams–*factual explanations (FE)* and *counterfactual explanations (CFE)*. FE aims to answer the question: *why GNNs*

Figure 1.1: Examples of Graph Data. In the domain of social networks, Graph Neural Networks (GNNs) have been effectively utilized for tasks such as friend recommendations [33], advertising optimization [74], and other pertinent applications. Within transaction networks, GNNs have demonstrated utility in fraud detection [39], credit estimation [54], and related endeavours. In the realm of molecular data analysis, GNNs have found applications in drug design [72], drug development [55], and various other areas of pharmaceutical research and development.

make that particular decision by finding the most important subgraphs/features. Notable examples include GNNExplainer [76], OrphicX [38], and ACGAN-GNNExplainer [34]. CFE, on the contrary, attempts to answer the question: *how to modify the original graphs so that GNNs could make the desired predetermined prediction*. CFE typically generates a new graph conditioned by the desired predetermined prediction. Recent works in this direction include CF-GNNExplainer [40], CFF [59], and CLEAR [42].

Particularly, FE elucidates the influential subgroups within graphs, shedding light on the underlying mechanisms driving GNN decisions. By understanding these crucial factors, stakeholders gain valuable insights into the rationale behind GNN predictions, enabling informed decision-making and fostering trust in model outcomes. For instance, consider a scenario in social network analysis where a GNN is employed to predict user engagement with online content. FE might reveal specific clusters of interconnected users whose collective behaviour strongly correlates with the predictions made by the GNN. These clusters could represent tightly-knit communities or influential nodes whose interactions contribute disproportionately to the overall network dynamics. By identifying and highlighting such important subgroups, FE not only enhances our understanding of GNN decisions but also unveils underlying patterns and structures within complex datasets, enabling more informed decision-making in practical scenarios.

On the other hand, CFE empowers stakeholders to explore hypothetical scenarios by identifying minimal modifications to input graphs that alter GNN predictions. This capability is invaluable for sensitivity analysis, risk assessment, and model refinement, as it allows stakeholders to assess the robustness of GNN predictions and proactively mitigate potential biases or errors. For instance, consider a recommendation system employing GNNs to suggest personalized items to users based on their historical interactions. A CFE approach might explore subtle adjustments to the user-item interaction graph, such as adding or removing edges representing past interactions, to influence the recommendations towards desired outcomes, such as increased user engagement or satisfaction. By discerning these minimal modifications, CFE empowers stakeholders to fine-tune GNNs for specific objectives while preserving the integrity of the underlying data structure.

Overall, both FE and CFE hold significant significance and importance in the realm of GNNs. They play pivotal roles in elucidating the decision-making processes of GNNs and enhancing their utility and reliability in real-world applications. Hence, this thesis is dedicated to elucidating the mechanisms of FE and CFE by devising three distinct GNN explainers. These innovative explanation methodologies not only enhance the explainability and transparency of GNN decision-making processes but also facilitate the responsible integration of GNNs into various domains, thereby contributing to the overarching objective of harnessing Artificial Intelligence (AI) for societal benefit.

1.2 Motivations

GNNs have been effectively implemented in a variety of real-world applications, while their underlying work mechanisms remain a mystery. To unveil this mystery and advocate trustworthy decision-making, many GNN explainers have been proposed. These methods have provided some elucidation of GNNs; however, substantial work is still

required in the following aspects:

- Explanation Scale (Local or Global Explanation). Evaluate whether the explanation focuses on a specific instance (local) or captures broader patterns shared by a group (global). Local explanations provide precision, while global explanations offer insights into overarching trends.

- Generalizability. Assess the explainer's ability to work on unseen graphs without retraining. High generalizability is essential for scalable and dynamic applications.

- Versatility. Measure whether the explainer can adapt to different tasks, such as node classification, graph classification, or link prediction. A versatile explainer should perform well on diverse objectives.

- Fidelity. Check whether the explanation accurately reflects the model's decision-making process. High fidelity ensures reliability and builds user trust in the explanations.

- Fairness. Evaluate whether the explainer generates unbiased explanations across different subgroups. Fairness is crucial to promoting equity in sensitive applications.

- Agnostic. Determine whether the explainer maintains consistent performance across various GNN architectures. An agnostic explainer is versatile and broadly applicable to different models.

Therefore, the primary objective of this thesis is to devise GNN explainers possessing the aforementioned desirable properties.

1.3 Research Questions and Objectives

This thesis is dedicated to offering comprehensive insights into the realms of FE and CFE, specifically for GNNs. Below, we outline our research questions and objectives.

1. **How to Generate FE for GNNs?**

   GNNs have gained a lot of attention in recent years because of their ability to capture the structural information of graphs and make predictions based on that information. Explaining GNNs is important for improving our understanding of these models and their applications and for building trust with users and stakeholders in the domains where they are used.

   The motivations behind developing techniques for explaining GNNs are to enhance the transparency, explainability, and trustworthiness of GNN models. GNNs are powerful models for processing graph-structured data and have been successful in various applications such as social network analysis, drug discovery, and recommendation systems. However, due to their complex structure and high-dimensional representations, GNNs can be challenging to understand and interpret. This lack of explainability can limit their application in domains where model explainability is crucial, such as healthcare and finance. Therefore, the development of explainability techniques for GNNs is motivated by the need to provide insights into how these models make predictions and build trust with users and stakeholders.

   Objective: Generating FE for GNNs. We are inspired by the explanations generated by other explainers, such as GNNExplainer [76], XGNN [77], and OrphicX [38]. These methods explain GNNs to some degree, while they often suffer from one or more limitations—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 limit to a specific task (e.g., node clas-

sification, graph classification, etc.). To solve these limitations, this work proposes a GAN-GNNExplainer, which is a GAN-based explanation method. Specifically, a generator is employed to generate explanations for original input graphs. And a discriminator is adopted to monitor the generation process to ensure the quality of the explanations and further improve the explanation accuracy. We experiment with our proposed method on both synthetic and real-world graph datasets and demonstrate the superiority of our proposed methods over other GNN explainers.

2. **How to Generate Faithful FE for GNNs?**

   As we mentioned before, proposing faithful explanations for GNNs is an important research topic that aims to ensure the reliability, stability, and explainability of these models in practical applications. GNNs are complex models that can capture complex patterns and relationships in graph-structured data, but their internal workings are often opaque and difficult to understand. This lack of explainability can limit their adoption and trust in domains where model explainability is critical, such as healthcare, finance, and law enforcement. Therefore, developing methods to generate reliable and faithful explanations of GNNs is essential to build trust and confidence in GNN models. These explanations offer insights into factors influencing model predictions, help identify potential errors, and assist domain experts in making informed decisions based on the model's outputs.

   Objective: Generating Faithful FE for GNNs. High accuracy and faithful explanations increase the trust level of the GNN models. This work aims to propose a GNN explainer that explains GNN models with high fidelity. In this work, we introduce the Auxiliary Classifier Generative Adversarial Network (ACGAN) [46] 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.

3. **How to Generate Fair CFE for GNNs?**

   Counterfactual explanations for GNNs play a crucial role in explaining GNNs from the perspective of *"how to minimally modify the input graphs so that the GNNs are able to make predictions as predetermined"*. Fairness is also an important part of GNNs. Although current CFE explainers have demonstrated impressive performance in synthetic graph datasets, the present CFE models are limited to generating a new graph that enables the GNNs to make the desired given prediction, like CF-GNNExplainer [40], CFF [59], and CLEAR [42]. They do not consider *fairness*, which has significant impacts in real-world applications. Without taking *fairness* into account, a CFE model may produce counterfactual explanations that exhibit a bias towards a specific gender or ethnicity. Such explanations could potentially mislead or even endanger practitioners in real-world applications such as credit evaluation and job marking. Thus, this work revolves around the development of fair counterfactual explanations for GNNs.

   Objective: Generating Fair CFE for GNNs. Our objective is to develop a method for generating fair CFE for GNN models. Several recent works have emerged to generate CFE with different strategies. However, these methods typically require a large amount of data for training, which might not be practical when such training data are not available, and worse, they have no control on ensuring the generated explanations are unbiased. To overcome these limitations, in this work, we propose *fairCFE*, which uses a deep decoder as our generative model and is conditioned by predetermined predictions. We jointly optimize the input seed and the network parameters of the deep decoder for a given graph, requiring no

additional training data sets. In addition, we introduce a novel *fair* loss to guide the entire generation process so that the generated CFE are guaranteed to be unbiased. To verify the effectiveness of our method, we have conducted experiments on both node classification and graph classification with different datasets (synthetic and real-world) and compared with state-of-the-art baselines. The experimental results demonstrate the superiority of our method over other models.

1.4 Research Contributions

Each objective, as we mentioned above, has certain contributions. To resolve the first research question, we propose *GAN-GNNExplainer*, a Generative Adversarial Network (GAN)-based explainer for GNN models that provides consistent explanations for predictions made by the GNN model. To be specific, GAN-GNNExplainer uses a GAN [20] as the backbone for generating explanations. The original graph $\mathbf{G}$ that we want to explain is fed into the generator and then learns to generate the explanations. To enhance the explanation accuracy and ensure the validity of a generated subgraph, a discriminator is adopted, and it attempts to distinguish "fake" and "real" graphs, which signals feedback to the generator and monitors the whole generation process. Although GAN has been widely used in the fields of computer vision [65], image processing [50], natural language processing [49], and healthcare [56], to the best of our knowledge, this is the first time to use GAN to explain GNN models. Our method *GAN-GNNExplainer* demonstrates the following merits: 1) it provides *global* explanation in nature; 2) it can generate explanations for unseen graphs without retraining; 3) the discriminator monitors the generation process and increases the chance of generating valid important subgraphs; 4) it performs consistently well across different tasks and graph datasets. Our contributions can be summarized below:

- We present a novel explainer, dubbed *GAN-GNNExplainer*, for GNN models, which uses a generator to generate explanations and uses a discriminator to monitor the generation process;

- We empirically evaluate and demonstrate the superiority of *GAN-GNNExplainer* in different datasets, including synthetic and real-world graph datasets, and different tasks, including node classifications and graph classifications.

In order to address the second research question, we propose a new GNN explanation method dubbed ACGAN-GNNExplainer, which uses the auxiliary classifier Generative Adversarial Network (ACGAN) [46] as its backbone to generate explanations for GNNs. 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 $\mathscr{F}(\mathbf{G})$, which is determined by the target GNN model $\mathscr{F}$. 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. 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 research question 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.

In the third research work, we attempt to propose a practical counterfactual explanation model for GNNs. It should not only produce *faithful* counterfactual explanations, but also ensure *fairness* in its explanations. To achieve this goal, we adopt deep decoders $\mathscr{D}_\omega$ parameterized by ω and $\mathscr{D}_\theta$ parameterized by θ as our CFE-generative model and learn a tailored optimal $\mathscr{D}_\omega$ and $\mathscr{D}_\theta$ for each given graph. By doing so, our model eliminates the need for massive training data and has naturally addressed the data distribution-shift issue. This *untrained* idea has been increasingly gaining popularity in the field of computer vision [23, 28, 30–32, 62, 85]. Furthermore, in order to guarantee that our CFE-generative model generates explanations that are fair, we propose a new *fairness* loss and incorporate it into the decoder's loss function to achieve our final optimization objective. Our main contributions to this research question include the following:

- We propose a novel *untrained* CFE-generative model dubbed *fairCFE* which could generate *faithful* CFEs, requiring no extra massive training data;

- We introduce a new *fairness* loss to guide the generation procedure of our *fairCFE* so that the generated CFEs are *fair* and *unbiased*;

- We demonstrate the effectiveness of our *fairCFE* through extensive experiments on different datasets.

1.5 Thesis Structure

This thesis tries to scrutinize explanations for GNNs. Three research questions are proposed and solved, including how to generate explanations, generate faithful explanations and generate fair counterfactual explanations. The structure of the remaining chapters of this thesis is organized as follows (see Figure 1.2.):

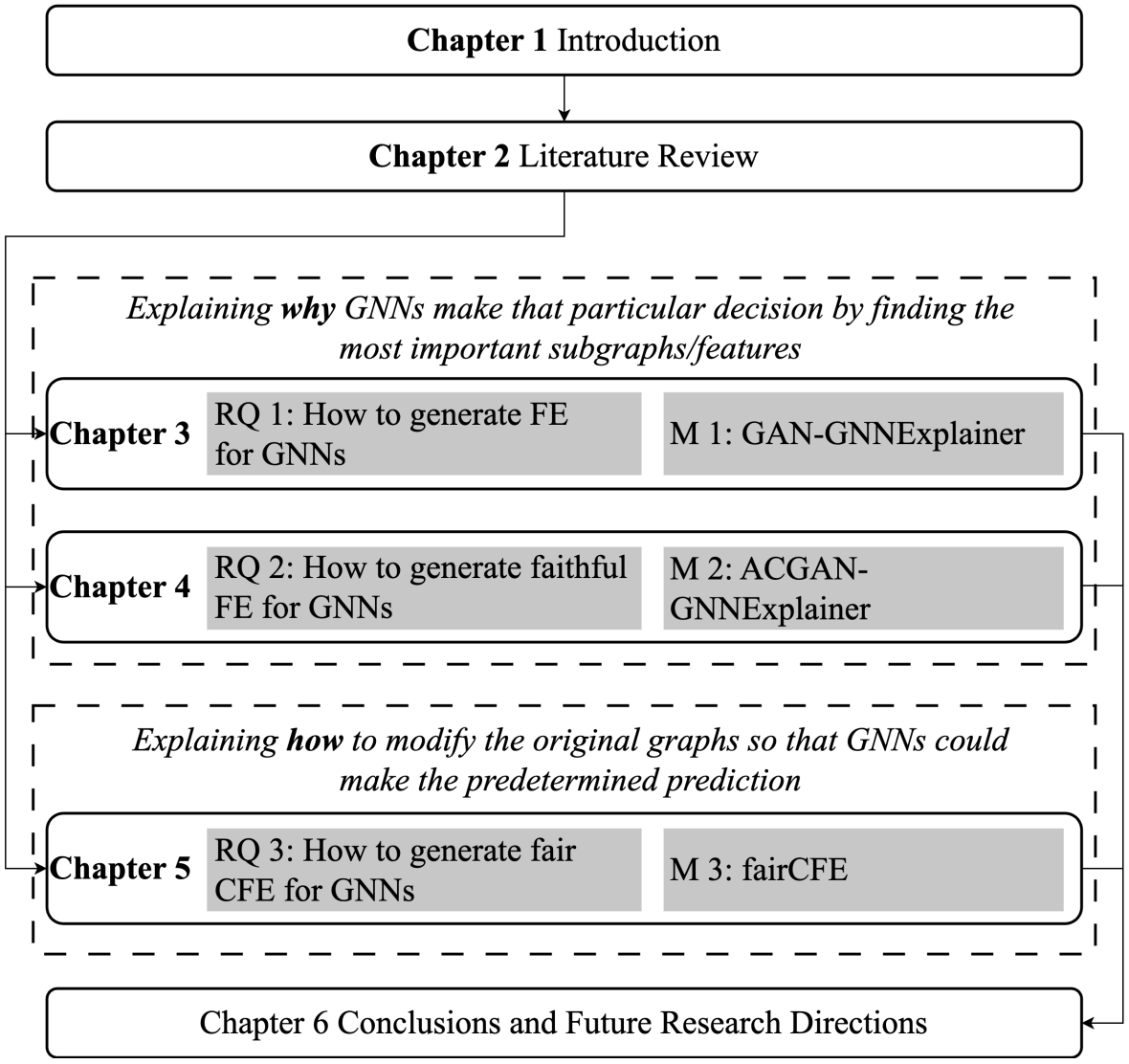

Figure 1.2: Thesis Structure. In this figure, "RQ" is the abbreviation for "Research Question". "M" is the abbreviation for "Method". "FE" means factual explanations, and the "CFE" means counterfactual explanations.

- Chapter 2 summarizes the literature on explainability, fairness, generative adversarial attack networks, and metrics for evaluating explainability.

- Chapter 3 presents a new framework to explain GNNs by incorporating GAN, dubbed as GAN-GNNExplainer, which is a Generative Adversarial Network (GAN) -based explanation method. Specifically, a generator is employed to generate explanations for original input graphs. And a discriminator is adopted to monitor the generation process to ensure the quality of the explanations and further improve the explanation accuracy. We experiment with our proposed method on both synthetic and real-world graph datasets and demonstrate the superiority of our proposed methods over other GNN explainers.

- Chapter 4 presents a novel faithful GNN explainer called ACGAN-GNNExplainer that adopts ACGAN as our backbone. Our approach leverages a generator to produce explanations for the original input graphs while incorporating a discriminator to oversee the generation process, ensuring explanation high 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.

- Chapter 5 presents a fair counterfactual explainer for GNNs, named fairCFE, which uses a deep decoder as our generative model and is conditioned by predetermined predictions. We jointly optimize the input seed and the network parameters of the deep decoder for a given graph, requiring no additional training data sets. In addition, we introduce a novel *fair* loss to guide the entire generation process so that the generated counterfactual explanations are guaranteed to be unbiased. To verify the effectiveness of our method, we have conducted experiments on both node classification and graph classification with different datasets (synthetic and

real-world) and compared with state-of-the-art baselines. The experimental results demonstrate the superiority of our method over other models.

- Chapter 6 summarizes the findings of this thesis and points out the directions of future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Graph Neural Network Explainers

Explaining the decision-making process of 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.

GNNs have emerged as a potent tool for handling datasets structured as graphs, demonstrating remarkable efficacy in diverse real-world applications, including social recommendation [73], credit estimation [54], and drug design and development [18]. An essential aspect lies in comprehending the functioning of GNN models. Research efforts have delved into explicating these models, broadly categorised into factual and counterfactual reasoning [35]. Within the domain of GNN explanations, factual reasoning entails generating sub-graphs that adhere to the condition "With these subgraphs,

consistent with the fact, the GNN prediction remains unchanged." Conversely, counterfactual reasoning involves generating sub-graphs that satisfy the condition "Without these subgraphs, inconsistent with the fact, the GNN prediction diverges." Essentially, factual reasoning strives to identify a sufficient set of edges/features that yield original predictions, while counterfactual reasoning seeks a necessary set of edges/features whose absence alters the prediction. The taxonomy of GNN explainers is shown in Figure 2.1.

Factual Explanations. Factual explanations elucidate the reasoning underlying individual predictions by identifying the minimal subgraph sufficient to produce the same prediction as the entire input graph. The *GNNExplainer* [76] as a pioneering method to provide factual explanations for GNNs, which identifies crucial subgraphs pivotal to the GNN model's predictions, offering localised explanations for GNN models. Similarly, *GraphLIME* [25] employs LIME values to generate local explanations for GNNs. In contrast, *PGExplainer* [41] utilises a probabilistic graph to provide instance-specific, model-level explanations with robust generalizability, distinguishing itself from GNNExplainer. While *SubgraphX* [80] explores subgraph information for GNN explanations.

Moreover, reinforcement learning offers another avenue for generating factual explanations in GNN models. For instance, Yuan et al. [77] introduce *XGNN*, a model-level explainer that trains a graph generator to maximise specific model predictions through generated graph patterns. Shan et al. [53] propose *RG-Explainer*, an enhanced explainer leveraging reinforcement learning in the inductive setting, showcasing superior generalisation capabilities.

Counterfactual Explanations. Counterfactual explanations aim to identify the smallest perturbation to the input graph that induces a change in the GNN's prediction. These perturbations typically involve edge removal or modifications to node features. For instance, CF-GNNExplainer [40] is designed to generate counterfactual explanations,

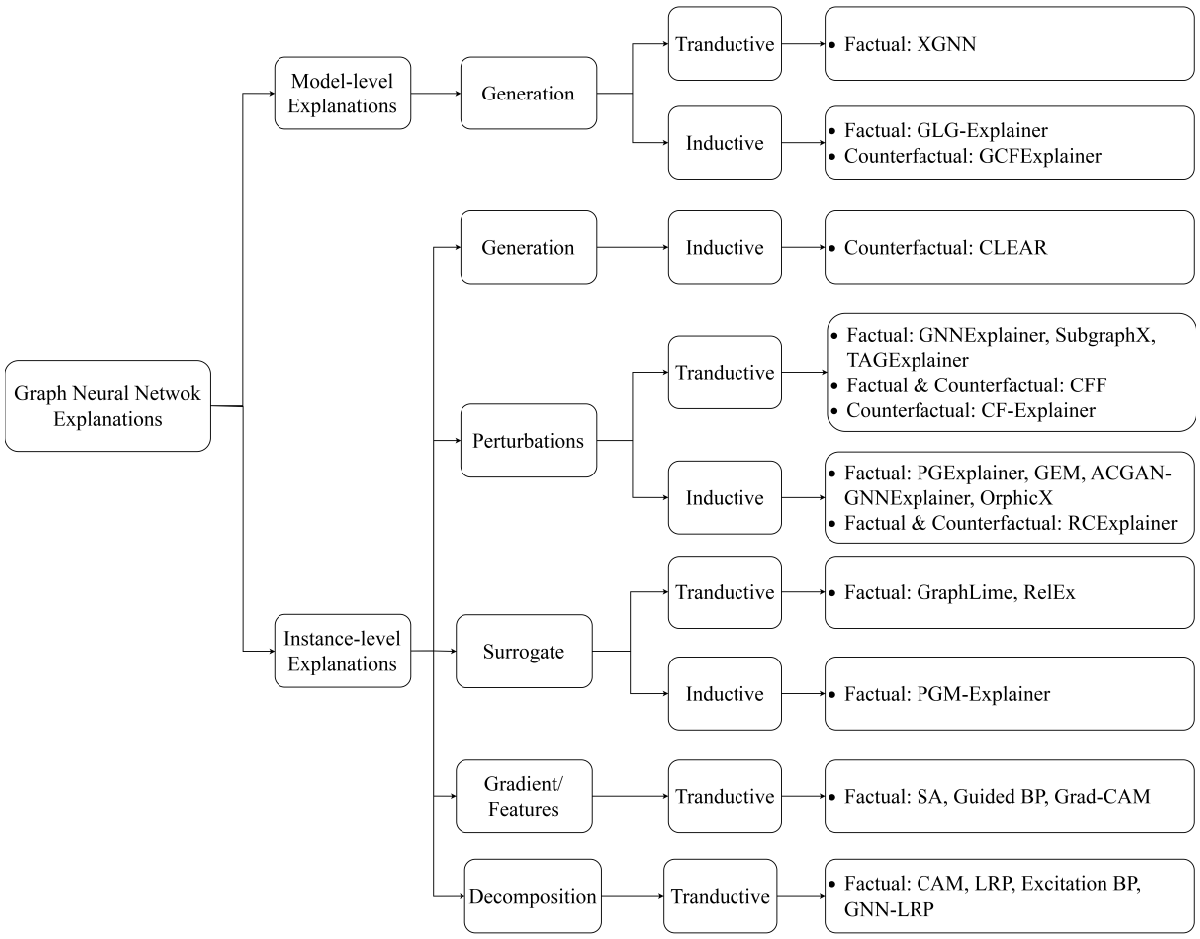

Figure 2.1: The Taxonomy of GNN Explainers.

primarily focusing on node classification datasets. Subsequently, Juntao et al. [59] enhanced CF-GNNExplainer, introducing CFF, a method that explicates GNNs through both factual and counterfactual perspectives. CFF enables the generation of both factual and counterfactual explanations controlled by a parameter.

Further advancements include the proposal by Bajaj et al. [7] of RCExplainer, specifically designed to generate robust counterfactual explanations. Jing et al. [42] present Clear, leveraging a generative model to produce counterfactual explanations for GNNs. Jialin et al. [10] introduce D4Explainer, utilising a discrete denoising diffusion model to generate counterfactual explanations for GNNs.

Expanding to specific domains, Carlo et al. [1] proposed a comprehensive density-

based counterfactual search framework. These contributions aim to generate counterfactual explanations for the explainable classification of brain networks, illustrating the diverse approaches undertaken to enhance understanding within this domain.

2.2 Evaluation Metrics for Graph Neural Network Explainers

Since explainers are used to explain why a certain decision has been made instead of depicting the whole black box, there is uncertainty about the fidelity of the explainer itself. Therefore, it is crucial to use the right metrics to evaluate the correctness and completeness of the interpretability techniques. Recently, GraphFramEx [5] and GRAPHXAI [2] have focused on defining the explainability metrics to evaluate the fidelity of GNNs explanations. Further, some evaluation metrics for XAI [83] are also available to be applied to GNN explainers. This section provides a short review of the prevalent evaluation metrics for GNNs explanations. Generally, we evaluate a GNN explainer from two aspects: performance and explanatory capability. Specifically, explanatory capability can be evaluated from qualitative and quantitative analyses, including accuracy and explainability evaluation. The taxonomy of metrics can be found in Figure 2.2.

2.2.1 Performance Evaluation

Efficiency. An efficient graph explanation algorithm should be able to provide explanations for a large number of decisions made by a machine learning model quickly and with minimal computational resources. This is particularly important in scenarios where real-time decision-making is required or where the volume of data is extremely large. In addition to being time and resource-efficient, an efficient graph explanation algorithm should also produce explanations that are accurate, interpretable, and fair. Achieving a

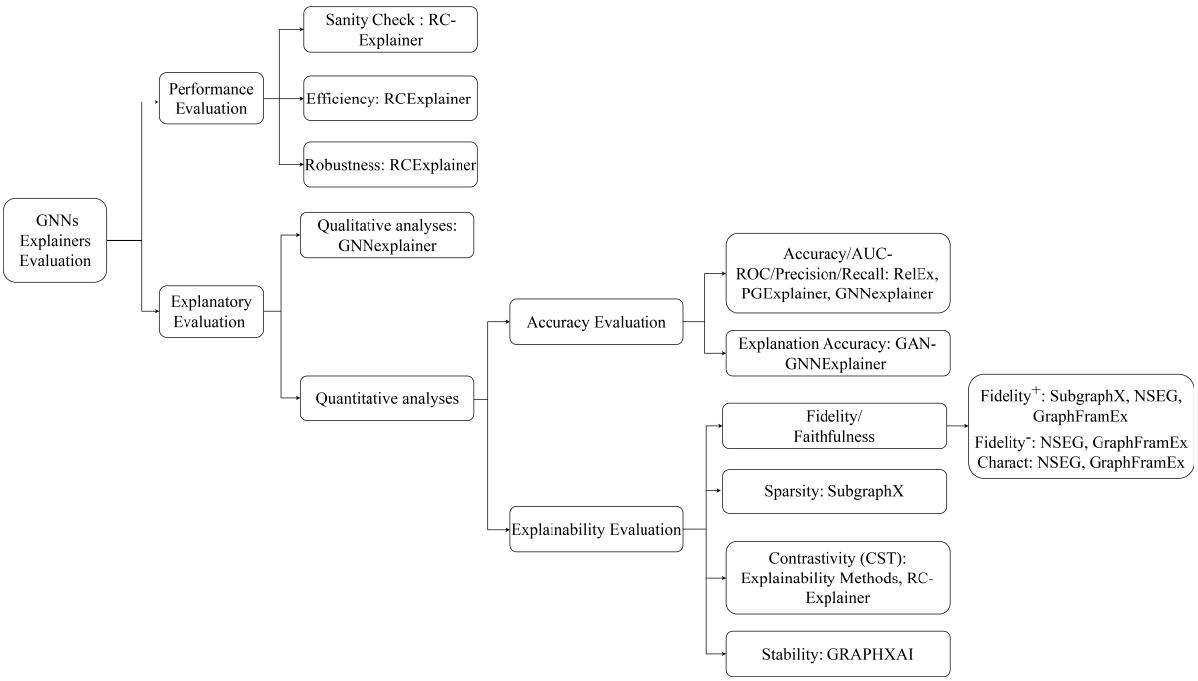

Figure 2.2: The Taxonomy of Metrics.

balance between efficiency and accuracy/fairness is an active research area in the field of graph explanation. In the paper [7], authors evaluate efficiency by comparing the average computation time taken for inference on unseen graph samples.

Robustness. It means the explanations of interpretation methods resist attacks such as input corruption/perturbation, adversarial attack and model manipulation. A robust interpretation method can provide similar explanations despite the presence of such attacks [41, 81]. Mohit et al. [7] define robustness by quantifying how much an explanation changes after adding noise to the input graph.

2.2.2 Explanatory Evaluation: Qualitative Analyses

Qualitative Analyses. Qualitative analyses are an important aspect of explainability in GNNs. These analyses involve examining the internal workings of a GNN to gain insight into how it makes decisions. This can be accomplished through techniques

such as visualization, feature importance analysis, and interpretation of node and edge embeddings. By conducting qualitative analyses, researchers and practitioners can better understand the factors that contribute to GNN decision-making, identify potential biases or errors, and improve the overall transparency and interpretability of the model. Ultimately, qualitative analyses are crucial for ensuring that GNNs are trustworthy and can be used effectively in real-world applications. Qualitative analyses have been widely used in recent research, such as GNNExplainer [76], PGExplainer [41], and GAN-GNNExplainer [36].

2.2.3 Explanatory Evaluation: Quantitative Analyses

2.2.3.1 Accuracy Evaluation

Accuracy evaluation refers to the process of assessing the correctness and fidelity of the explanations generated by an algorithm or model. Accurate explanations are essential for building trust in the machine learning model's decision-making process and for ensuring fairness and transparency. Therefore, accuracy evaluation is a crucial step in developing and evaluating graph explanation algorithms.

Accuracy (ACC). ACC is the proportion of explanations that are "correct". There are two definitions to measure the *accuracy* of explainable methods. First, one can use the percentage of the identified important features (e.g., nodes, node features, and edges) to the true important truth [81] (see Equation 2.1):

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\frac{|s_i|}{|S_i|_{gt}} \tag{2.1}$$

where $|S_i|_{gt}$ represents the ground-truth important number of features; while $|s_i|$ is the important features identified by the explainable methods; N is the total number of samples. While this approach is simple and intuitive, however, it requires the ground-

truth explanations of datasets, which is often hard to obtain in the real world. The other one is explanation accuracy.

Explanation Accuracy. This is derived from the perspective of model predictions and measures the prediction accuracy [36]. They use the predictions of the target GNN for the explanations to calculate the accuracy of the explanation. The accuracy of the explanation can be defined as Equation 3.3:

$$ACC_{exp} = \frac{|\mathscr{F}(\mathbf{G}) = \mathscr{F}(\mathbf{G}^s)|}{|T|} \tag{2.2}$$

where $\mathscr{F}$ is the pre-trained target GNN, $\mathbf{G}$ is the original graph we want to explain, and $\mathbf{G}^s$ is its corresponding explanation (e.g., the important subgraph), $|\mathscr{F}(\mathbf{G}) = \mathscr{F}(\mathbf{G}^s)|$ is the corrected classified number which means $\mathscr{F}(\mathbf{G}) = \mathscr{F}(\mathbf{G}^s)$, $|T|$ is the total number of the test set.

2.2.3.2 Explainability Evaluation

Fidelity. It measures whether the explanations are faithfully important to the model's predictions. The $Fidelity^+$ [79, 80] metric indicates the difference in predicted probability between the original predictions and the new prediction after removing important input features. In contrast, the metric $Fidelity^-$ [79, 80] represents prediction changes by keeping important input features and removing unimportant structures.

$$Fidelity^+ = \frac{1}{N}\sum_{i=1}^{N}(\mathscr{F}(G_i)_{y_i} - \mathscr{F}(G_i^{1-m_i})_{y_i}) \tag{2.3}$$

$$Fidelity^- = \frac{1}{N}\sum_{i=1}^{N}(\mathscr{F}(G_i)_{y_i} - \mathscr{F}(G_i^{m_i})_{y_i}) \tag{2.4}$$

where N is the total number of samples, and y_i is the class label. $\mathscr{F}(G_i)_{y_i}$ and $\mathscr{F}(G_i^{1-m_i})_{y_i}$ are the prediction probabilities of y_i when using the original graph G_i and the occluded graph $G_i^{1-m_i}$, which is gained by occluding important features found by explainers from the original graph. Thus, a *higher* $Fidelity^+$ (↑) is desired. $\mathscr{F}(G_i^{m_i})_{y_i}$ is the prediction probabilities of y_i when using the explanation graph $G_i^{m_i}$, which is obtained by important structures found by explainable methods. Thus a *lower* $Fidelity^-$ (↓) is desired. Specifically, $Fidelity^+$ and $Fidelity^-$ are used to quantify the necessity and sufficiency of the explanations, respectively. The higher $Fidelity^+$, the more necessary the explanation. On the contrary, the lower $Fidelity^-$, the more sufficient the explanation.

Characterization Score. The characterization score [5, 9] is a global evaluation metric that attempts to balance the sufficiency and necessity requirements. This approach is analogous to combining precision and recall in the Micro-F1 metric. The characterization score is the weighted harmonic mean of Fidelity+ and Fidelity- as defined below:

$$Charact = \frac{2 \times Fidelity^+ \times (1 - Fidelity^-)}{Fidelity^+ + (1 - Fidelity^-)} \tag{2.5}$$

Sparsity. It measures the fraction of features selected as important by explanation methods [48, 80], which is defined in Equation 2.6:

$$Sparsity = \frac{1}{N} \sum_{i=1}^{N} (1 - \frac{|s_i|}{|S_i|_{total}}) \tag{2.6}$$

where the $|S_i|_{total}$ represents the total number of features (e.g., nodes, nodes features, or edges) in the original graph model; while $|s_i|$ is the size of important features/nodes found by the explainable methods and it is a subset of $|S_i|$; N is the total number of samples. Note that higher sparsity values indicate that explanations are sparser and likely to capture only the most essential input information. Hence, a *higher* $Sparsity$ (↑) is desired.

Contrastivity (CST). CST means the ratio of the Hamming distance between binarized heat-maps for positive and negative classes [48]. The underlying idea behind contrastivity is that the highlighted features by an explanation method should vary across classes. [48] and [70] defined and used CST to evaluate the explainability of their methods. One can define fidelity as shown in Equation 2.7. And a *lower* CST^{-} ($\downarrow$) is desired.

$$\text{CST} = \mathbb{E}_{\mathcal{G}\sim\mathbb{G}}\mathbb{E}_{s\neq\hat{y}}[\rho(\Phi(\mathcal{G},s),\Phi(\mathcal{G},\hat{y}))] \tag{2.7}$$

Stability. Graph explanation stability refers to the ability of a GNN to produce consistent explanations even when the input graph is slightly altered or perturbed. This is important for ensuring the reliability and interpretability of the model's decisions. In [2], authors measure graph explanation stability by computer the instability degree. They calculate the instability as Equation 2.8.

$$\text{GES}\left(\mathbf{M}^{p}_{\mathcal{S}_{u'}},\mathbf{M}_{\mathcal{S}^{p}_{u}}\right) = \max D\left(\mathbf{M}_{\mathcal{S}^{p}_{u'}},\mathbf{M}_{\mathcal{S}^{p}_{u'}}\right), \quad \forall \mathcal{S}_{u'} \in \beta(\mathcal{S}_u) \tag{2.8}$$

where, $\mathcal{S}_u$ is the subgraph of node u, and the $\mathcal{S}_{u'}$ is the subgraph of perturbed node u'; $\max D$ represents the cosine distance metric, $\mathbf{M}_{\mathcal{S}^{p}_{u}}$ and $\mathbf{M}_{\mathcal{S}^{p}_{u'}}$ are the predicted explanation masks for $\mathcal{S}_u$ and $\mathcal{S}_{u'}$; and β represents a δ-radius ball around $\mathcal{S}_u$ for which the model behavior is same.

Fairness. It is the concept that explanations provided by machine learning models should be accurate and fair, and should not perpetuate or amplify existing biases. It promotes transparency, accountability, and fairness in decision-making processes. In the paper [2], authors propose Graph Explanation Counterfactual fairness mismatch (GECF) and Graph Explanation Group Fairness mismatch (GEGF) to evaluate the explanations on the respective datasets. To measure counterfactual fairness, they verify

if the explanations corresponding to $\mathscr{S}_u$ and its counterfactual counterpart are similar if the underlying model predictions are similar. They calculate counterfactual fairness mismatch as:

$$\text{GECF}\left(\mathbf{M}^p, \mathbf{M}_s^p\right) = D\left(\mathbf{M}^p, \mathbf{M}_s^p\right) \tag{2.9}$$

where $\mathbf{M}^p$ and $\mathbf{M}_s^p$ are the predicted explanation mask for $\mathscr{S}_u$ and for the counterfactual counterpart of $\mathscr{S}_u$. $D(\cdot)$ means to computer the difference. It should be noted that they anticipate a decrease in the GECF score for graphs that have ground-truth explanations that exhibit weak forms of unfairness. This is because the explanations for both the original and counterfactual graphs are likely to be similar. In contrast, for graphs with ground-truth explanations that exhibit strong forms of unfairness, we expect to observe an increase in the GECF score. This is because modifying the protected attribute is likely to result in changes to the explanations provided by the model.

They measure group fairness mismatch as follows:

$$\text{GEGF}\left(\hat{\mathbf{y}}_{\mathscr{K}}, \hat{\mathbf{y}}_{\mathscr{K}}^{\mathbf{E}_u}\right) = \| \text{SP}(\hat{\mathbf{y}}_{\mathscr{K}}) - \text{SP}\left(\hat{\mathbf{y}}_{\mathscr{K}}^{\mathbf{E}_u}\right) \| \tag{2.10}$$

where $\hat{\mathbf{y}}_{\mathscr{K}}$ and $\hat{\mathbf{y}}_{\mathscr{K}}^{\mathrm{E}_u}$ are predictions for a set of K graphs using the original and the essential features identified by an explanation, respectively. And SP is the statistical parity. The higher values of GEGF indicate that the explanation does not preserve group fairness.

There are various evaluation metrics, and each one has its respective emphasis and reflects different aspects of an explainable model. One should, therefore, use a combination of multiple metrics to attain reasonable and practical explainable systems. However, as mentioned above, it is also important that one should take the characteristics of datasets and explainable methods into account in order to choose suitable evaluation metrics.

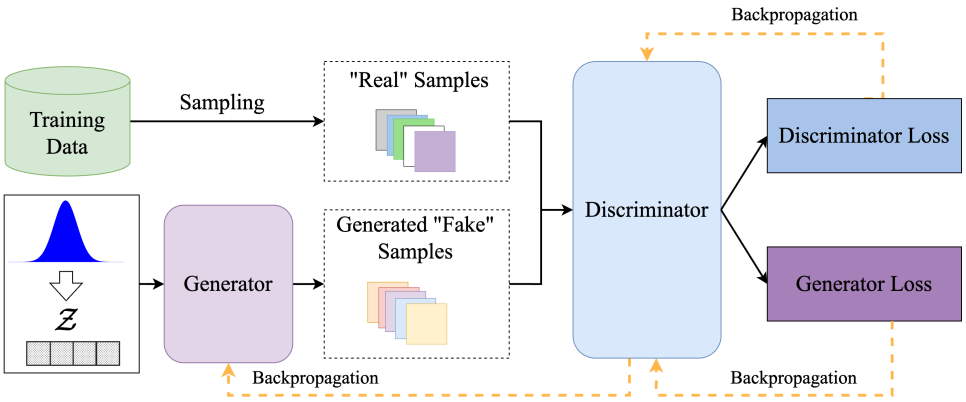

Figure 2.3: The Example of GAN Architecture.

2.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [20] are composed of two neural networks: a generator and a discriminator, trained in a game-like manner. The structure can be found in Figure 2.3. 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 distinguish between real and fake samples accurately. 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 [44] proposes providing both the generator and discriminator with class labels to produce class conditional samples. Researchers in [63] demonstrate that class conditional synthesis significantly improves the quality of generated samples. An-

other 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. [11] 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 [45], 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. [46] 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 [52] propose a semi-supervised image classifier based on ACGAN. Waheed et al. [66] apply ACGAN in medical image analysis. Furthermore, in [84], authors

augment the data by applying ACGAN in the electrocardiogram classification system. Ding et al. [15] propose a tabular data sampling method that integrates the Knearest neighbour method and tabular ACGAN to balance normal and attack samples.

2.4 Summary

In the literature review chapter, we provided an overview of contemporary techniques for elucidating GNNs and the criteria employed for assessing the effectiveness of GNN explainers. Following this, we delved into an exposition of pertinent literature concerning GANs, which serves as the generative framework underpinning our proposed methodology.

CHAPTER 3

GAN-based Explainer for 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 dependency on specific instances, limited generalizability to unseen graphs, the potential for producing invalid explanations, and restrictions to specific tasks like node or graph classification. To overcome these limitations, we, in this work, introduce the Generative Adversarial Networks (GANs) [20] into the field of GNN explanation and propose a new GNN explainer dubbed *GAN-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 quality 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.

3.1 Introduction

GNNs have swiftly progressed as a powerful method for processing graph-structured data, showing outstanding performance across various real-world applications, including crime prediction [68], traffic flow estimation [26], event forecasting [14], 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 [58], recommendation systems [43], 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) [51, 64].

The explanation of GNN has attracted substantial scholarly interest, and many explainers [24, 37, 41, 75, 78] 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 limit to a specific task (e.g., node classification, graph classification, etc.). In particular, the seminal method GNNExplainer [75] limits itself to *local* explanation and lacks the *generalizability*. After that, XGNN [78], 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 the *generalizability*, and worse, it may generate some nonexisting important subgraphs. Recent Gem [37] 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 work introduces a novel GNN explainer, *GAN-GNNExplainer*, which uses the generative method to produce explanations for GNNs. Our method consists of a generator and a discriminator. In particular, the generator learns to produce explanations for the input graph $\mathbf{G}$, 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.

Key contributions of this work include:

- We introduce a novel explainer, **GAN-GNNExplainer**, specifically designed for GNN models. This method leverages a generator to produce explanations guided by a discriminator to ensure reliability and consistency throughout the process.

- Our methods are rigorously evaluated on diverse graph datasets, including both synthetic and real-world data. The results consistently demonstrate the superiority of our approach compared to existing methods.

3.2 Method

3.2.1 Problem Formulation

The concepts of "Interpret" and "Explain" are crucial in understanding how GNNs generate predictions. Interpretation involves comprehending how the model arrived at its decision, focusing on transparency and the ability to track the decision-making process. In contrast, an explanation provides a rationale or justification for the GNN's prediction, aiming to offer clear and concise reasoning for the results.

In this study, our primary objective is to identify the specific subgraph within a test graph that has a significant influence on the GNN's prediction for that particular test graph. To generate explanations that elucidate the reasoning behind the GNN's predictions, we train our GNN explainer using the real explanation subgraphs (or explanation ground truth) we obtained.

A GNN explainer generates a faithful and compact subgraph that identifies essential features, providing clues as to why the GNN model makes its predictions. The explained subgraph must be a valid subgraph for the original input graph, containing a subset of the vertices and edges from the input graph. Ultimately, our aim is to improve explainability in machine learning models, which is critical for their adoption in real-world applications.

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 $\mathscr{F}$ and Y denotes its predictions, $\mathscr{F}(\mathbf{G}) \rightarrow y$. We further define $\mathscr{E}(\mathscr{F}(\mathbf{G}), \mathbf{G})) \rightarrow \mathbf{G}^s$ as the explanation of a GNN explainer. Ideally, when feeding the explanation into the GNN model $\mathscr{F}$, it would produce the exact same prediction y, which means that $\mathscr{F}(\mathbf{G})$ equals $\mathscr{F}(\mathscr{E}(\mathscr{F}(\mathbf{G}), \mathbf{G}))$. We also expect that the explanation $\mathscr{E}(\mathscr{F}(\mathbf{G}), \mathbf{G})) \rightarrow \mathbf{G}^s$ 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.

3.2.2 Obtaining Causal Real Explanations

Our objective in this work is to elucidate the reasoning behind the predictions made by the target GNN model $\mathscr{F}$. To achieve this, we regard the target GNN model $\mathscr{F}$ 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 $\mathscr{F}$. 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 work, we employ Granger causality [21], 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 $\mathscr{F}$. 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 $\mathscr{F}$. 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 [21] directly to explain a target GNN model $\mathscr{F}$ 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.

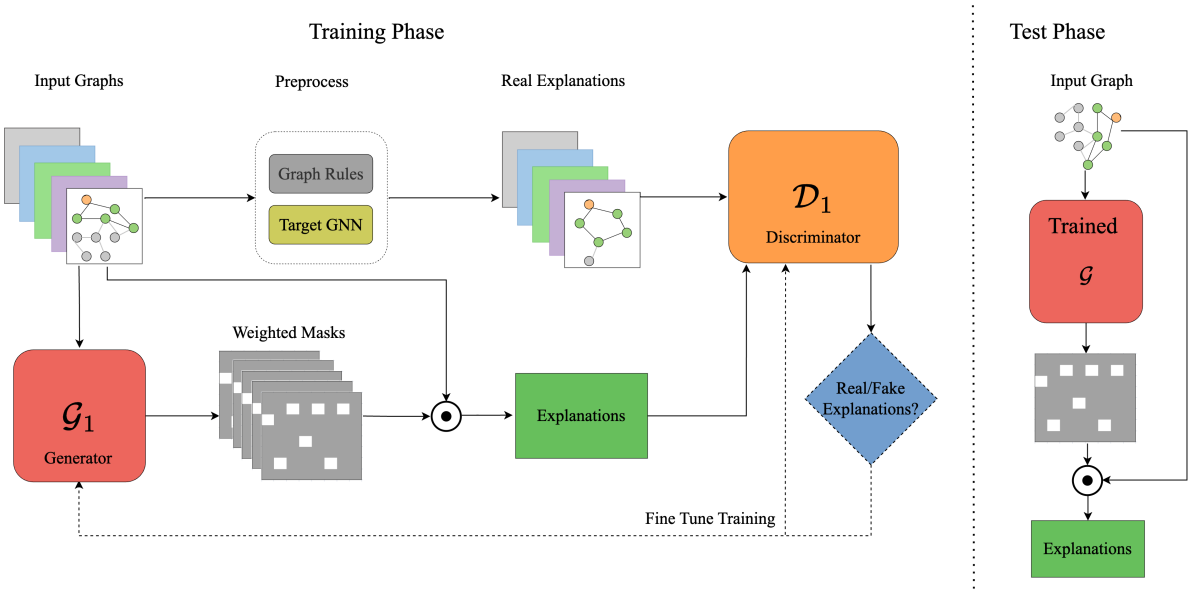

Figure 3.1: The Framework of GAN-GNNExplainer. The ⊙ means element-wise multiplication. This figure is comprised of two phases: Training and Test. During the Training Phase, the objective is to train the generator and discriminator components of the GAN-GNNExplainer model. Following successful training, the Test Phase then utilizes the trained generator to generate explanations for the testing data.

3.2.3 GAN-GNNExplainer

By leveraging the GAN's generating capacity, in this work, we propose a GAN-based explanation method for GNN, which is called GAN-GNNExplainer. It consists of a generator ($\mathcal{G}_1$) and a discriminator ($\mathcal{D}_1$), which is depicted in Figure 3.1.

Unlike the typical way of training GAN where random noise $\mathbf{z}$ is fed into the generator $\mathcal{G}$, in our model, we feed $\mathcal{G}$ with the original graph $\mathbf{G}$ 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 $\mathbf{G}$. In addition, the generator $\mathcal{G}$ trained under this mechanism can be easily generalized 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 would project the original input graph $\mathbf{G}$ into a compact hidden representation and then the decoder would reconstruct 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, $\mathscr{G}_1$ 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 $\mathscr{D}_1$ is introduced. $\mathscr{D}_1$ 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 $\mathscr{G}_1$ and $\mathscr{D}_1$, we first need to obtain the "real" explanations. This is done through a pre-processing step in our framework (Figure 3.1), where Granger causality generates the "real" explanations as ground truth for training the discriminator. Details of this process can be found in Section 3.2.2. Once the input graph $\mathbf{G}$ and corresponding subgraph are identified, the model is trained to generate a weighted mask that emphasizes the important edges and nodes in $\mathbf{G}$ that play a key role in the decision-making process of the GNN model $\mathscr{F}$. 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.

3.2.4 Improved Loss Function

In a GAN framework, the generator and discriminator 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 (3.1):

$$\min_{\mathscr{G}} \max_{\mathscr{D}} \quad \mathbb{E}_{\mathbf{G}^{gt} \sim p_{(\mathbf{G}^{gt})}}[\log \mathscr{D}(\mathbf{G}^{gt})] + \mathbb{E}_{\mathbf{G} \sim p_{(\mathbf{G})}}[\log(1 - \mathscr{D}(\mathscr{G}(\mathbf{G})))], \tag{3.1}$$

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

When we simply adopt Equation (3.1) as our objective function to train our $\mathscr{G}_1$ and $\mathscr{D}_1$ simultaneously, we empirically observe that the accuracy of the final explanation is not optimistic. We suppose it is because Equation (3.1) 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 (3.2).

$$\begin{aligned} \min_{\mathscr{G}_1} \max_{\mathscr{D}_1} \quad & \mathbb{E}_{\mathbf{G}^{gt} \sim p_{(\mathbf{G}^{gt})}}[\log \mathscr{D}_1(\mathbf{G}^{gt})] \\ & + \mathbb{E}_{\mathbf{G} \sim p_{(\mathbf{G})}}[\log(1 - \mathscr{D}_1(\mathscr{G}_1(\mathbf{G})))] \\ & + \lambda \frac{1}{N} \sum_{i=1}^{N} (f(\mathbf{G}) - f(\mathscr{G}_1(\mathbf{G})))^2, \end{aligned} \tag{3.2}$$

where $\mathscr{F}$ denotes a pre-trained target GNN model, N represents the node set of $\mathbf{G}$, and $\mathbf{G}$ is the input graph we aim to explain, while $\mathbf{G}^{gt}$ 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 $\mathscr{F}$. If λ is set to zero, Equation (3.2) becomes identical to Equation (3.1).

3.2.5 Pseudocode of GAN-GNNExplainer

GAN-GNNExplainer is a powerful method for providing explanations for the predictions generated by GNNs. To better comprehend the methodology behind GAN-GNNExplainer, we present the following pseudo-code Algorithm 1.

Algorithm 1: Training a GAN-GNNExplainer

Input: Training graph data $G = \{g_1, \cdots, g_n\}$, real explanations for training graphs $G^{gt} = \{g_1^{gt}, \cdots, g_n^{gt}\}$ (obtained in preprocess phase), trained GNN $\mathscr{F}$.

Output: Generator $\mathscr{G}_1$, discriminator $\mathscr{D}_1$.

1 Initialize $\mathscr{G}_1$ and $\mathscr{D}_1$ with random weights.

2 **for** *epoch in epochs* **do**

3 Sample a minibatch of m real data samples $\{g^{(1)}, \cdots, g^{(m)}\}$ from training data G

4 Generate fake data samples $\{\tilde{g}^{(1)}, \cdots, \tilde{g}^{(m)}\} = G(g^{(1)}, \cdots, g^{(m)})$

5 Update $\mathscr{D}_1$ by taking m gradient steps on the objective function in order to maximize its classification ability:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log(\mathscr{D}_1(g^{gt(i)})) + \log(1 - \mathscr{D}_1(\tilde{g}^{(i)})) \right]$$

Update $\mathscr{G}_1$ by taking m gradient steps on the objective function in order to confuse the $\mathscr{D}_1$ maximally:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - \mathscr{D}_1(\tilde{g}^{(i)})) + \left(\mathscr{F}(g^{(i)}) - \mathscr{F}(\tilde{g}^{(i)}) \right)^2$$

6 **end**

Once the input graph and target GNN have been acquired, the GAN-GNNExplainer can be trained to produce a weighted mask that effectively highlights the edges and nodes in the original graph and significantly contributes to the GNN's decision-making process. By multiplying the mask with the original adjacency matrix of the input graph, we can obtain the corresponding explanation. These explanations are particularly useful for understanding the reasoning behind the GNN's predictions and identifying the salient features of the input graph.

Table 3.1: 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

3.3 Experiments

In this section, we conduct experiments to evaluate the performance of our GAN-GNNExplainer. We first describe the datasets we used and our implementation details in Section 3.3.1. After that, we present and analyze the experimental results on synthetic datasets in Section 3.3.2 and real-world datasets in Section 3.3.3.

3.3.1 Experimental Settings

Datasets. We focus on two widely used synthetic node classification datasets, including BA-Shapes and Tree-Cycles [75], and two real-world graph classification datasets, Mutagenicity [27] and NCI1 [67]. Details of these datasets are shown in Table 3.1.

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 8-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 categorized into two classes:

non-mutagenic 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 non-mutagenic graphs, rendering them non-discriminative.

NCI1 is a curated subset of chemical compounds evaluated for their efficacy against non-small cell lung cancer. It encompasses over 4,000 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.

Baseline Approaches. With the wide application of GNNs, more and more GNN explainers have been proposed to address the problem of explaining GNN models. *GNNExplainer* is a seminal method in explaining GNN models. In addition, the PGExplainer and Gem are more related to our method. However, Gem has shown its superiority over PGExplainer methods. Thus, we only consider GNNExplainer and Gem as alternative approaches. We set all the hyperparameters of the baselines as reported in the corresponding papers.

Different Top Edges (K or R). After obtaining the weight (importance) of each edge for the input graph $\mathbf{G}$, 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 Ratio (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 $K = \{5, 6, 7, 8, 9\}$ 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. We maintain the consistency of the testing data. Both training data and validation data are used in their entirety during the training process. The value of K/R and the split ratio of the data set are consistent with the experimental settings of Gem so that we can fairly compare our results to those of Gem, which are our baseline.

Evaluation Metrics. A better explainer should be able to generate more compact subgraphs yet maintain the prediction accuracy while the associated explanations are fed into the target GNN. After comparing the characteristics of each metric [35], we chose quantitative and qualitative evaluations. In particular, we generate explanations for the test set using *GNNExplainer* [75], *Gem* [37], *OrphicX* [38], and ACGAN-GNNExplainer (our method), respectively. We then feed these generated explanations to the pre-trained target GNN $\mathscr{F}$ to compute the accuracy, which can be formally defined as Equation (3.3):

$$ACC_{exp} = \frac{|\mathscr{F}(\mathbf{G}) = \mathscr{F}(\mathbf{G}^s)|}{|T|} \tag{3.3}$$

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

3.3.2 GAN-GNNExplainer on Synthetic Datasets

Firstly, we conduct experiments on synthetic datasets, including BA-Shapes and Tree-Cycles. We evaluate the accuracy of explanations provided by GNNExplainer, Gem,

Table 3.2: The Accuracy of Explanations on the BA-Shapes Dataset.

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 3.3: The Accuracy of Explanations on the Tree-Cycles Dataset.

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**

and GAN-GNNExplainer (our model). We also present quantitative and qualitative evaluations of our experiments.

Quantitative Analysis. The accuracy of explanations for synthetic datasets with various K settings is detailed in Table 3.2 and Table 3.3. The results indicate that GAN-GNNExplainer consistently provides the most accurate explanations in all cases. On the BA-Shapes dataset, GNNExplainer, Gem, and GAN-GNNExplainer perform well for synthetic datasets. However, GAN-GNNExplainer also incorporates a number of enhancements. GAN-GNNExplainer outperforms GNNExplainer and Gem on BA-Shapes. On the Tree-Cycles dataset, GAN-GNNExplainer performs well, whereas GNNExplainer and Gem cannot get good explanations.

It should be noted that while our experiments were conducted on the relatively simple BA-Shapes dataset, the performance of the Gem may be favourable when K is relatively low. However, even when we increase the value of K, the accuracy of the Gem remains unchanged. In contrast, our model's accuracy improves as more information is provided, that is, as the value of K increases. This distinction between the Gem and our model is

particularly relevant when dealing with more complex datasets. In such cases, our model may require more information to achieve optimal performance, but it may ultimately yield higher accuracy than the Gem.

Qualitative Analysis. Qualitative evaluation is an effective way to compare the explanations. We obtain the difference in explanations among GNNExplainer, Gem, and GAN-GNNExplainer by qualitative analysis. We visualize the explanations of BA-Shapes when $K = [8,9]$, and Tree-Cycles when $K = 8$, as shown in Figure 3.2 and Figure 3.3, respectively. In particular, we select two types of nodes whose predictions of GNN are correct or incorrect, respectively, for visualization. In Figure 3.2, the first row is the visualization of nodes where GNN makes a correct prediction. That is, $\mathscr{F}(\mathbf{G})$ equals to the node label $\mathbf{L}_{\mathbf{G}}$. We can find the visualization of nodes where GNN makes an incorrect prediction in the second row. That is, $\mathscr{F}(\mathbf{G})$ is different from the label of the node $\mathbf{L}_{\mathbf{G}}$. For the explanations, we expect to get the same prediction as the original adjacency matrix prediction of GNN. However, after getting the explanations, we fed the explanations into the target GNN, and we got different predictions for different explanations. We note that the explanation of Gem ($\mathbf{G}^s_{gem}$) gives the wrong prediction. While the explanations of GNNExplainer ($\mathbf{G}^s_{gnnex}$) and GAN-GNNExplainer ($\mathbf{G}^s_{ganex}$) get the correct prediction. That is, $\mathscr{F}(\mathbf{G}^s_{gem})$ is distinct from $\mathscr{F}(\mathbf{G})$, while $\mathscr{F}(\mathbf{G}^s_{gnnex})$ and $\mathscr{F}(\mathbf{G}^s_{ganex})$ are equivalent to $\mathscr{F}(\mathbf{G})$. Thus, we observe that the GNNExplainer and GAN-GNNExplainer provide accurate explanations.

Similarly, Figure 3.3 visualizes, in the first and second rows, the explanations of various Tree-Cycles nodes whose GNN predictions are incorrect and correct, respectively. We observe that Gem is unable to obtain accurate explanations for nodes that GNN predicts incorrectly or correctly, whereas GAN-GNNexplainer does. To put it another way, when node $\mathscr{F}(\mathbf{G})$ differs from the label of node $\mathbf{L}_{\mathbf{G}}$, we achieve $\mathscr{F}(\mathbf{G}^s_{ganex})$ being equivalent to $\mathscr{F}(\mathbf{G})$, while $\mathscr{F}(\mathbf{G}^s_{gem})$ and $\mathscr{F}(\mathbf{G}^s_{gnnex})$ differ from $\mathscr{F}(\mathbf{G})$. Furthermore,

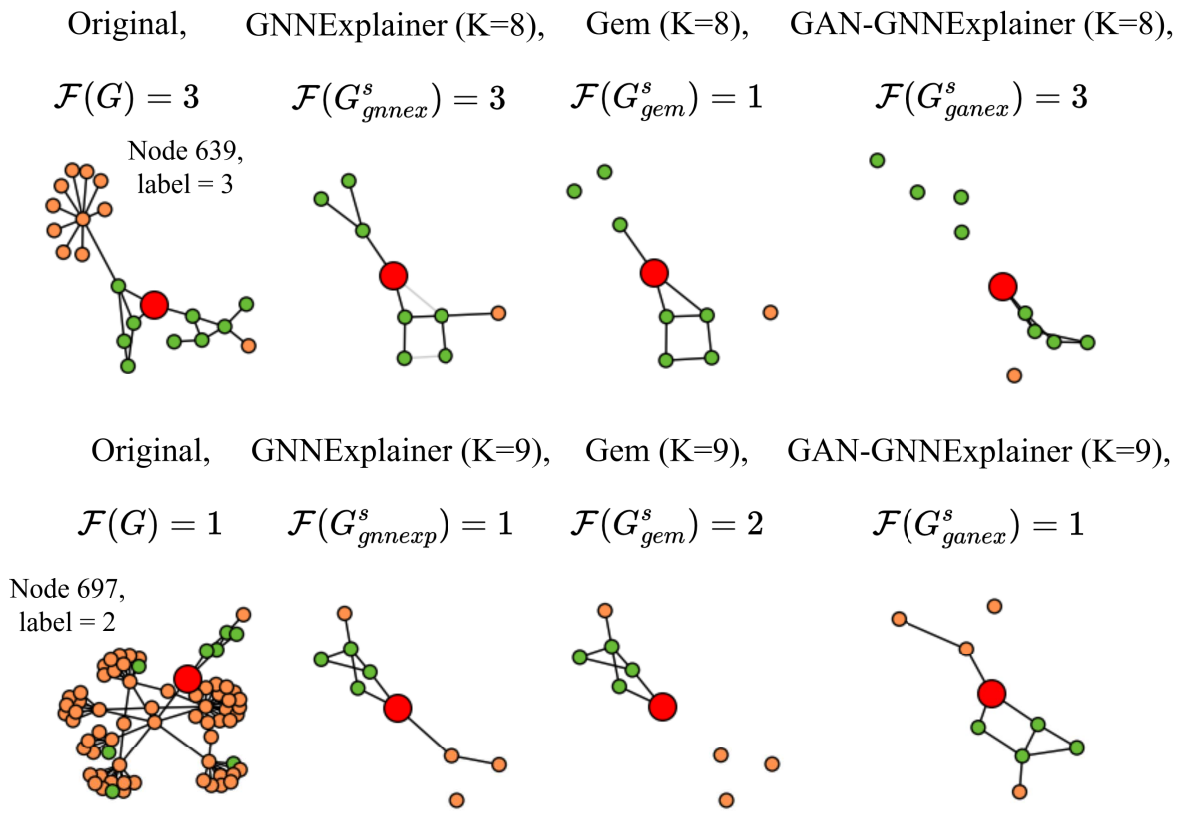

Figure 3.2: The Explanation Visualization on BA-Shapes, When K = {8,9}. The 1^{st} column is the original graph structure and the prediction by the pre-trained GNN. The 2^{nd} column to the 4^{th} column is explanations from GNNExplainer, Gem, and GAN-GNNExplainer, respectively when K is set to be 8 (the 1^{st} row) and 9 (the 2^{nd} row). When K is set to be 8 (the 1^{st} row), the GNN prediction for explanations from GNNExplainer and GAN-GNNExplainer are identical to the original graph while the GNN prediction for explanations from Gem is wrong. When K is set to be 9 (the 2^{nd} row), a similar pattern is observed as the K of 8.

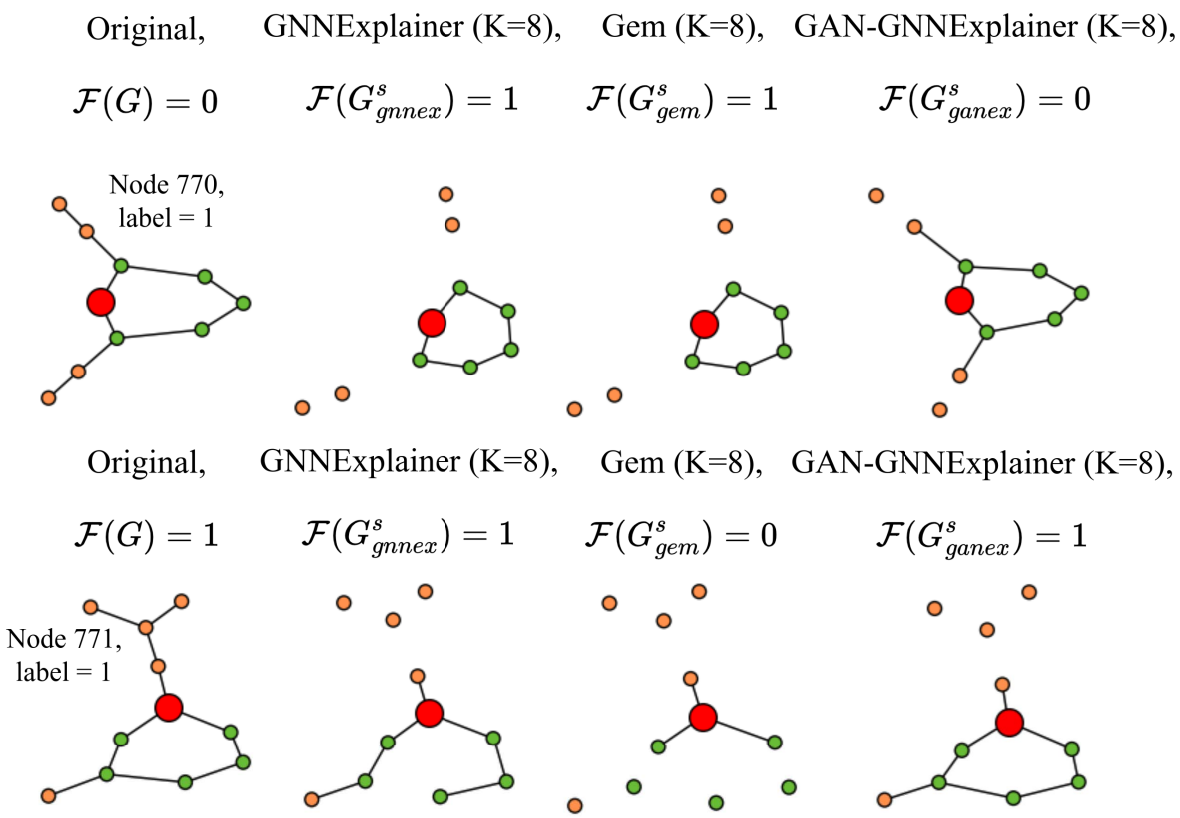

Figure 3.3: The Explanation Visualization on Tree-Cycles, When $K = 8$. The first column is the original graph structure that the GNN predicts. The second through fourth columns contain the respective explanations from GNNExplainer, Gem, and GAN-GNNExplainer.

for these that $\mathscr{F}(\mathbf{G})$ equal to the label of the node $\mathbf{L_G}$, we find that $\mathscr{F}(\mathbf{G}^s_{gnnex})$ and $\mathscr{F}(\mathbf{G}^s_{ganex})$ are equivalent to $\mathscr{F}(\mathbf{G})$, while $\mathscr{F}(\mathbf{G}^s_{gem})$ is distinct from $\mathscr{F}(\mathbf{G})$. Thus, the GAN-GNNExplainer precisely explains the GNN model.

3.3.3 GAN-GNNExplainer on Real-world Datasets

This subsection reports the experimental results with real-world datasets. The quantitative evaluation is shown in Table 3.4 and Table3.5. 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.

Table 3.4: The Accuracy of Explanations on the Mutagenicity Dataset.

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 3.5: The Accuracy of Explanations on the NCI1 Dataset.

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

Quantitative Analysis. 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.

Qualitative Analysis. To further check the explainability of the generated explanations, we report the qualitative evaluation of Mutagenicity and NCI1, setting $K = 15$ in Figure 3.4 and Figure 3.5, respectively. We visualize the explanations of these selected nodes where GNN predicts correctly and incorrectly, respectively. When the target GNN gets the prediction of the graph, we expect to get the same prediction for the explanation. In particular, when the GNN makes an incorrect prediction (e.g. the second-row in Figure 3.4), we want to explain why the target GNN makes the incorrect prediction. Thus, we visualize the explanations from GNNExplainer, Gem, and GAN-GNNExplainer,

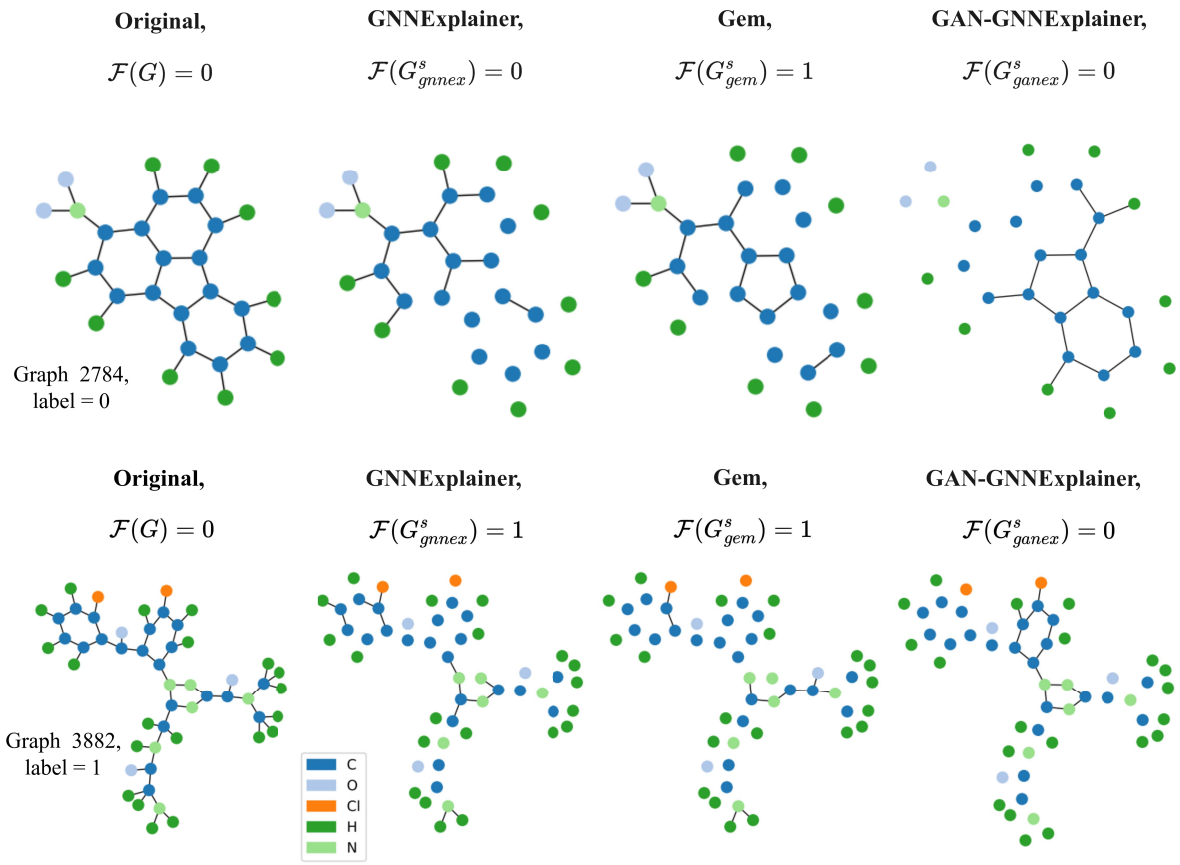

Figure 3.4: The Explanation Visualization on Mutagenicity, When $K = 15$. The 1^{st} column represents the initial graph structure from which the GNN makes a prediction. The explanations from GNNExplainer, Gem, and GAN-GNNExplainer are displayed in the 2^{nd} through 4^{th} columns, respectively.

respectively. From Figure 3.4, we note that the explanations from GNNExplainer and GAN-GNNExplainer get the same prediction as the original graph, after feeding the explanations into the target GNN. However, the explanation from Gem makes a different prediction. Thus, we can conclude that the explanations provided by GNNExplainer and GAN-GNNExplainer are correct, while the explanation from Gem is incorrect. Furthermore, comparing the explanation of GNNExplainer and GAN-GNNExplainer, we note that the explanation of GAN-GNNExplainer provides a more integrated explanation. Specifically, in Figure 3.5, we observe that for the graph that GNN correctly predicts, the GNNExplainer and GAN-GNNExplainer get correct explanations. Furthermore, for

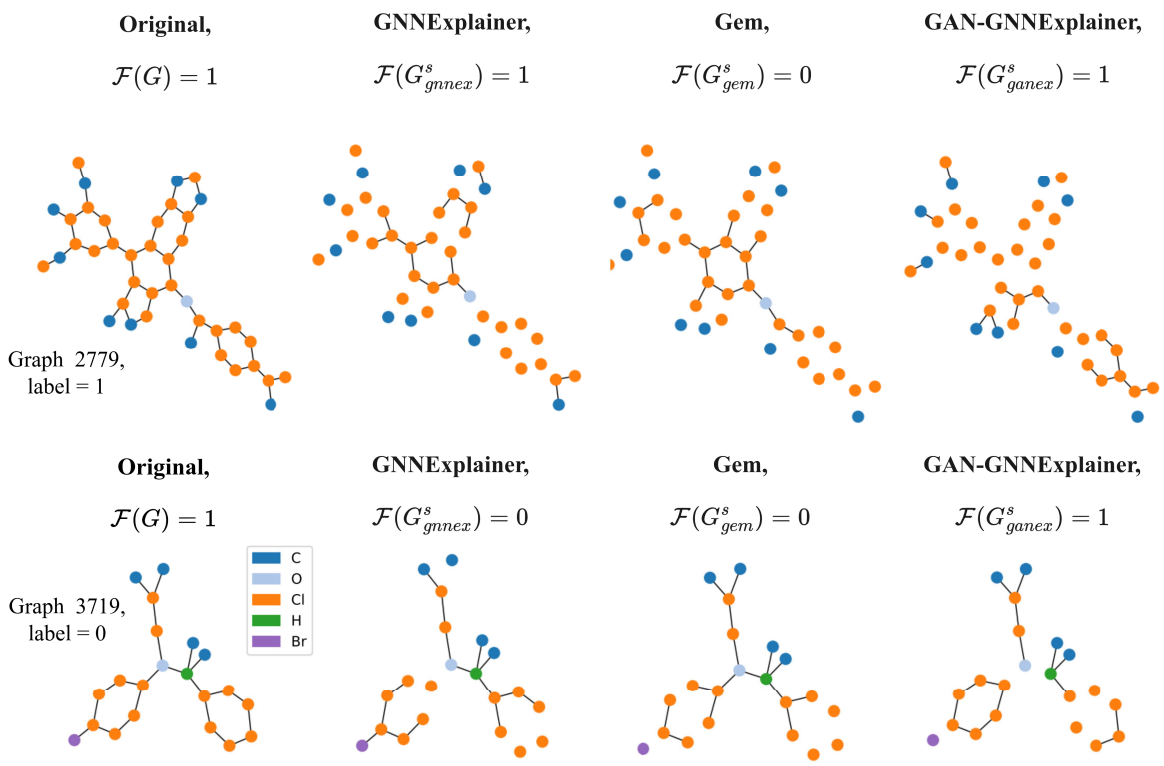

Figure 3.5: The Explanation Visualization on NCI1, When $K = 15$. The 1^{st} column contains the initial graph structure that the GNN predicts. The explanations from GNNExplainer, Gem, and GAN-GNNExplainer are located in the 2^{nd} through 4^{th} columns, respectively.

another graph in that GNN makes an incorrect prediction, only the GAN-GNNExplainer obtains a correct explanation. To put it another way, in the case where $\mathscr{F}(\mathbf{G})$ equals to the graph label, we note that $\mathscr{F}(\mathbf{G}^s_{gnnex})$ and $\mathscr{F}(\mathbf{G}^s_{ganex})$ correspond to $\mathscr{F}(\mathbf{G})$, while $\mathscr{F}(\mathbf{G}^s_{gem})$ is distinct from $\mathscr{F}(\mathbf{G})$. However, when $\mathscr{F}(\mathbf{G})$ is different from the graph label, we find that only $\mathscr{F}(\mathbf{G}^s_{ganex})$ is equivalent to $\mathscr{F}(\mathbf{G})$, while $\mathscr{F}(\mathbf{G}^s_{gnnex})$ and $\mathscr{F}(\mathbf{G}^s_{gem})$ is distinct from $\mathscr{F}(\mathbf{G})$. Thus, we obtain precise explanations from the GAN-GNNExplainer for the GNN.

3.4 Limitations and Discussions

As highlighted in Section 3.1, GAN-GNNExplainer represents a notable advancement in the accuracy of GNN explainability, successfully addressing some limitations of current

popular GNN explainers. However, several challenges still require further investigation:

Insufficient Reliability on Real-world Datasets. Performance of GAN-GNNExplainer falls short when applied to real-world datasets, lacking the necessary accuracy and reliability to produce meaningful results. This deficit in accuracy poses significant challenges in practical applications, where the discrepancy between model predictions and real-world outcomes can lead to ineffective decision-making and compromised solutions. Addressing this issue requires a concerted effort to enhance the model's robustness through rigorous data collection, preprocessing, and algorithmic refinement, ensuring its viability and trustworthiness in real-world scenarios.

Absence of Fidelity. In our evaluation criteria, fidelity isn't treated as a measure of performance. While fidelity typically denotes faithfulness or accuracy in various contexts, such as relationships or data modelling, we prioritize other aspects in our assessments. This approach suggests that our focus lies elsewhere, perhaps emphasizing different metrics or qualities deemed more pertinent to the task or objective at hand.

Therefore, future work could concentrate on creating a model that can generate faithful and accurate explanations, which will improve its performance on real-world datasets and make it more reliable and faithful. Finally, it is worth considering the possibility of training the interpreter using a simulated dataset and subsequently applying the trained interpreter to interpret real datasets, which could hold greater practical significance.

3.5 Summary

Explaining the underlying work mechanism of a GNN is crucial for increasing confidence in the model's predictions, assuring the security of real-world applications, and facilitating the development of trustworthy GNNs. To achieve these goals, many recent approaches have been proposed in recent years. While they function relatively well

in some respects, they suffer from limitations in different aspects. To mitigate these limitations, in this work, we proposed GAN-based explanations for GNNs, dubbed GAN-GNNExplainer. It is composed of 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 of the generated explanations. To verify the effectiveness of our proposed method, we experimented with our approach on synthetic and real-world graph datasets and conducted qualitative and quantitative comparisons with other popular GNN explainers. The experimental results demonstrated the superiority of our proposed method.

CHAPTER 4

ACGAN-BASED EXPLAINER FOR GRAPH NEURAL NETWORKS

To address the shortcomings observed in our *GAN-GNNExplainer*, specifically its inadequate reliability on real-world datasets and lack of fidelity, we introduce *ACGAN-GNNExplainer*, which leverages the Auxiliary Classifier Generative Adversarial Network (ACGAN) [46] as its backbone to generate explanations for GNNs. This novel approach is distinguished by four key attributes: global scope explanation, enhanced generalizability, versatility across different tasks, and high fidelity. ACGAN-GNNExplainer employs a generator to create explanations while integrating a discriminator to supervise the generation process. Specifically, the input graph $\mathbf{G}$, along with its corresponding label $\mathscr{F}(\mathbf{G})$ (determined by the target GNN model $\mathscr{F}$), is fed into the generator, which learns to produce explanations. To ensure the validity and accuracy of the generated subgraph, the discriminator distinguishes between "real" and generated explanations, assigns a prediction label to each, and provides feedback to the generator, thereby reinforcing fidelity and enhancing accuracy. Extensive experimentation on synthetic and real-world datasets demonstrates the effectiveness of ACGAN-GNNExplainer, showcasing its superiority

over existing GNN explainers.

4.1 Introduction

Due to GNNs ability to capture complex relationships between nodes and extract meaningful features from graphs, they have emerged as a powerful tool for modelling graph-structured data and a natural choice for a variety of real-world applications. 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 [75], XGNN [78], and PGExplainer [41] 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 contributed valuable insights into GNNs; however, significant challenges remain in the following areas:

- *Explanation Scale*: This refers to the scope of the explanation,Äîwhether it focuses on specific instances for fine-grained insights or captures overarching patterns that generalize across similar instances. Ideally, explanations should balance these perspectives, encompassing both instance-specific details and broader patterns shared among similar cases.

- *Generalizability*: A generalizable explainer should deliver effective explanations for unseen graphs or tasks without requiring retraining. This adaptability ensures consistent performance across diverse datasets and scenarios, enhancing usability and efficiency in dynamic environments.

- *Fidelity*: Explanations must accurately reflect the truly influential subgraphs within the input graphs, faithfully representing the decision-making process of the

underlying model. High-fidelity explanations accurately reflect the decision-making processes of the underlying model, bolstering their reliability and trustworthiness.

- *Versatility*: This is the ability of an explainer to provide reliable and insightful explanations across a variety of tasks, such as node classification, graph classification, and link prediction. A versatile explainer demonstrates robustness and adaptability, ensuring applicability to a wide range of graph analysis scenarios.

The pioneering method GNNExplainer [75], for instance, is limited to local explanation and lacks generalizability. Later, XGNN [78] addressed this limitation but still lacks generalizability. Recent Gem [37] 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 work, propose a new GNN explanation method dubbed ACGAN-GNNExplainer, which uses the auxiliary classifier Generative Adversarial Network (ACGAN) [46] as its backbone to generate explanations for GNNs. 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 $\mathscr{F}(\mathbf{G})$, which is determined by the target GNN model $\mathscr{F}$. In addition, a discriminator is adopted to distinguish whether the generated explanations are "real" or "fake" and to designate a prediction label for 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 [52], data augmentation [84], medical

image analysis [66], 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:

- Global-scale Explanations: The method captures the underlying patterns of graphs, enabling it to naturally provide explanations at a global level.

- Generalizability: After capturing the underlying patterns, it can generate explanations for unseen graphs without requiring retraining.

- High-fidelity Explanations: The consistent monitoring of the discriminator increases the likelihood of identifying valid and significant subgraphs.

- Task Versatility: The method performs effectively across various tasks, including node classification and graph classification.

Our main contributions to this work 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.

4.2 Method

4.2.1 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, emphasizing the transparency and traceability of decisions, explanation supports GNN predictions by providing a logical and coherent rationale for the observed outcomes.

As previously defined, a graph is represented as $\mathbf{G} = (\mathbf{V}, \mathbf{A}, \mathbf{X})$, with a GNN model $\mathscr{F}$ producing predictions $\mathscr{F}(\mathbf{G}) \rightarrow Y$. We also use $\mathscr{E}(\mathscr{F}(\mathbf{G}), \mathbf{G}) \rightarrow \mathbf{G}^s$ to denote the explanation, which satisfies $\mathscr{F}(\mathbf{G}) = \mathscr{F}(\mathscr{E}(\mathscr{F}(\mathbf{G}), \mathbf{G}))$ and ensures $\mathbf{G}^s \in \mathbf{G}$ as a valid subgraph. (Refer to Section 3.2.1 for detailed definitions.)

4.2.2 Obtaining Causal Real Explanations

The aim of our study is to uncover the rationale behind the predictions produced by the target GNN model $\mathscr{F}$. Instead of delving into the intricate internal mechanisms of $\mathscr{F}$, we treat it as a black box. Our focus lies in identifying the subgraphs that exert a significant influence on predictions of $\mathscr{F}$. To accomplish this, we employ a generative model ACGAN to autonomously produce these subgraphs (explanations). For the generative model to generate accurate explanations, it must first undergo training with "real" explanations (ground truth). However, obtaining such ground truths is often unfeasible in real-world scenarios. To address this challenge, we adopt the approach utilised in GAN-GNNExplainer (Section 3.2.2), leveraging Granger causality [21] - a method commonly used to assess causal relationships between variables. This allows us to overcome the absence of ground truth explanations and proceed with our investigation.

4.2.3 ACGAN-GNNExplainer

Using the generating capacity of ACGAN, in this work, we propose an ACGAN-based explanation method for GNN models, which is termed ACGAN-GNNExplainer. It consists of a generator ($\mathscr{G}_2$) and a discriminator ($\mathscr{D}_2$). The generator $\mathscr{G}_2$ is used to generate the explanations, while the discriminator $\mathscr{D}_2$ is used to monitor the generation process. The detailed framework of our method ACGAN-GNNExplainer is depicted in Figure 4.1. In

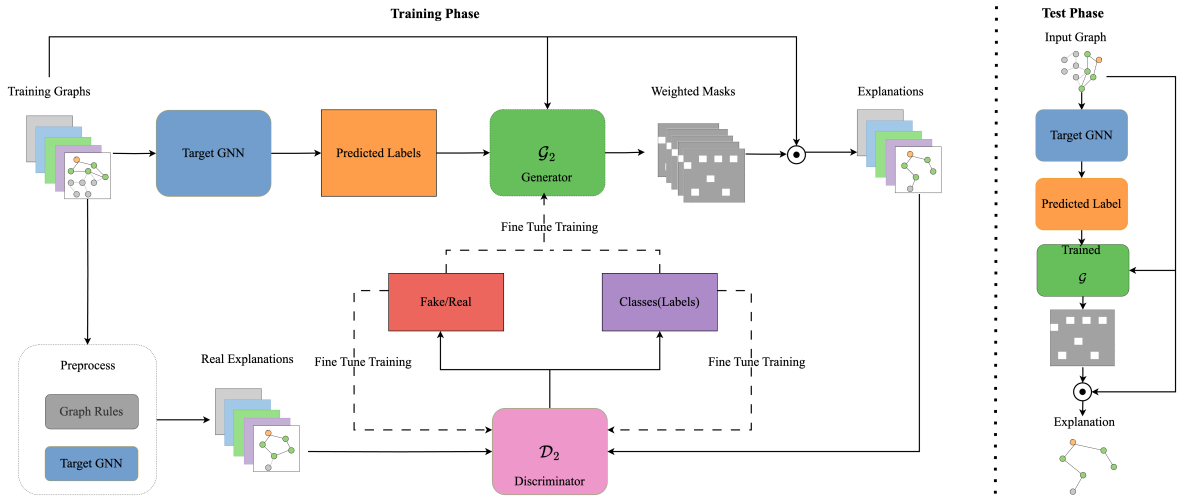

Figure 4.1: The Framework of ACGAN-GNNExplainer. The $\odot$ means element-wise multiplication. This figure includes two phases: the training phase and the test phase. During the Training Phase, the objective is to train the generator and discriminator of the ACGAN-GNNExplainer model. After successful training, the Test Phase then utilizes the trained generator to generate explanations for the testing data.

contrast to the conventional strategy of training an ACGAN, in which random noise $\mathbf{z}$ is fed into the generator $\mathscr{G}_2$, our model feeds the generator $\mathscr{G}_2$ 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 $\mathscr{F}$. Employing this strategy, we ensure that the explanation produced by the generator $\mathscr{G}_2$, which plays a crucial role in determining the predictions of the GNN model $\mathscr{F}$, corresponds to the original input graph $\mathbf{G}$. In addition, the generator $\mathscr{G}_2$ trained under this mechanism can be easily generalized to unseen graphs without

significant retraining, thus saving computational costs. For the generator $\mathscr{G}_2$, we employ an encoder-decoder network where the encoder would project the original input graph **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 $\mathscr{G}_2$ 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 $\mathscr{D}_2$ to monitor the generating process of $\mathscr{G}_2$. Specifically, our discriminator $\mathscr{D}_2$ is a graph classifier with five convolutional layers. It is fed with the real explanation and the explanation generated by our generator $\mathscr{G}_2$. 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 $\mathscr{G}_2$ and further encourages the generator $\mathscr{G}_2$ to produce faithful explanations.

In addition, in order to train our generator $\mathscr{G}_2$ and discriminator $\mathscr{D}_2$, we need to obtain the "real" explanations first. To achieve this goal, we incorporate pre-processing in our framework (Figure 4.1), which uses the Granger causality [21] to acquire the "real" explanations. The details can be found in Section in Section 4.2.2. Once the input graph **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 **G** that significantly contributes to the decision-making process of the given GNN model $\mathscr{F}$. 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.

4.2.4 Improved Loss Function

The generator $\mathscr{G}_2$ produces explanations or subgraphs $\mathbf{G}^s \subseteq \mathbf{G}$ based on two key inputs: the original graph $\mathbf{G}$ and the predicted label Y, expressed as $\mathbf{G}^s \leftarrow \mathscr{G}_2(\mathbf{G}, Y)$. Simultaneously, the discriminator $\mathscr{D}_2$ evaluates both the origin probability $P(S \mid \mathbf{G})$ (whether "real" or generated) and the probability of class classification $P(Y \mid \mathbf{G})$, where Y represents the predicted label of the graph $\mathbf{G}$, denoted as $\mathscr{F}(\mathbf{G}) \rightarrow Y$. The loss function of the discriminator consists of two components: the likelihood of the correct source $\mathscr{L}_S$, as defined in Equation (4.1), and the likelihood of the correct class $\mathscr{L}_Y$, as defined in Equation (4.2).

$$\mathscr{L}_S = \mathbb{E}[\log P(S = \text{ real} \mid \mathbf{G})] + \mathbb{E}\left[\log P\left(S = \text{ generated } \mid \mathbf{G}^{\mathbf{s}}\right)\right], \tag{4.1}$$

$$\mathscr{L}_Y = \mathbb{E}[\log P(Y = L \mid \mathbf{G})] + \mathbb{E}\left[\log P\left(Y = L \mid \mathbf{G}^{\mathbf{s}}\right)\right]. \tag{4.2}$$

where $\mathbf{G}$ means the original graph that requires an explanation, and L means its class label.

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

Conversely, the generator $\mathscr{G}_2$ seeks to minimize the ability of $\mathscr{D}_2$ to distinguish between "real" and generated graphs while simultaneously maximizing its capacity to classify them correctly. This results in a combined objective of maximizing ($-\mathscr{L}_S + \mathscr{L}_Y$). Therefore, based on Equation (4.1) and Equation (4.2), the objective functions for $\mathscr{D}_2$ and $\mathscr{G}_2$ are given in Equation (4.3) and Equation (4.4), respectively.

$$\begin{aligned}\mathscr{L}_{(\mathscr{D}_2)} = &- \mathbb{E}_{\mathbf{G}^{gt}\sim P(\mathbf{G}^{gt})} \log \mathscr{D}_2(\mathbf{G}^{gt}) \\ &- \mathbb{E}_{\mathbf{G}\sim P(\mathbf{G})} \log[1 - \mathscr{D}_2(\mathscr{G}_2(\mathbf{G}, L))] \\ &- \mathbb{E}_{\mathbf{G}^{gt}\sim P(\mathbf{G}^{gt})} P(Y \mid \mathbf{G}^{gt}) \\ &- \mathbb{E}_{\mathbf{G}\sim P(\mathbf{G})} \log(P(Y \mid \mathscr{G}_2(\mathbf{G}, L)),\end{aligned} \tag{4.3}$$

$$\begin{aligned}\mathscr{L}_{(\mathscr{G}_2)} = &- \mathbb{E}_{\mathbf{G}\sim P(\mathbf{G})} \log \mathscr{D}_2(\mathscr{G}_2(\mathbf{G}, L)) \\ &- \mathbb{E}_{\mathbf{G}\sim P(\mathbf{G})} \log P(Y \mid \mathscr{G}_2(\mathbf{G}, L)),\end{aligned} \tag{4.4}$$

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

Using the objective functions from Equation (4.3) and Equation (4.4) to train $\mathscr{D}_2$ and $\mathscr{G}2$, we observe that the fidelity of the generated explanations is unsatisfactory. This may be because the generator loss $\mathscr{L}(\mathscr{G}_2)$, as defined in Equation (4.4), does not explicitly consider fidelity information from the target GNN model $\mathscr{F}$. To resolve this and improve both fidelity and accuracy, we incorporate fidelity directly into the generator,Äôs objective function. Consequently, we derive an enhanced loss function for $\mathscr{G}_2$, as shown in Equation (4.5).

$$\begin{aligned}\mathscr{L}_{(\mathscr{G}_2)} = &- \mathbb{E}_{\mathbf{G}\sim P(\mathbf{G})} \log \mathscr{D}_2(\mathscr{G}_2(\mathbf{G}, L)) \\ &- \mathbb{E}_{\mathbf{G}\sim p(\mathbf{G})} \log P(Y \mid \mathscr{G}_2(\mathbf{G}, L)) \\ &+ \lambda \mathscr{L}_{Fid},\end{aligned} \tag{4.5}$$

$$\mathscr{L}_{Fid} = \frac{1}{N} \sum_{i=1}^{N} ||\mathscr{F}(\mathbf{G}) - \mathscr{F}(\mathscr{G}_2(\mathbf{G}))||^2, \tag{4.6}$$

where $\mathscr{L}_{Fid}$ denotes the component of the loss function that captures fidelity. The symbol $\mathscr{F}$ refers to a pre-trained target GNN model, while N represents the number of nodes

in the graph $\mathbf{G}$, which is the original graph being explained. The term $\mathbf{G}^{gt}$ refers to the ground truth explanation, such as the true important subgraph. The parameter λ is a trade-off hyperparameter that balances the relative importance of the ACGAN model's learning objectives with accuracy of the explanations derived from the pre-trained GNN model $\mathscr{F}$.

Algorithm 2: Training a ACGAN-GNNExplainer

Input: Graph data $\mathbf{G} = \{g_1, \cdots, g_n\}$ with labels $L = \{l_1, \cdots, l_n\}$, real explanations for graph data $\mathbf{G}^{gt} = \{g_1^{gt}, \cdots, g_n^{gt}\}$ (obtained in preprocessing phase), a pre-trained GNN model $\mathscr{F}$.

Output: A well-trained Generator $\mathscr{G}_2$, a well-trained discriminator $\mathscr{D}_2$.

1 Initialize the Generator $\mathscr{G}_2$ and the Discriminator $\mathscr{D}_2$ with random weights

2 **for** *epoch in epochs* **do**

3 Sample a minibatch of m real data samples $\{g^{(1)}, \cdots, g^{(m)}\}$ and real labels $\{l^{(1)}, \cdots, l^{(m)}\}$

4 Generate fake data samples $\{g^{s(1)}, \cdots, g^{s(m)}\} \leftarrow \mathscr{G}_2(g^{(1)}, \cdots, g^{(m)})$ and obtain their labels $\{l^{s(1)}, \cdots, l^{s(m)}\}$

5 Update $\mathscr{D}_2$ with the gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\mathscr{L}_{(\mathscr{D}_2)} \right]$$

6 Update $\mathscr{G}_2$ with the gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[\mathscr{L}_{(\mathscr{G}_2)} \right]$$

7 **end**

4.2.5 Pseudocode of ACGAN-GNNExplainer

In Equations (4.2.3) and (4.2.4), we detailed our framework and loss functions. For further clarity, the pseudocode is provided in Algorithm 2.

4.3 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 4.3.1. After that, we show the quantitative evaluation of our method in comparison with other representative GNN explainers on synthetic datasets (see Section 4.3.2) and real-world datasets (see Section 4.3.3). 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 4.3.4.

4.3.1 Implementation Details

Datasets. In alignment with the methodology employed in GAN-GNNExplainer, in this work, our experimentation also utilizes four datasets: two synthetic datasets, namely BA-Shape and Tree-Cycles, and two real-world datasets, Mutagenicity and NCI1. Comprehensive descriptions of each dataset are available in Section 3.3.1.

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* [75], *OrphicX* [38] and *Gem* [37]. For these competitors, we adopt their respective official implementations.

Different Top Edges (K or R). The parameters K and R are configured identically to those in the GAN-GNNExplainer. Further elaboration on these settings is provided in Section 3.3.1.

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 [35]. In our experiments, we use the accuracy and fidelity of the explanation as our performance metrics. The definition of accuracy is available in Section 3.3.1.

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 $Fidelity^{-}$ 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 (4.7) and Equation (4.8), respectively.

$$Fid^{+} = \frac{1}{N}\sum_{i=1}^{N}(\mathscr{F}(\mathbf{G}_i)_{l_i} - \mathscr{F}(\mathbf{G}_i^{1-s})_{l_i}) \tag{4.7}$$

$$Fid^{-} = \frac{1}{N}\sum_{i=1}^{N}(\mathscr{F}(\mathbf{G}_i)_{l_i} - \mathscr{F}(\mathbf{G}_i^{s})_{l_i}) \tag{4.8}$$

In these equations, N denotes the total number of samples, and l_i represents the class label for instance i. $\mathscr{F}(\mathbf{G}_i)_{l_i}$ and $\mathscr{F}(\mathbf{G}_i^{1-s})_{l_i}$ 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, $\mathscr{F}(\mathbf{G}_i^s)_{l_i}$ 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.

4.3.2 ACGAN-GNNExplainer on Synthetic Datasets

We first conduct experiments on two common synthetic datasets, including BA-Shapes and Tree-Cycles [75], of which the details can be found in Section 3.3.1. We assess the fidelity and accuracy of the explanations generated by GNNExplainer, Gem, OrphicX, and our proposed ACGAN-GNNExplainer (our method). Table 4.1 and Table 4.2 present the fidelity and accuracy of explanations for BA-Shapes and Tree-Cycles datasets across different K, respectively.

Table 4.1: 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 4.2: 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

When examining the results for the BA-Shapes, as shown in Table 4.1, 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 4.2, 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 4.1). 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 4.3.3, facilitating a thorough analysis.

4.3.3 ACGAN-GNNExplainer on Real-world Datasets

Here we further experiment with our method with two popular real-world datasets including Mutagenicity [27] and NCI1 [67]. The experimental results for Mutagenicity and NCI1 are shown in Table 4.3 and Table 4.4, respectively.

From Table 4.3, 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 4.4, one can observe that ACGAN-GNNExplainer consistently outperforms other competitors in terms of fidelity and accuracy in different values of 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 highlight the effectiveness of our proposed method in producing faithful explanations.

Table 4.3: 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 4.4: 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**

4.3.4 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 $\mathscr{F}$ 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 $\mathscr{F}$—resulting in a correct prediction or causing a wrong prediction. Specifically, when the target GNN model $\mathscr{F}$ yields a correct prediction (e.g., the first-row visualization example in Figure 4.2), 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 $\mathscr{F}$ produces an incorrect prediction (e.g., the second-row visualization example in Figure 4.2), 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 $\mathscr{F}$. In particular, when the target GNN model $\mathscr{F}$ 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 4.2, 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 $\mathscr{F}$ 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.

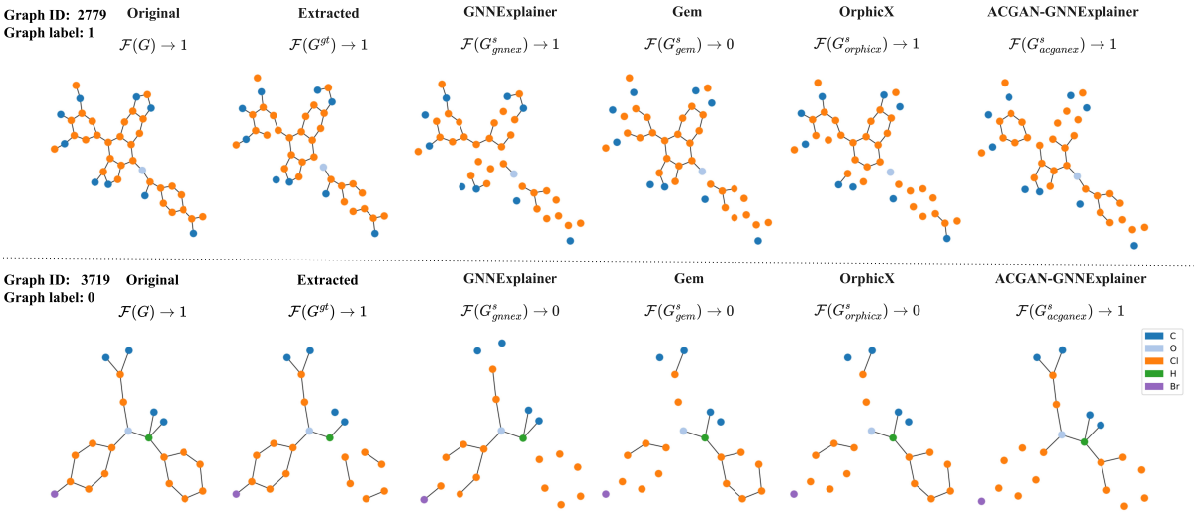

Figure 4.2: The Explanation Visualization on NCI1, When $R = 0.5$. $\mathscr{F}(\cdot) \to \{0,1\}$ means predictions made by the target GNN model $\mathscr{F}$. The 1^{st} column contains the initial graph. The 2^{nd} column showcases the real explanation that we obtained during the preprocessing stage. 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 $\mathscr{F}$. 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.

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.

4.4 Limitations and Discussions

Despite achieving competitive performance in terms of accuracy and fidelity, our ACGAN-GNNExplainer is still with its limitations, warranting further investigation and refinement.

Need for Preprocessing. The current preprocessing step utilized to distillate real explanations for training data imposes significant computational overheads and time constraints. Future research endeavours could concentrate on refining the model architecture to mitigate the reliance on ground-truth data, thereby streamlining the preprocessing stage and improving efficiency.

Requirement of a Large Number of Training Graphs. Another limitation arises from the substantial requirement for a large number of training graphs to effectively train our method. This demand for extensive training data may pose practical challenges in scenarios where acquiring such datasets is costly or labour-intensive. Addressing this limitation may involve exploring techniques for efficient data augmentation or semi-supervised learning to alleviate the demand for extensive training data.

Lack of Consideration for Fairness. Additionally, our current methodology overlooks the crucial aspect of fairness by not explicitly considering the balance between different subgroups when generating explanations. This oversight raises ethical concerns regarding the potential perpetuation or exacerbation of biases within the model's explanations. Addressing this limitation calls for integrating fairness-aware techniques into the explanation generation process, ensuring equitable treatment across diverse demographic groups and mitigating the risk of algorithmic bias.

By acknowledging and addressing these limitations, we can advance the capabilities and applicability of ACGAN-GNNExplainer, fostering more robust and equitable interpretability solutions for graph-based models in various domains.

4.5 Summary

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 to obtain good performance on real-world datasets.

To address this limitation, we, in this work, propose an ACGAN-based explainer, dubbed ACGAN-GNNExplainer, for GNNs. 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.

CHAPTER 5

DECODER-BASED COUNTERFACTUAL EXPLAINER FOR GRAPH NEURAL NETWORKS

In response to the limitations identified in *ACGAN-GNNExplainer*, this work introduces *fairCFE*, a method designed to produce fair CFEs for GNNs. CFEs for GNNs are essential in explaining these models by addressing the question: *"How can we minimally modify input graphs to make GNNs produce specific, predetermined predictions?"* While recent approaches have emerged to generate CFEs using various strategies, these methods typically require extensive training data, which may not always be feasible, and lack mechanisms to ensure unbiased explanations. Unlike prior approaches, fairCFE simultaneously optimizes both the input seed and the deep decoder's network parameters for a specific graph, eliminating the need for additional training data. To further ensure fairness, we incorporate a novel fairness loss component into the optimization process, guiding the generation procedure to produce fair CFEs. This work not only advances GNN explainability but also ensures fairness in explanation generation. Extensive experiments on diverse datasets demonstrate the superiority of fairCFE over state-of-the-art

baselines, affirming its effectiveness in generating fair CFEs for GNNs.

5.1 Introduction

The explanation of GNNs, which is essential to understanding the fundamental working mechanism of complex GNNs, guaranteeing the safety of their applications and promoting the reliability of GNNs, has attracted significant attention in recent years. These active research works could be categorized into two mainstreams–*factual explanations (FE)* and *counterfactual explanations (CFE)*. FE aims to answer the question: *why GNNs make that particular decision by finding the most important subgraphs/features*. Notable examples include GNNExplainer [76], OrphicX [38], and ACGAN-GNNExplainer [34]. CFE, on the contrary, attempts to answer the question: *how to modify the original graphs so that GNNs could make the desired predetermined prediction*. CFE typically generates a new graph conditioned by the desired predetermined prediction. Recent works in this direction include CF-GNNExplainer [40], CFF [59], and CLEAR [42]. In this work, we also focus on CFE.

CFE empowers stakeholders to explore hypothetical scenarios by identifying minimal modifications to input graphs that alter GNN predictions. This capability is invaluable for sensitivity analysis, risk assessment, and model refinement, as it allows stakeholders to assess the robustness of GNN predictions and proactively mitigate potential biases or errors. For instance, consider a recommendation system employing GNNs to suggest personalized items to users based on their historical interactions. A CFE approach might explore subtle adjustments to the user-item interaction graph, such as adding or removing edges representing past interactions, to influence the recommendations towards desired outcomes, such as increased user engagement or satisfaction. By discerning these minimal modifications, CFE empowers stakeholders to fine-tune GNNs for specific objectives while preserving the integrity of the underlying data structure.

Practical Issues Although current CFE explainers have demonstrated impressive performance in synthetic graph datasets, they present several limitations in the following aspects that hamper the practical deployment of CFE models.

Data. The existing CFE models generally require a substantial amount of graph dataset for training in order to produce satisfactory counterfactual explanations. However, it might be expensive or even impossible to collect such a large training data set due to concerns about privacy, legal constraints, and ethical implications. Worse, real-world graph datasets are never ideal and could be contaminated by different corruptions. Thus, a data-driven trained CFE model may be easily broken down during reference time when the test graphs stem from a distribution that differs from the training graphs.

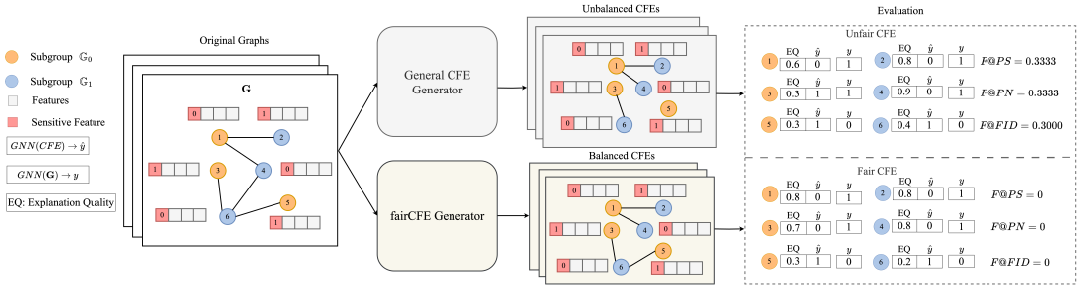

Figure 5.1: Motivation for fairCFE. A CFE refers to a graph that is minimally modified from its original form to achieve a desired prediction. Higher-quality CFEs lead to different predictions in the target GNN models. This figure illustrates the difference between unbalanced CFEs, which fail to maintain performance parity between subgroups (e.g., $\mathbb{G}_0$ and $\mathbb{G}_1$), and balanced CFEs, which ensure consistent performance across these subgroups. In this figure, $F@PS$, $F@PN$, and $F@FID$ represent the differences in probability of sufficiency, probability of necessity, and fidelity of explanations between different subgroups, respectively.

Fairness. The present CFE models are limited to generating a new graph that enables the GNNs to make the desired given prediction. They do not consider other important factors, such as *fairness*, significantly impacting real-world applications. Without taking *fairness* into account, a CFE model may produce counterfactual explanations that exhibit a bias towards a specific gender or ethnicity. Such explanations could poten-

tially mislead or even endanger practitioners in real-world applications such as credit evaluation and job marking.

As illustrated in Figure 5.1, the original graph consists of subgroups differentiated by the value of a sensitive feature: $\mathbb{G}_0$ when the sensitive feature is 0, and $\mathbb{G}_1$ when the sensitive feature is 1. The CFE generator should maintain a balanced performance across different subgroups, such as $\mathbb{G}_0$ and $\mathbb{G}_1$. However, the current CFE generator does not account for group fairness during the generation process, resulting in CFEs that fail to maintain subgroup balance. To address this issue, we propose fairCFE, a method designed to generate fair CFEs by actively regulating the distribution of nodes among different subgroups throughout the generation process.

Our Focus and Contributions In this work, we attempt to propose a practical CFE model for GNNs. It should not only produce *faithful* CFEs but also ensure *fairness* in its explanations. To achieve this goal, we employ deep decoders $\mathscr{D}\omega$ parameterized by ω for the adjacency matrix and $\mathscr{D}\theta$ parameterized by θ for the feature matrix as our CFE-generative model, learning a tailored optimal $\mathscr{D}\omega$ and $\mathscr{D}\theta$ for each given graph. By doing so, our model eliminates the need for massive training data and has naturally addressed the data distribution-shift issue. This *untrained* idea has been increasingly gaining popularity in the field of computer vision [23, 28, 30–32, 62, 69, 85]. Furthermore, in order to guarantee that our CFE-generative model generates explanations that are fair, we propose a new *fairness* loss and incorporate it into the decoder's loss function to achieve our final optimization objective. We defer the details of our method to Section 5.2.

Our main contributions to this work include the following:

- Novel Untrained CFE-Generative Model: We introduce fairCFE, an innovative untrained model capable of generating faithful counterfactual explanations (CFEs) without the need for massive training data.

- Fairness Loss for Unbiased CFEs: We propose a new fairness loss to guide the generation process in fairCFE, ensuring that the resulting CFEs are both fair and unbiased.

- Comprehensive Evaluation: We demonstrate the effectiveness of our fairCFE method through extensive experiments conducted on diverse datasets, comparing it against state-of-the-art baseline methods.

5.2 Method

5.2.1 Problem Formulation

Counterfactual explanations (CFEs) for GNNs answer the question: *how can we adjust the original graphs so that we can expect the GNNs to yield the desired given predictions?*

Consider a graph $G = (\mathbf{V}, \mathbf{A}, \mathbf{X})$, where $\mathbf{V} = \{v_1, v_2, \ldots, v_n\}$ means the set of graph nodes; $\mathbf{A}$ is the adjacency matrix with $A_{ij} = 1$ indicating an edge between nodes i and j, and $A_{ij} = 0$ otherwise; and $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ represents the set of node features. For each node, it has m features such that $X_i = \{f_1, f_2, \ldots, f_{m-1}, f^s\}$, where f^s is the sensitive feature within this set (e.g., f^s could correspond to a feature such as gender or ethnicity). We define the subgraph neighborhood of a node v as the set of the nodes and edges within a specified range, represented as a tuple: $G_v = (\mathbf{A}_v, \mathbf{X}_v)$, where $\mathbf{A}_v$ is the adjacency matrix of the subgraph, and $\mathbf{X}_v$ is the feature matrix of the nodes that are at most h hopes away from v. And we define a set $\mathcal{G} = \{G_1, G_2, \cdot, G_n\}$ conclude all neighborhood subgraphs.

Further consider a GNN $\mathcal{F}_\phi$ which is parameterized by ϕ. For node classification, the function $\mathcal{F}_\phi$ is employed to accurately classify each node. We assume that we have obtained predictions $Y = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{F}_\phi$ for nodes. $\mathcal{F}_\phi(G_v) \rightarrow y$.

We define the corresponding counterfactual predictions as $Y^* = \{y_1^*, y_2^*, \ldots, y_n^*\}$, where $y_i^* \neq y_i$ representing our desired predictions/labels. We further define G_v^{cf} as the CFE

of the original graph G_v generated by the explainer model $\mathscr{D}$, which should meet the requirements: $\mathscr{F}_\phi(G_v) \rightarrow y$ and $\mathscr{F}_\phi(G_v^{cf}) \rightarrow y^*$ and $y \neq y^*$. In this work, our goal is to propose an explanation model $\mathscr{D}$ that is able to generate *faithful* and *fair*/*unbiased* CFEs.

5.2.2 Preliminaries

Fair CFE is crucial for ensuring fairness, transparency, and accountability in GNNs by mitigating biases and providing equitable explanations. This work seeks to achieve fairness in CFE generation by tackling data distribution shifts and inherent group biases. Specifically, we apply the decoder as the backbone to mitigate data distribution shifts and incorporate a fairness loss to guide the generation process, ensuring that the resulting CFEs exhibit group fairness.

We introduce two novel definitions for evaluating CFE quality: Probability of Sufficiency (PS^{cf}) and Probability of Necessity (PN^{cf}), based on established principles of PS and PN [59]. Furthermore, we propose new fairness metrics derived from PS^{cf} and PN^{cf}, detailed in Section 5.3.3.

Definition 1. *Probability of Sufficiency for CFEs (PS^{cf}). It represents the proportion of generated explanations that suffice for an instance to attain the intended prediction, distinct from the prediction derived from the entire graph. The formulation for PS^{cf} is articulated in Equation* (5.1).

$$PS^{cf} = \frac{\sum_{G_i \in \mathbf{G}} ps_i^{cf}}{|\mathbf{G}|}, \text{ where } ps_i^{cf} = \begin{cases} 1, \text{ if } \mathscr{F}(G^{cf}) \neq \mathscr{F}(G) \\ 0, \text{ else} \end{cases} \tag{5.1}$$

Specifically, PS^{cf} measures the proportion of graphs for which the explanation (subgraph) alone induces a modification in the GNN prediction, signifying its sufficiency

for CFEs. The calculation of PS^{cf} is elucidated in Equation (5.2).

$$PS^{cf} = \frac{\mathbf{1}(\mathscr{F}(G^{cf}) = y^*)}{|\,\mathbf{G}\,|} = \frac{\mathbf{1}(\mathscr{F}(G^{cf}) \neq y)}{|\,\mathbf{G}\,|}, \tag{5.2}$$

where $|\mathbf{G}|$ represents the size of the set $\mathbf{G}$. The function $\mathbf{1}(\cdot)$ is an indicator function, producing 1 when the specified condition is true and 0 otherwise.

Definition 2. *Probability of Necessity for CFEs (PN^{cf}). It represents the proportion of CFEs that are necessary for the instance to achieve the desired prediction, distinct from the prediction derived using the entire graph. The PN^{cf} is formally defined as Equation* (5.3).

$$PN^{cf} = \frac{\sum_{G_i \in \mathbf{G}} pn_i^{cf}}{|\mathbf{G}|}, \text{ where } pn_i^{cf} = \begin{cases} 1, \text{ if } \mathscr{F}(G^{cf}) = \mathscr{F}(G) \\ 0, \text{else} \end{cases} \tag{5.3}$$

Intuitively, PN^{cf} measures the percentage of graphs where removing the CFE subgraph maintains the GNN prediction similar to using the entire graph, indicating its necessity. The computation of PN^{cf} is determined in Equation (5.4).

$$PN^{cf} = \frac{\mathbf{1}(\mathscr{F}(G^{pn}) \neq y^*)}{|\,\mathbf{G}\,|} = \frac{\mathbf{1}(\mathscr{F}(G^{pn}) = y)}{|\,\mathbf{G}\,|}, \tag{5.4}$$

where the $|\mathbf{G}|$ and $\mathbf{1}(\cdot)$ have the same meaning as defined in Equation (5.2).

5.2.3 Counterfactual Explanations Generation

Precursors of Decoder The seminal work [60] discovers that jointly optimizing both the network input seed and parameters of the decoder is sufficient to capture the non-linear correlations among the data, resulting in more efficient in reducing data dimensionality than training an encoder-decoder network. Later, this idea has been expanded to deep nonlinear matrix factorization [17], 3D shape representation [47], and robust manifold learning [29]. Especially, [8] shows that a deep decoder can function as a

generative model and exhibits numerous desirable properties of generative adversarial networks (GANs). More recently, Deepdecoder [23] and ConvDecoder [13] have further explored this idea for image restoration without using additional training datasets.

Decoder for Counterfactual Explanations Generation In this work, we also adopt deep decoders as our backbone to generate counterfactual explanations. To our knowledge, this is the first time that this idea—*deep decoder functions as a generative model*—has been introduced into the domain of GNN explanation. To be specific, we elaborately split a counterfactual explanation into two parts: an adjacency matrix $\mathbf{A}^{cf}$ indicating the connections between the nodes and a feature matrix $\mathbf{X}^{cf}$ containing the information of the nodes' features. We employ $\mathscr{D}_\omega$ parameterized by ω to generate the adjacency matrix $\mathbf{A}^{cf}$ and utilize $\mathscr{D}_\theta$ parameterized by θ to generate the feature matrix $\mathbf{X}^{cf}$. In contrast to a traditional encoder-decoder architecture, where the decoder receives a learned latent code generated by the encoder, our $\mathscr{D}_\omega$ and $\mathscr{D}_\theta$ are fed a random input seed $\mathbf{z}$ sampled from a Gaussian distribution $\mathscr{N}(0, 0.001)$. Furthermore, during the optimization process, we update the input seed $\mathbf{z}$ and the network parameters ω and θ so that an optimal $\mathbf{z}^*$, ω^*, and θ^* together determine the final desired CFE.

Empirically, if no constraints are present, the deep decoders could generate arbitrary explanations that could substantially deviate from the original input graphs. This undermines a fundamental principle of our initial goal: explanations should be *faithful* such that they should be meaningful explanations with minimal modifications to the original input graphs and should not be too different from the original input graphs. To prevent the generation of arbitrary and trivial explanations, we first introduce a similarity loss that measures the similarity between the generated explanation and the original input graph, as shown in Equation (5.5):

$$\begin{aligned} \mathscr{L}_{sim} &= \mathbf{CE}(A, A^{cf}) + \eta \cdot \mathbf{Dist}(\mathbf{X}, \mathbf{X}^{cf}) \\ \mathbf{A}^{cf} &= \mathscr{D}_\omega(\mathbf{z}, y^*), \mathbf{X}^{cf} = \mathscr{D}_\theta(\mathbf{z}, y^*), \end{aligned} \tag{5.5}$$

where $\mathbf{A}$ is the adjacency matrix of the given graph G, $\mathbf{X}$ is the feature matrix of the given graph G, $\mathbf{z}$ is the input seed for our deep decoders, and y^* is the predetermined counterfactual prediction label, the $\mathbf{CE}(\cdot)$ means to calculate the Cross-Entropy, the $\mathbf{Dist}(\cdot)$ represents the distance, in here, we adopt Mean Square Error (MSE); the first term measures the similarity between the original adjacency matrix and the generated counterfactual adjacency matrix and the second term measures the similarity between the original features and the generated counterfactual features; the hyperparameter η is used to weigh the importance of these two similarities.

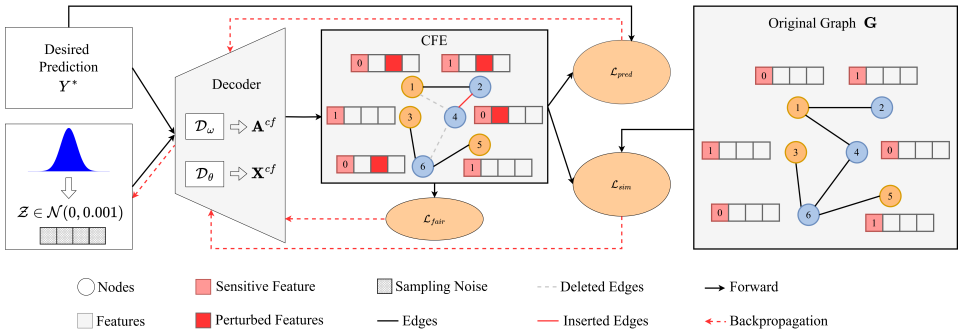

Figure 5.2: The Framework of fairCFE. It adopts a *Decoder* as its generative model. The input seed $\mathbf{z}$ is sampled from a Gaussian distribution $\mathcal{N}(0, 0.001)$. The generation of counterfactual explanations is conditioned by the desired given prediction Y^*. The input seed $\mathbf{z}$ and the *Decoder* are updated simultaneously by the combined optimization objective: $\mathscr{L}_{fairCFE} = \alpha\mathscr{L}_{sim} + \beta\mathscr{L}_{pred} + \gamma\mathscr{L}_{fair}$.

Furthermore, simply focusing on the similarity between the generated and original input graph is insufficient, as it may cause the decoder to become a "lazy learner", overfitting to the original input graph and producing trivial generations. To overcome this and ensure the generation of counterfactual explanations (CFEs) that achieve the desired prediction from the GNN models, we condition the decoders $\mathscr{D}$ on the predetermined target prediction $\mathbf{Y}^*$. This is achieved by incorporating a prediction loss, as outlined in Equation (5.6):

$$\mathscr{L}_{pred} = \mathbf{CE}(\mathscr{F}_\phi(\mathbf{A}^{cf}, \mathbf{X}^{cf}), y^*)\,, \tag{5.6}$$

where $\mathbf{CE}(\cdot)$ means to calculate the Cross-Entropy, $\mathscr{F}_\phi(G^{cf})$ is the prediction of a pre-trained/target GNN $\mathscr{F}_\phi$ on the explanation generated by the decoder $\mathscr{D}$ and y^* is the desired predetermined prediction. We then calculate the negative log-likelihood between these two predictions. Ideally, these two predictions should be as close as possible.

5.2.4 Fairness Safeguard

Adopting the deep decoders as our generative models and the learning strategy that we have mentioned in Section 5.2.3, our model is able to generate *faithful* counterfactual explanations. However, it ignores one important practical factor—*fairness*. For example, the generated counterfactual explanations might be biased to a particular gender or ethnicity. To ensure that the decoders produce *unbiased* counterfactual explanations, in this work, we introduce a novel loss to guide the entire generation process so that each node/feature in the generated explanations from different subgroups (according to their sensitive feature) has an equal probability (see Equation (5.7)):

$$\mathscr{L}_{fair} = \mathbf{Dist}(\,\mathbf{A}_0^{cf}, \mathbf{A}_1^{cf}) + \mathbf{Dist}(\,\mathbf{X}_0^{cf}, \mathbf{X}_1^{cf})\,, \tag{5.7}$$

where $\mathbf{A}_0^{cf}$ and $\mathbf{X}_0^{cf}$ denote the adjacency and feature matrices of a graph with the sensitive feature as 0 ($f^s = 0$), respectively, while $\mathbf{A}_1^{cf}$ and $\mathbf{X}_1^{cf}$ represent the adjacency and feature matrices of a graph with the sensitive feature set as 1 ($f^s = 1$), respectively. The term $\mathbf{Dist}(\cdot)$ denotes the distance; in this context, we utilize the MSE as our chosen distance metric.

5.2.5 fairCFE Optimization Objective

Now, we have our final model dubbed *fairCFE* that is capable of generating *faithful* and *unbiased* counterfactual explanation conditioned by the desired predetermined prediction for an input graph without additional training datasets. We depict its framework in Figure 5.2. In addition, we jointly optimize the input seed $\mathbf{z}$ and the parameters θ of the decoder $\mathscr{D}_\theta$ according to the combined optimization objective, as shown in Equation (5.8):

$$\mathscr{L}_{fairCFE} = \alpha \mathscr{L}_{sim} + \beta \mathscr{L}_{pred} + \gamma \mathscr{L}_{fair}\,, \tag{5.8}$$

where α, β, and γ are there hyper-parameters used to balance the trade-off: 1) to what degree of minimal modifications should we make to the original input graph (*faithful*); 2) how to ensure the generated explanation lead the target GNN to make a desired prediction (*counterfactual*); and 3) how to guarantee the generated explanation is unbiased (*fairness*).

5.3 Experiments

5.3.1 Experimental Datasets

To demonstrate the effectiveness of our method, fairCFE, we conduct experiments across a range of node classification datasets. Specifically, we select three real-world datasets: Math [12], Por [12], and German [6]. Each dataset incorporates a sensitive feature, such as gender. Detailed statistics for these datasets are presented in Table 5.1, followed by detailed descriptions of each dataset.

Student-Mathematics (Math)/Portuguese (Por) [12]: Math and Por datasets are the same students but differ in their class labels, i.e., performance in a Math or Portuguese course. The sensitive attribute is gender, and the task is to predict whether a student,Äôs final year grade in the Math/Portuguese course is high.

Table 5.1: Dataset Statistics. In the table, $\#F$ denotes the count of node features; S represents the sensitive feature.

Datasets	#Graphs	#Nodes	#Edges	#F	#Classes	S	Label
Math	1	395	1,540	32	2	Gender	Grade
Por	1	649	3,125	32	2	Gender	Grade
German	1	1,000	24,970	27	2	Gender	Credit Risk

German Credit (German) [6]: This dataset has 1,000 nodes representing customers in a German bank, connected based on the similarity of their credit applications. The sensitive attribute is gender, and the task is classifying clients into good vs. bad credit risks.

5.3.2 Experimental Settings

We systematically evaluate the efficacy of our fairCFE by conducting comparative evaluations against four state-of-the-art baselines: Random [40], CFF [59], RCExplainer [7], and CLEAR [42]. To ensure consistency, we adopt the experimental setup of CF-GNNExplainer [40] for Random [40], and adhere to the parameters outlined in the original paper for CFF [59]. We extend the original implementations of RCExplainer [7] and CLEAR [42], initially designed for graph classification, to suit the node classification datasets used in our evaluation.

For our fairCFE, we assign the desired label y^* for each node as its flipped label (e.g., if $y = 0$, then $y^* = 1$) across all experiments. We use SGD as our optimizer. For each setting, we run five different initializations and report their mean and standard deviation.

The dataset split details and the accuracy of GNNs are provided in Table 5.2. Additionally, Table 5.3 presents the parameters used for fairCFE on each dataset.

Table 5.2: Datasets Split and Accuracy. We split each dataset into training, validation, and test data by the ratio of 0.8, 0.2, and 0.2, respectively. We keep the training, validation, and test data the same in modelling GNNs and explainers.

Datasets	# Training	# Val	# Test	GIN (Acc)	GCN (Acc)	GAT (Acc)	SAGE (Acc)
German	799	100	101	0.7400	0.7200	0.7400	0.7300
Math	301	39	40	0.8750	0.8750	0.8500	0.8250
Por	404	65	65	0.9077	0.8923	0.8923	0.8615

Table 5.3: Parameters for fairCFE. In this table, lr-exp means the learning rate for training fairCFE, and lr-noise means the learning rate for updating the input noise.

GNNs	Datasets	α	β	γ	lr-exp	lr-noise
GCN	Math	10	5	3	0.001	0.001
	Por	10	3	3	0.001	0.001
	German	10	3	5	0.001	0.001
GIN	Math	10	5	3	0.001	0.001
	Por	10	3	10	0.0005	0.001
	German	10	3	5	0.0001	0.0001
GAT	Math	10	10	1	0.005	0.0001
	Por	10	15	10	0.001	0.0005
	German	10	5	3	0.001	0.01
SAGE	Math	10	5	3	0.001	0.001
	Por	10	5	3	0.001	0.001
	German	10	3	5	0.01	0.001

5.3.3 Evaluation Metrics

We evaluate the generated CFEs from the perspective of reliability and fairness. To evaluate reliability, we use Counterfactual Accuracy (ACC^{cf}) and Fidelity (FID). To evaluate fairness, we adopt group fairness preservation (GFP), CFE fairness under PS ($F@PS$), CFE fairness under PN ($F@PN$), and CFE fairness under fidelity ($F@FID$).

Ideally, we should expect high values in reliability-evaluation metrics—$ACC^{cf}(\uparrow)$ and $FID(\uparrow)$ and low values in the fairness-evaluation metrics—-$GFP(\downarrow)$, $F@PS(\downarrow)$, and $F@PN(\downarrow)$, $F@FID(\downarrow)$.

Counterfactual Accuracy. It is defined as the proportion of generated explanations that change the model,Äôs prediction. The definition of CF accuracy is shown in the Equation (5.9).

$$ACC^{cf} = 1 - \frac{1}{|\mathbf{G}|} \sum_{G \in \mathbf{G}} (\mathbf{1}(\mathscr{F}(G) = \mathscr{F}(G^{cf}))), \tag{5.9}$$

where $|\mathbf{G}|$ denotes the size of $\mathbf{G}$. $\mathbf{1}(\cdot)$ is the indicator function to check whether $\mathscr{F}(G)$

equals to $\mathscr{F}(G^{cf})$. Since we aim to generate counterfactual explanations, a higher counterfactual accuracy (↑) is better.

Fidelity. It measures the change in output probability over the original class. The definition of $Fidelity$ is shown in the Equation (5.10).

$$FID = \frac{1}{|\mathbf{G}|} \sum_{G \in \mathbf{G}} [\mathscr{F}(G \mid y) - \mathscr{F}(G^{cf} \mid y)], \tag{5.10}$$

where $\mathscr{F}(G \mid y)$ denotes the output probability of the GNN model $\mathscr{F}$ for graph G over class y. A higher fidelity (↑) score indicates better counterfactual explanations.

CFE Fairness@Group Fairness Preservation [3]. It is to preserve the group fairness property of the underlying model (e.g. GNN models) without exhibiting significant bias against any sensitive group.

For a given sensitive attribute (e.g., gender or race), we partition the nodes into different groups. Group Fairness Preservation requires that the explanation method treats instances within the same sensitive group similarly, avoiding any systematic unfair treatment towards certain instances, thereby preserving the inherent fairness properties of the underlying model. Formally, we calculate the degree of group fairness preservation by the following Equation (5.11):

$$GFP = \|GF(\mathscr{F}(\mathbf{G})) - GF(\mathscr{F}((\mathbf{G}^{cf})))\|, \tag{5.11}$$

where $GF(\cdot)$ is to computer the group fairness of a model. There are two traditional metrics to evaluate group fairness, including statistical parity (SP) [16] and equality of opportunity (EO) [22]. In our work, we choose SP to evaluate group fairness. The metric GFP measures the fairness preservation ability of the explainers, and a smaller value relates to a fairer model.

Additionally, we propose novel metrics, $F@PS$ and $F@PN$, to evaluate the group fairness of CFE based on Definition 1 and Definition 2. These metrics assess the quality gap of CFE among subgroups, using sufficiency and necessity ratios as quality indicators.

CFE Fairness@PS. Inspired by the definition of SP, the value of PS should remain consistent across different subgroups. Therefore, we define CFE fairness in terms of PS, as illustrated in Equation (5.12).

$$F@PS = \|(PS^{cf} \mid f^s = 1) - (PS^{cf} \mid f^s = 0)\|, \tag{5.12}$$

The fairness indicator for explanation quality is determined by the discrepancy in PS^{cf} values between subgroups, with a smaller difference indicating higher fairness.

Table 5.4: Results for GCN. The presented outcomes encompass averages from five runs alongside their corresponding standard deviations. Notably, the best-performing results have been emphasised in bold.

Dataset	Explainers	$ACC^{cf}(\uparrow)$	$FID(\uparrow)$	$F@PS(\downarrow)$	$F@PN(\downarrow)$	$F@FID(\downarrow)$	$GFP(SP)(\downarrow)$
Math	Random	0.3400 ± 0.0285	0.4113 ± 0.0405	0.1283 ± 0.0767	0.0546 ± 0.0446	0.0988 ± 0.1094	0.1624 ± 0.0871
	CFF	0.5350 ± 0.0487	0.4977 ± 0.1441	0.1609 ± 0.0936	0.1589 ± 0.0903	0.0122 ± 0.0047	0.1188 ± 0.0965
	RCExplainer	0.0526 ± 0.0322	0.3276 ± 0.1559	0.0526 ± 0.0322	0.5526 ± 0.0000	0.1623 ± 0.0422	0.0305 ± 0.0206
	CLEAR	0.6300 ± 0.0512	0.5684 ± 0.0400	0.2035 ± 0.0860	0.0441 ± 0.0288	0.1359 ± 0.0621	0.1023 ± 0.0967
	fairCFE	**0.9500 ± 0.0012**	**0.7230 ± 0.0019**	**0.0251 ± 0.0021**	**0.0052 ± 0.0015**	**0.0052 ± 0.0015**	**0.0301 ± 0.0043**
Por	Random	0.1754 ± 0.0371	0.2266 ± 0.0217	0.0884 ± 0.0721	0.0597 ± 0.0000	0.1484 ± 0.0388	0.0297 ± 0.0277
	CFF	0.5077 ± 0.0713	0.4101 ± 0.2421	0.0816 ± 0.0822	0.1254 ± 0.0973	0.1465 ± 0.0813	0.1441 ± 0.1137
	RCExplainer	0.1556 ± 0.0465	0.5855 ± 0.0955	**0.0074 ± 0.0101**	0.3333 ± 0.0000	0.6901 ± 0.1783	0.1024 ± 0.0469
	CLEAR	0.6115 ± 0.2529	0.6036 ± 0.1939	0.0481 ± 0.0509	**0.0275 ± 0.0000**	**0.0413 ± 0.0160**	0.0182 ± 0.0217
	fairCFE	**0.9938 ± 0.0084**	**0.9239 ± 0.0076**	0.0121 ± 0.0210	0.0597 ± 0.0000	0.0653 ± 0.0148	**0.0121 ± 0.0166**
German	Random	0.3060 ± 0.0404	0.3850 ± 0.0251	0.2026 ± 0.0933	**0.0478 ± 0.0974**	0.0955 ± 0.0451	0.1276 ± 0.1043
	CFF	0.5520 ± 0.0526	0.5019 ± 0.1275	0.0669 ± 0.0517	0.0982 ± 0.0611	0.0683 ± 0.0641	0.0912 ± 0.0962
	RCExplainer	0.5200 ± 0.0204	0.6201 ± 0.0130	0.1360 ± 0.0062	0.1290 ± 0.0105	0.1651 ± 0.0135	0.1460 ± 0.0580
	CLEAR	0.7880 ± 0.2504	0.6680 ± 0.2504	0.1360 ± 0.0000	0.1360 ± 0.0000	0.1360 ± 0.0000	0.1360 ± 0.0000
	fairCFE	**0.9700 ± 0.0000**	**0.8446 ± 0.0000**	**0.0478 ± 0.0000**	**0.0478 ± 0.0000**	**0.0286 ± 0.0000**	**0.0478 ± 0.0000**

CFE Fairness@PN. Similarly, the difference in PN^{cf} values between $f^s = 1$ and $f^s = 0$ determines the fairness of CFEs, with a smaller difference indicating increased fairness. We define CFE fairness in terms of PN, as illustrated in Equation (5.13).

$$F@PN = \|(PN^{cf} \mid f^s = 1) - (PN^{cf} \mid f^s = 0)\|, \tag{5.13}$$

where lower values (↓) signify enhanced fairness.

CFE Fairness@Fidelity [82]. It measures the unfairness of the average explanation quality by computing the difference of the subgroup,Äôs average explanation quality as shown in Equation (5.14):

$$F@FID = \| \frac{1}{|\mathbf{G}_0|} \sum_{\mathbf{G}_i \in \mathbf{G}_0} Fid(\mathbf{G}_i^{cf}) - \frac{1}{|\mathbf{G}_1|} \sum_{\mathbf{G}_i \in \mathbf{G}_1} Fid(\mathbf{G}_i^{cf}) \|, \tag{5.14}$$

here, $Fid(\cdot)$ means the fidelity score of the CFE, which we compute by the Equation (5.10). The $\mathbf{G}_0$ means the subgroup when the sensitive feature of nodes is 0 ($f^s = 0$), and the $\mathbf{G}_1$ means the subgroup when the sensitive feature of nodes is 1 ($f^s = 1$). Similarly, a smaller value indicates a higher level of fairness.

5.3.4 Performance Analysis

We conduct the experiments on various datasets (Math, Por, and German). We adopt the pretrained GCN as our target GNN model. We then explore five different GNN explainers—Random, CFF, RCExplainer, CLEAR, and fairCFE (ours)—to generate counterfactual explanations. We then evaluate the generated results and report the results in Table 5.4.

Based on the experimental results presented in Table 5.4, we observe that the proposed method, **fairCFE**, consistently outperforms other explainers across multiple datasets. The results are evaluated using several metrics: ACC^{cf}, FID, $F@PS$, $F@PN$,

$F@FID$, and $GFP(SP)$, where higher values for ACC^{cf} and FID are preferred, while lower values for the remaining metrics indicate better performance.

Across all datasets, the fairCFE method consistently achieves the highest accuracy, indicating that it provides the most reliable and accurate explanations. Similarly, fairCFE shows the highest FID scores, with values such as 0.9239 for the Por dataset, suggesting that the generated explanations are of the highest quality in terms of diversity and realism.

In contrast, the other methods, such as Random, CFF, RCExplainer, and CLEAR, demonstrate inferior performance across these metrics, with notable deviations in accuracy and FID. For instance, the Random method exhibits significantly lower accuracy, particularly on the Por dataset, with a score of 0.1754, indicating a high degree of inconsistency and unreliability in its explanations.

When examining the metrics intended to be minimized ($F@PS$, $F@PN$, $F@FID$, $GFP(SP)$), fairCFE consistently produces the lowest values, which highlights the fairness of generated CFEs. This quality is particularly evident in the German dataset, where fairCFE scores 0.0478 for $F@PS$ and $GFP(SP)$, substantially outperforming other methods.

The experimental results clearly demonstrate that fairCFE outperforms other explanation methods across all evaluated metrics. The high accuracy and FID scores, coupled with the minimal values in $F@PS$, $F@PN$, $F@FID$, and $GFP(SP)$, indicate that fairCFE is not only effective in producing accurate and high-quality explanations but also robust in maintaining fairness. These findings suggest that fairCFE is the most reliable and faithful method for generating explanations in GCN models across diverse datasets.

5.3.5 Ablation Studies

We further conduct ablation studies to explore how our method works for different target GNNs, as discussed in Section 5.3.5.1. The findings emphasise the versatility and effectiveness of *fairCFE* in delivering accurate explanations across various GNN architectures.

Additionally, in Section 5.3.5.2, we investigate how the proposed fairness loss, $\mathscr{L}_{fair}$, affects the performance of *fairCFE*. The results reveal that the inclusion of fairness loss significantly enhances the overall performance, further improving the quality and fairness of the generated explanations.

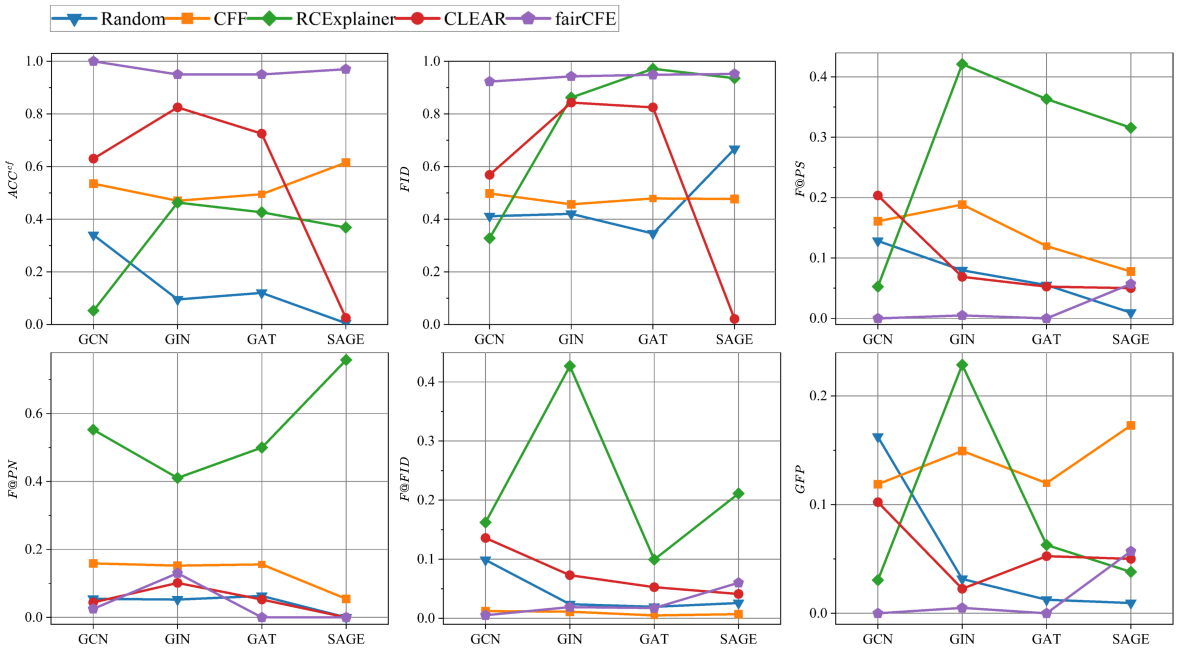

Figure 5.3: Comparative Performance of Various Explainability Methods Across Four GNN Architectures (GCN, GIN, GAT, and SAGE) on the Math Dataset.

5.3.5.1 Agnostic to Target GNNs

Our analysis involves a comparative assessment of various explainability methods across different target GNNs, specifically GCN, GIN, GAT, and SAGE, using two reliability

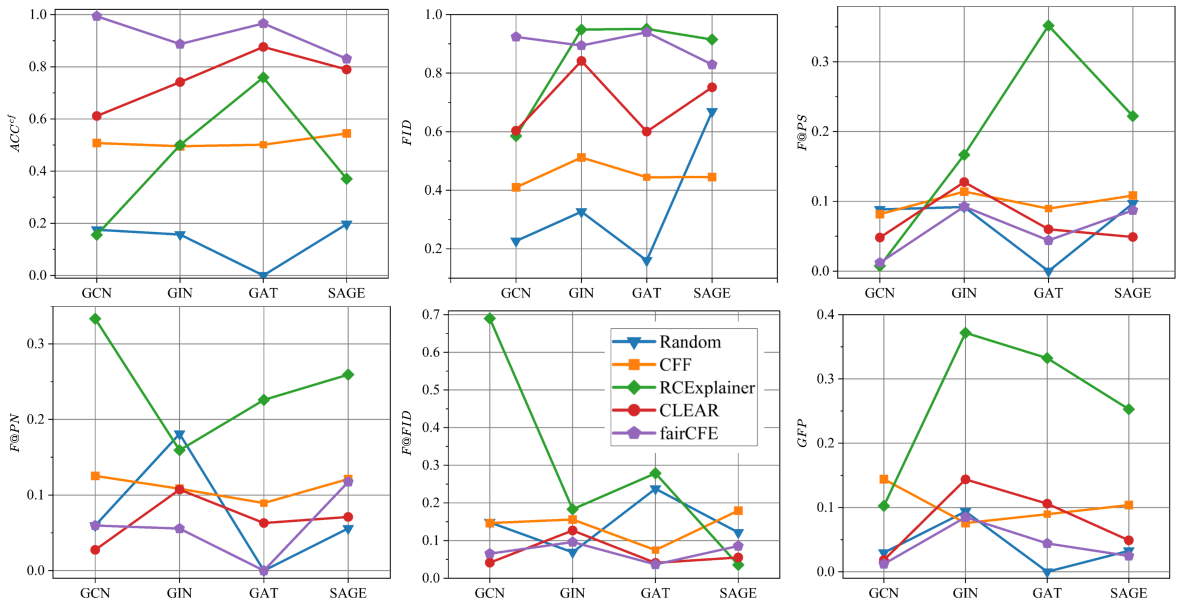

Figure 5.4: Comparative Performance of Various Explainability Methods Across Four GNN Architectures (GCN, GIN, GAT, and SAGE) on the Por Dataset.

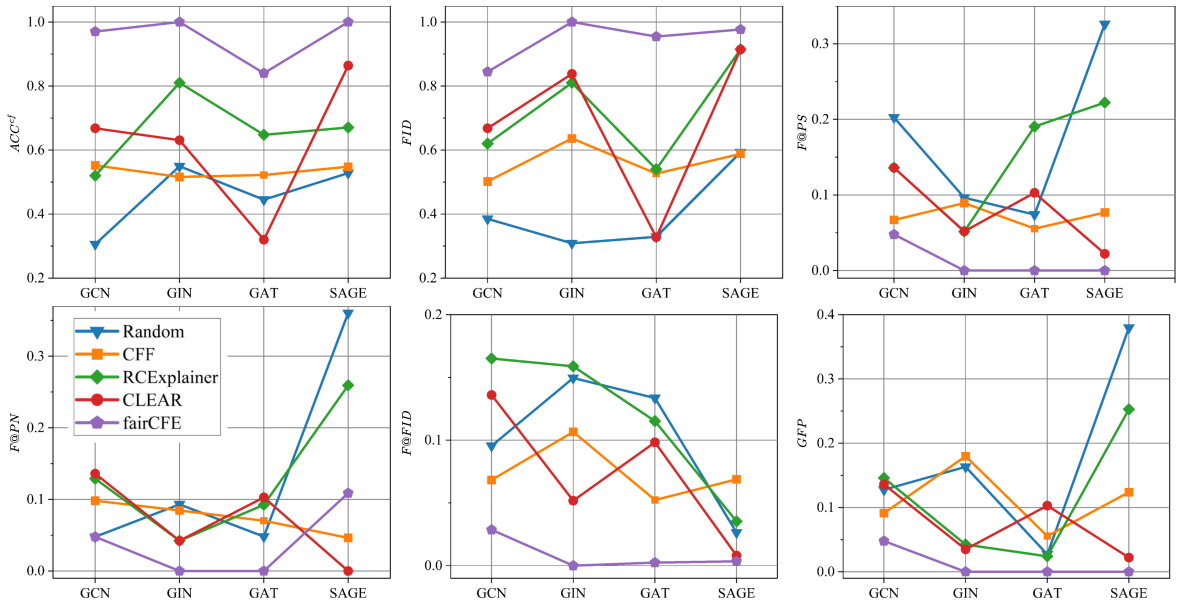

Figure 5.5: Comparative Performance of Various Explainability Methods Across Four GNN Architectures (GCN, GIN, GAT, and SAGE) on the German Dataset.

metrics (ACC^{cf} and FID) and four fairness-related metrics ($F@PS$, $F@PN$, $F@FID$, and GFP). This investigation aims to assess the agnosticism of the explainers, with an expectation of higher values on reliability metrics and lower values on fairness-related metrics across diverse GNNs. The results for the Math, Por, and German are presented in Figure 5.3, Figure 5.4, and Figure 5.5, respectively.

Figure 5.3 demonstrates that fairCFE consistently outperforms other explainability methods across all GNN architectures in both reliability and fairness-related metrics. fairCFE achieves the highest scores in accuracy (ACC^{cf}) and FID, reflecting the high reliability and quality of the generated counterfactuals. Additionally, it records the lowest values across all fairness-related metrics ($F@PS$, $F@PN$, $F@FID$, GFP), showcasing its ability to minimize bias between different subgroups while maintaining high fidelity. These results underscore fairCFE's effectiveness and versatility, establishing it as the most reliable and fair method across various GNN models.

As depicted in Figure 5.4, fairCFE consistently outperforms other explainability methods across all GNN architectures on the Por. It achieves the highest accuracy (ACC^{cf}) and FID scores, indicating its superior reliability and the high quality of its generated counterfactuals. Additionally, fairCFE records the lowest values across all fairness metrics ($F@PS$, $F@PN$, $F@FID$, and GFP), highlighting its effectiveness in minimizing bias and ensuring fidelity to the original data. These results underscore fairCFE's robustness and versatility, establishing it as the most reliable and fair method across diverse GNN models.

Figure 5.5 illustrates the performance of various explainability methods across four GNN architectures on the German dataset. fairCFE consistently demonstrates superior performance, achieving the highest accuracy (ACC^{cf}) and FID scores, indicating reliable and high-quality counterfactuals across all GNNs. It also excels in fairness metrics, consistently showing the lowest values in $F@PS$, $F@PN$, $F@FID$, and GFP, reflecting

its ability to minimize errors and biases effectively. fairCFE consistently outperforms other methods, achieving higher reliability and fairness with minimal variation across different GNN models.

These figures clearly demonstrate that fairCFE is the most effective method for generating fair, accurate, and faithful CFEs across a range of GNN models. The consistent performance of fairCFE across all evaluated metrics and various datasets underscores its robustness and reliability in ensuring fairness in GNN applications. In contrast, the other methods, particularly Random and CFF, show significant variability and generally underperform, especially in maintaining fairness across subgroups.

5.3.5.2 The Impact of $\mathscr{L}_{fair}$

Another ablation study was undertaken to evaluate the relevance and significance of the introduced fairness loss incorporated into our method. We anticipated observing enhanced performance, reflected in increased values in reliability metrics and reduced values in fairness metrics. Results for various datasets (Math, Por, and German) based on different GNN architectures (GCN, GIN, GAT, and SAGE) are presented in Figure 5.6, Figure 5.7, and Figure 5.8 respectively.

Figure 5.6 illustrates the impact of incorporating fairness loss into our explainability method across four GNN architectures. The results demonstrate that the inclusion of fairness loss significantly enhances the fairness of the generated explanations, as evidenced by the substantial reduction in $F@PS$, $F@PN$, $F@FID$, and GFP values across all GNN architectures. Notably, these improvements in fairness metrics are achieved without compromising the reliability of our method, as the ACC^{cf} remains consistently high and FID scores improve with the inclusion of fairness loss. These findings highlight the effectiveness of the fairness loss in producing high-quality, reliable, and fair counterfactual explanations, confirming its importance in the overall performance of the method.

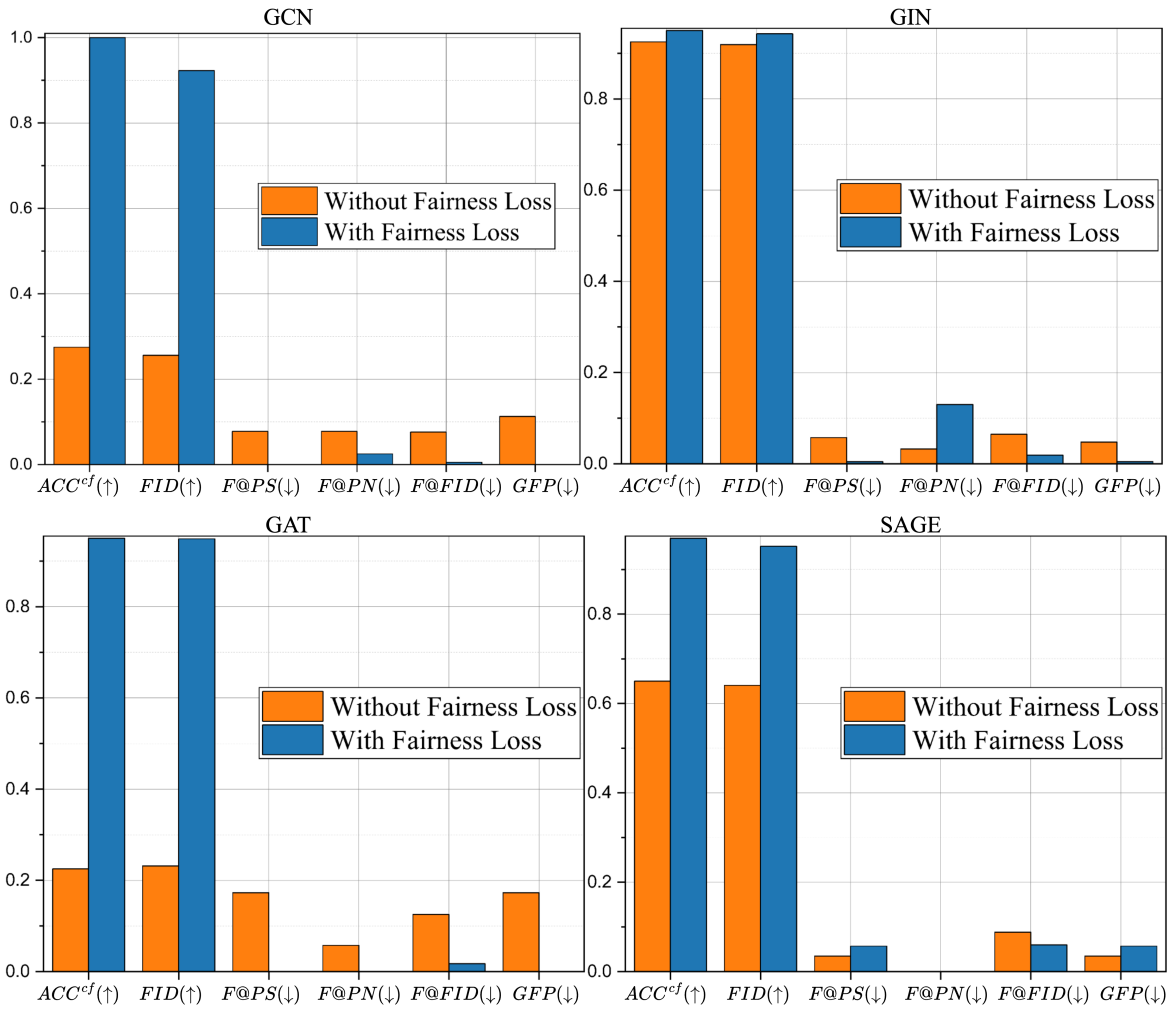

Figure 5.6: Fairness Loss Evaluation on Various GNNs Using the Math. We compare the performance of fairCFE with and without the proposed fairness loss. The results show that our proposed fairness loss significantly improves the performance of fairCFE.

Figure 5.7 shows the impact of incorporating fairness loss into the explainability method across four GNN architectures on the Por. The results clearly indicate that the inclusion of fairness loss leads to significant improvements in both reliability and fairness metrics. Specifically, the FID scores are markedly higher with fairness loss across all GNNs, reflecting enhanced quality and diversity in the generated counterfactuals. Furthermore, all fairness-related metrics ($F@PS$, $F@PN$, $F@FID$, GFP) exhibit lower values when fairness loss is applied, demonstrating a substantial reduction in biases and

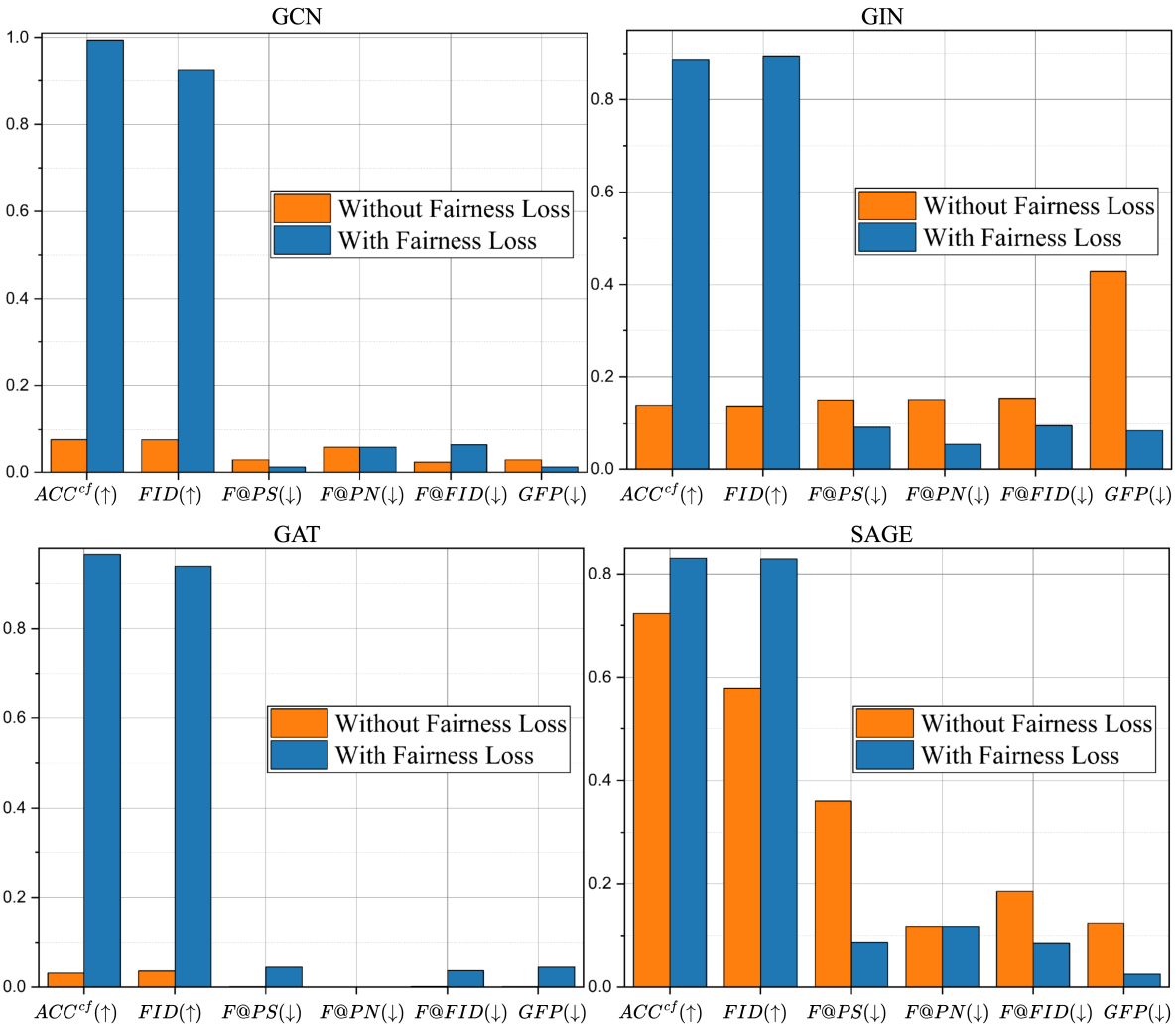

Figure 5.7: Fairness Loss Evaluation on Various GNNs Using the Por. We compare the performance of fairCFE with and without the proposed fairness loss. The results show that our proposed fairness loss significantly improves the performance of fairCFE.

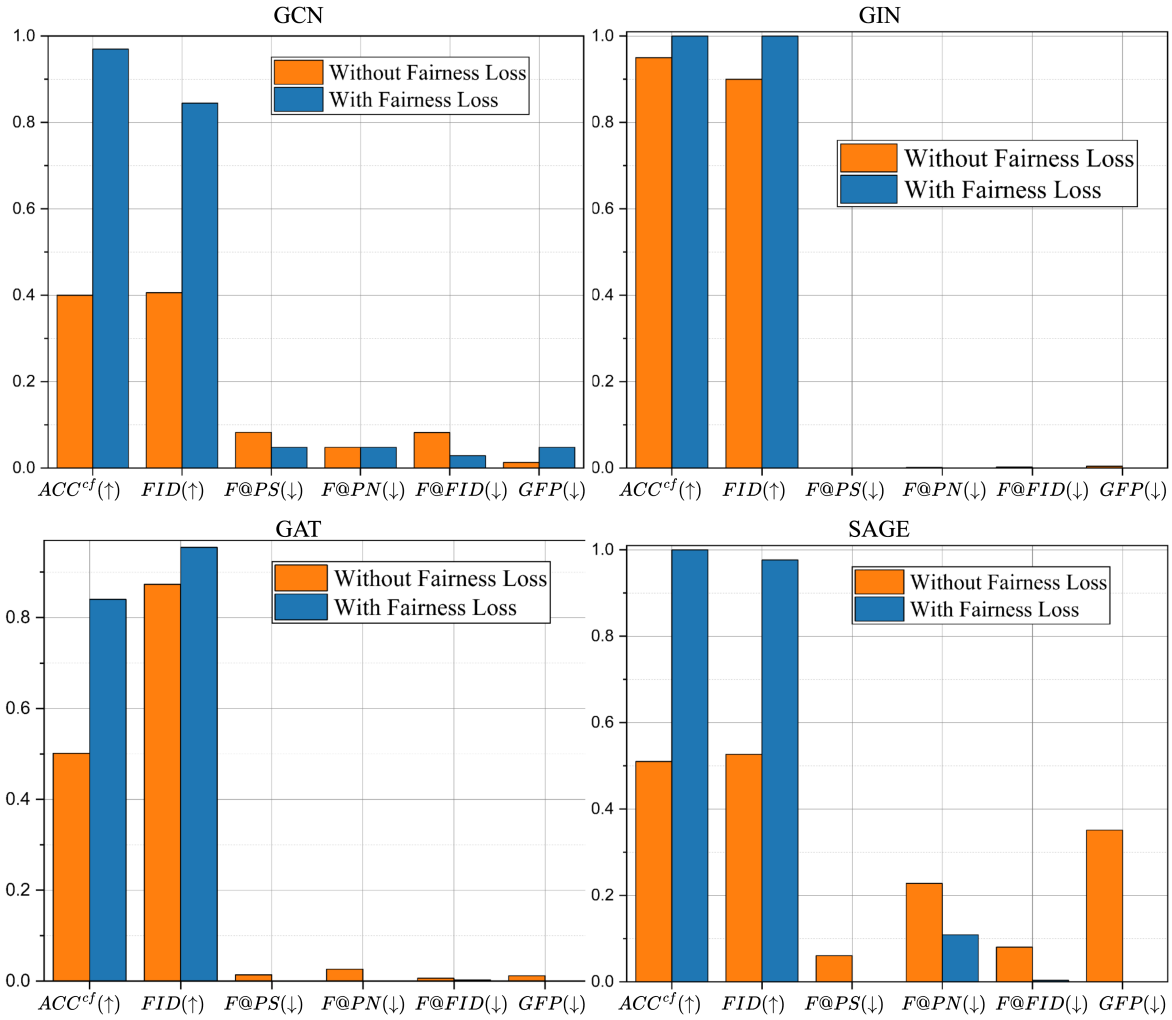

Figure 5.8: Fairness Loss Evaluation on Various GNNs Using the German. We compare the performance of fairCFE with and without the proposed fairness loss. The results show that our proposed fairness loss significantly improves the performance of fairCFE.

errors. Importantly, these improvements are achieved without compromising accuracy (ACC^{cf}), which remains consistently high across all architectures. Overall, the figure underscores the effectiveness of fairness loss in improving the fairness and reliability of counterfactual explanations across diverse GNN models.

Figure 5.8 presents the effect of incorporating fairness loss into our explainability method across four GNN architectures on the German dataset. The results show a consistent improvement in both reliability and fairness metrics when fairness loss is applied. Specifically, the inclusion of fairness loss leads to significantly higher FID scores across all GNN architectures, indicating enhanced quality and diversity in the generated counterfactuals. Additionally, fairness metrics ($F@PS$, $F@PN$, $F@FID$, and GFP) show marked reductions, demonstrating that the method with fairness loss effectively mitigates biases while maintaining high fidelity to the original data. The ACC^{cf} scores remain high in both scenarios, suggesting that the integration of fairness loss does not compromise the accuracy of the generated counterfactuals. These findings underscore the importance of fairness loss in improving the overall performance of the explainability method, ensuring both fairness and reliability across different GNN models.

These results clearly demonstrate that incorporating a fairness loss component into GNN models significantly enhances both the accuracy and fairness of CFEs. Fairness loss not only improves the alignment of explanations with desired outcomes but also ensures that these explanations are equitable across different subgroups. The consistent improvements across all evaluated metrics suggest that fairness loss is an effective way of mitigating biases and promoting fairness. This is particularly important in sensitive domains where fairness is a critical consideration.

5.4 Limitations and Discussions

Our method, fairCFE, offers several notable advantages. First, it does not rely on training data, which makes it robust to data distribution shifts. This characteristic is particularly valuable in real-world applications where distribution shifts are common. Second, the method ensures group fairness in the generated CFEs, addressing critical equity concerns in machine learning applications.

Despite these strengths, there are limitations that warrant further investigation. One significant limitation is that the generated explanations are not constrained to be subsets of the original graph. While this design choice allows for greater flexibility in generating explanations, it may reduce the explainability and fidelity of the explanations. Additional justification for this approach and potential ways to refine it are needed.

Another limitation is the reliance on group fairness metrics, which are inherently designed for evaluating fairness across multiple samples. This reliance makes the method less applicable in scenarios where only a single sample is available, posing challenges for fairness evaluation in such cases.

To address these limitations, future work will focus on two key directions. First, we will explore ways to justify and improve the explanation subset constraint, potentially by incorporating subset-based constraints without sacrificing the model's flexibility. Second, we will investigate methods to evaluate fairness in scenarios with limited or single samples, possibly by adapting existing metrics or developing new ones tailored to such contexts. These efforts aim to enhance the explainability, applicability, and fairness of the method in a broader range of scenarios.

5.5 Summary

In this work, we propose a novel counterfactual explainer for GNNs, dubbed fairCFE, which utilizes deep decoders as its backbone to generate counterfactual explanations. The deep decoders are conditioned by the desired predetermined prediction and fed with a random input seed. During the optimization process, both the input seed and the network parameters are updated jointly. To ensure the generation of unbiased explanations, a fair loss is introduced to guide the generation process.

We conduct extensive experiments on various datasets, rigorously testing the performance of our proposed method. Compared with state-of-the-art baselines, our fairCFE demonstrates superior in generating reliable, faithful, and unbiased counterfactual explanations for GNNs. This highlights the versatility of fairCFE across different contexts and its potential to be widely applicable in real-world scenarios. For future work, it would be valuable to explore additional dimensions of fairness, such as causal fairness, to further enhance the fairness of explanations and address more complex biases that may arise in real-world data. Additionally, investigating the scalability of fairCFE to more complex graphs could provide further insights into its applicability in broader contexts.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

6.1 Conclusions

In conclusion, this thesis delves into the critical realm of explaining GNNs, an area of increasing importance due to the widespread applications of GNNs in real-world scenarios. Through the exploration of FE and CFE, this research contributes to the understanding of GNN mechanisms, ensuring their safe and reliable deployment.

The investigation commenced with the development of GAN-GNNExplainer, a novel GNN explanation method utilizing GAN to produce explanations for original input graphs. Despite its initial promise, it was noted that this approach encounters performance limitations and fidelity constraints when applied to real-world datasets.

In response to these limitations, the study progressed to address the research question of generating fidelity explanations for GNNs. Introducing the ACGAN into the GNN explanation domain led to the creation of the ACGAN-GNNExplainer. This method significantly enhanced the fidelity of explanations on real-world datasets, outperforming existing GNN explainers through comprehensive experimental evaluations.

Moreover, the exploration extended to CFE for GNNs, focusing on the generation of CFE facilitating desired predictions through minimal modifications to input graphs. The development of fairCFE addressed challenges posed by the need for extensive training data and the risk of bias in generated explanations. Leveraging a deep decoder and a novel fairness loss, fairCFE ensures unbiased counterfactual explanations without additional training datasets, demonstrating superior performance compared to existing models across diverse datasets.

In this thesis, we made significant contributions towards addressing key questions in the realm of GNNs and their explainability.

- Firstly, we introduced GAN-GNNExplainer, a novel approach that leverages GANs to provide consistent explanations for GNN predictions. Our method stands out by offering global explanations, versatility across diverse datasets and tasks, and the ability to generate explanations without the need for retraining. Through comprehensive empirical evaluations, we demonstrated the superior performance of GAN-GNNExplainer compared to existing methods.

- Secondly, we proposed ACGAN-GNNExplainer, an innovative GNN explanation model that utilizes an ACGAN to generate explanations. By iteratively refining the generation process through the interplay between the generator and discriminator, our method achieves enhanced explanation accuracy and scalability to unseen graphs. Empirical validations across various datasets and tasks reaffirmed the effectiveness and robustness of ACGAN-GNNExplainer, establishing it as a leading solution in the domain of GNN explainability.

- Lastly, we introduced fairCFE, a groundbreaking counterfactual explanation model for GNNs designed to produce faithful explanations while ensuring fairness. Notably, fairCFE eliminates the reliance on massive training data and addresses data

distribution-shift issues. By incorporating a novel fairness loss into the generation process, our method generates unbiased and fair counterfactual explanations. Through extensive experiments spanning node and graph classifications across diverse datasets, we showcased the efficacy and fairness of fairCFE, underscoring its significance in advancing the explainability of GNN models.

In essence, this thesis advances the frontier of GNN explanation methodologies, providing insights and techniques crucial for understanding, deploying, and ensuring the reliability of GNNs in real-world applications. Future research may further refine and expand upon these methodologies to address emerging challenges and complexities in the evolving landscape of graph-based machine learning.

6.2 Future Research Directions

Our exploration of GNN explainability highlights several intriguing research directions that warrant further investigation:

1. **Exploration of More Effective Explanation Methods with Subgraphs**: Expanding on the notion of using subgraphs to explain GNNs, future work could focus on refining and implementing more effective methods for leveraging subgraph-based explanations. By delving deeper into the structure and dynamics of subgraphs within GNNs, researchers can potentially unlock richer insights into model behaviours and predictions. This approach holds promise for enhancing the explainability of GNNs by providing more granular and intuitive explanations rooted in the underlying subgraph structures.

2. **Development of a Fair FE Generative Explainer for GNNs**: Given the paramount importance of fairness in GNNs and their explanations, a compelling future direction involves the creation of a fair GNN explainer. This future endeavour

would seek to devise an explainability framework specifically tailored to preserve the fairness of GNNs while simultaneously ensuring high levels of counterfactual fairness. By integrating fairness considerations into the explainability process, this proposed Fair GNN Explainer could contribute significantly to the creation of more transparent and equitable GNN models.

3. **Evaluation and Validation of Explainability Techniques**: In the pursuit of advancing GNN explainability, future research could prioritize the rigorous evaluation and validation of various explainability techniques. This entails conducting comprehensive empirical studies to assess the efficacy, robustness, and fairness implications of different explanation methods, including fidelity explanations and fair counterfactual explanations. Through systematic evaluation, researchers can gain a deeper understanding of the strengths and limitations of existing approaches, thereby guiding the development of more reliable and trustworthy explainability tools for GNNs.

4. **Application in Real-World Scenarios**: Beyond theoretical advancements, the practical application of GNNs and their explanations holds significant promise for addressing real-world challenges. Future research efforts could focus on deploying GNN-based systems in various domains, such as healthcare, finance, and social sciences, where complex relational data is prevalent. By integrating explainability mechanisms into these applications, stakeholders can gain actionable insights into the decision-making processes of GNN models, fostering trust and transparency.

By embarking on these future research endeavours, we can further propel the field of GNN explainability towards the dual goals of transparency and fairness, ultimately fostering greater trust and acceptance of GNN models in real-world applications.

BIBLIOGRAPHY

[1] C. ABRATE, G. PRETI, AND F. BONCHI, *Counterfactual explanations for graph classification through the lenses of density*, in Explainable Artificial Intelligence - First World Conference, xAI 2023, Lisbon, Portugal, July 26-28, 2023, Proceedings, Part I, vol. 1901 of Communications in Computer and Information Science, Springer, 2023, pp. 324–348.

[2] C. AGARWAL, O. QUEEN, H. LAKKARAJU, AND M. ZITNIK, *Evaluating explainability for graph neural networks*, CoRR, abs/2208.09339 (2022).

[3] C. AGARWAL, M. ZITNIK, AND H. LAKKARAJU, *Probing GNN explainers: A rigorous theoretical and empirical analysis of GNN explanation methods*, in International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event, vol. 151 of Proceedings of Machine Learning Research, PMLR, 2022, pp. 8969–8996.

[4] D. AHMEDT-ARISTIZABAL, M. A. ARMIN, S. DENMAN, C. FOOKES, AND L. PETERSSON, *Graph-based deep learning for medical diagnosis and analysis: Past, present and future*, Sensors, 21 (2021), p. 4758.

[5] K. AMARA, R. YING, Z. ZHANG, Z. HAN, Y. SHAN, U. BRANDES, S. SCHEMM, AND C. ZHANG, *Graphframex: Towards systematic evaluation of explainability methods for graph neural networks*, CoRR, abs/2206.09677 (2022).

[6] K. BACHE AND M. LICHMAN, *UCI machine learning repository*, 2013.

[7] M. BAJAJ, L. CHU, Z. Y. XUE, J. PEI, L. WANG, P. C. LAM, AND Y. ZHANG, *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, 2021, pp. 5644–5655.

[8] P. BOJANOWSKI, A. JOULIN, D. LOPEZ-PAZ, AND A. SZLAM, *Optimizing the latent space of generative networks*, in Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, J. G. Dy and A. Krause, eds., vol. 80 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 599–608.

[9] R. CAI, Y. ZHU, X. CHEN, Y. FANG, M. WU, J. QIAO, AND Z. HAO, *On the probability of necessity and sufficiency of explaining graph neural networks: A lower bound optimization approach*, CoRR, abs/2212.07056 (2022).

[10] J. CHEN, S. WU, A. GUPTA, AND R. YING, *D4explainer: In-distribution GNN explanations via discrete denoising diffusion*, in NeurIPS, 2023.

[11] X. CHEN, Y. DUAN, R. HOUTHOOFT, J. SCHULMAN, I. SUTSKEVER, AND P. ABBEEL, *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, 2016, pp. 2172–2180.

[12] P. CORTEZ AND A. M. G. SILVA, *Using data mining to predict secondary school student performance*, (2008).

[13] M. Z. DARESTANI AND R. HECKEL, *Accelerated MRI with un-trained neural networks*, IEEE Trans. Computational Imaging, 7 (2021), pp. 724–733.

[14] S. DENG, H. RANGWALA, AND Y. NING, *Dynamic knowledge graph based multi-event forecasting*, in KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020, ACM, 2020, pp. 1585–1595.

[15] H. DING, L. CHEN, L. DONG, Z. FU, AND X. CUI, *Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection*, Future Gener. Comput. Syst., 131 (2022), pp. 240–254.

[16] C. DWORK, M. HARDT, T. PITASSI, O. REINGOLD, AND R. S. ZEMEL, *Fairness through awareness*, in Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012, ACM, 2012, pp. 214–226.

[17] J. FAN AND J. CHENG, *Matrix completion by deep matrix factorization*, Neural Networks, 98 (2018), pp. 34–41.

[18] T. GAUDELET, B. DAY, A. R. JAMASB, J. SOMAN, C. REGEP, G. LIU, J. B. R. HAYTER, R. VICKERS, C. ROBERTS, J. TANG, D. ROBLIN, T. L. BLUNDELL, M. M. BRONSTEIN, AND J. P. TAYLOR-KING, *Utilizing graph machine learning within drug discovery and development*, Briefings Bioinform., 22 (2021).

[19] G. B. GOH, N. O. HODAS, C. SIEGEL, AND A. VISHNU, *Smiles2vec: An interpretable general-purpose deep neural network for predicting chemical properties*, arXiv preprint arXiv:1712.02034, (2017).

[20] I. J. GOODFELLOW, J. POUGET-ABADIE, M. MIRZA, B. XU, D. WARDE-FARLEY, S. OZAIR, A. C. COURVILLE, AND Y. BENGIO, *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, 2014, pp. 2672–2680.

[21] C. GRANGER, *Investigating causal relations by econometric models and cross-spectral methods*, in Essays in econometrics: collected papers of Clive WJ Granger, 2001, pp. 31–47.

[22] M. HARDT, E. PRICE, AND N. SREBRO, *Equality of opportunity in supervised learning*, in Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016, pp. 3315–3323.

[23] R. HECKEL AND P. HAND, *Deep decoder: Concise image representations from untrained non-convolutional networks*, in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019.

[24] Q. HUANG, M. YAMADA, Y. TIAN, D. SINGH, AND Y. CHANG, *GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks*, IEEE Transactions on Knowledge and Data Engineering, (2022), pp. 1–6.

[25] Q. HUANG, M. YAMADA, Y. TIAN, D. SINGH, AND Y. CHANG, *Graphlime: Local interpretable model explanations for graph neural networks*, IEEE Trans. Knowl. Data Eng., 35 (2023), pp. 6968–6972.

[26] W. JIANG AND J. LUO, *Graph neural network for traffic forecasting: A survey*, Expert Syst. Appl., 207 (2022), p. 117921.

[27] J. KAZIUS, R. MCGUIRE, AND R. BURSI, *Derivation and validation of toxicophores for mutagenicity prediction*, Journal of Medicinal Chemistry, 48 (2005), pp. 312–320.

[28] T. LI, A. LAHIRI, Y. DAI, AND O. MAYER, *Joint demosaicing and denoising with double deep image priors*, CoRR, abs/2309.09426 (2023).

[29] T. LI, R. MEHTA, Z. QIAN, AND J. SUN, *Rethink autoencoders: Robust manifold learning*, ICML Workshop on Uncertainty and Robustness in Deep Learning, (2020).

[30] T. LI, H. WANG, Z. ZHUANG, AND J. SUN, *Deep random projector: Accelerated deep image prior*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023, IEEE, 2023, pp. 18176–18185.

[31] T. LI, Z. ZHUANG, H. LIANG, L. PENG, H. WANG, AND J. SUN, *Self-validation: Early stopping for single-instance deep generative priors*, in 32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021, BMVA Press, 2021, p. 108.

[32] T. LI, Z. ZHUANG, H. WANG, AND J. SUN, *Random projector: Efficient deep image prior*, in IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023, IEEE, 2023, pp. 1–5.

[33] X. LI, L. SUN, M. LING, AND Y. PENG, *A survey of graph neural network based recommendation in social networks*, Neurocomputing, 549 (2023), p. 126441.

[34] Y. LI, J. ZHOU, Y. DONG, N. SHAFIABADY, AND F. CHEN, *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, United Kingdom, October 21-25, 2023, I. Frommholz, F. Hopfgartner, M. Lee, M. Oakes, M. Lalmas, M. Zhang, and R. L. T. Santos, eds., ACM, 2023, pp. 1259–1267.

[35] Y. LI, J. ZHOU, S. VERMA, AND F. CHEN, *A survey of explainable graph neural networks: Taxonomy and evaluation metrics*, CoRR, abs/2207.12599 (2022).

[36] Y. LI, J. ZHOU, B. ZHENG, AND F. CHEN, *Ganexplainer: Gan-based graph neural networks explainer*, CoRR, abs/2301.00012 (2023).

[37] W. LIN, H. LAN, AND B. LI, *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, vol. 139 of Proceedings of Machine Learning Research, PMLR, 2021, pp. 6666–6679.

[38] W. LIN, H. LAN, H. WANG, AND B. LI, *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, 2022, pp. 13719–13728.

[39] Y. LIU, X. AO, Z. QIN, J. CHI, J. FENG, H. YANG, AND Q. HE, *Pick and choose: A gnn-based imbalanced learning approach for fraud detection*, in WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021, ACM / IW3C2, 2021, pp. 3168–3177.

[40] A. LUCIC, M. A. TER HOEVE, G. TOLOMEI, M. DE RIJKE, AND F. SILVESTRI, *Cf-gnnexplainer: Counterfactual explanations for graph neural networks*, in International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event, vol. 151 of Proceedings of Machine Learning Research, PMLR, 2022, pp. 4499–4511.

[41] D. LUO, W. CHENG, D. XU, W. YU, B. ZONG, H. CHEN, AND X. ZHANG, *Parameterized explainer for graph neural network*, in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[42] J. MA, R. GUO, S. MISHRA, A. ZHANG, AND J. LI, *CLEAR: generative counterfactual explanations on graphs*, in NeurIPS, 2022.

[43] U. S. MALHI, J. ZHOU, A. RASOOL, AND S. SIDDEEQ, *Efficient visual-aware fashion recommendation using compressed node features and graph-based learning*, Machine Learning and Knowledge Extraction, 6 (2024), pp. 2111–2129.

[44] M. MIRZA AND S. OSINDERO, *Conditional generative adversarial nets*, CoRR, abs/1411.1784 (2014).

[45] A. M. NGUYEN, A. DOSOVITSKIY, J. YOSINSKI, T. BROX, AND J. CLUNE, *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, 2016, pp. 3387–3395.

[46] A. ODENA, C. OLAH, AND J. SHLENS, *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, vol. 70 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 2642–2651.

[47] J. J. PARK, P. R. FLORENCE, J. STRAUB, R. A. NEWCOMBE, AND S. LOVEGROVE, *Deepsdf: Learning continuous signed distance functions for shape representation*, in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 165–174.

[48] P. E. POPE, S. KOLOURI, M. ROSTAMI, C. E. MARTIN, AND H. HOFFMANN, *Explainability methods for graph convolutional neural networks*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10772–10781.

[49] T. QIAO, J. ZHANG, D. XU, AND D. TAO, *Mirrorgan: Learning text-to-image generation by redescription*, in IEEE Conference on Computer Vision and Pattern

Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 1505–1514.

[50] A. RADFORD, L. METZ, AND S. CHINTALA, *Unsupervised representation learning with deep convolutional generative adversarial networks*, in 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, Y. Bengio and Y. LeCun, eds., 2016.

[51] L. RIZZO, D. VERDA, S. BERRETTA, AND L. LONGO, *A novel integration of data-driven rule generation and computational argumentation for enhanced explainable ai*, Machine Learning and Knowledge Extraction, 6 (2024), p. 2049.

[52] S. ROY, E. SANGINETO, N. SEBE, AND B. DEMIR, *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, 2018, pp. 684–688.

[53] C. SHAN, Y. SHEN, Y. ZHANG, X. LI, AND D. LI, *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, 2021, pp. 22523–22533.

[54] Z. SONG, Y. ZHANG, AND I. KING, *Towards fair financial services for all: A temporal GNN approach for individual fairness on transaction networks*, in Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023, ACM, 2023, pp. 2331–2341.

[55] C. SU, Y. HOU, AND F. WANG, *Gnn-based biomedical knowledge graph mining in drug development*, Graph Neural Networks: Foundations, Frontiers, and Applications, (2022), pp. 517–540.

[56] J. SUN, L. PENG, T. LI, D. ADILA, Z. ZAIMAN, G. B. MELTON-MEAUX, N. E. INGRAHAM, E. MURRAY, D. BOLEY, S. SWITZER, J. L. BURNS, K. HUANG, T. ALLEN, S. D. STEENBURG, J. W. GICHOYA, E. KUMMERFELD, AND C. J. TIGNANELLI, *Performance of a Chest Radiograph AI Diagnostic Tool for COVID-19: A Prospective Observational Study*, Radiology: Artificial Intelligence, 4 (2022), p. e210217.

[57] R. SUN, H. DAI, AND A. W. YU, *Does GNN pretraining help molecular representation?*, in Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022.

[58] W. SUN, J. XU, W. ZHANG, X. LI, Y. ZENG, AND P. ZHANG, *Funnel graph neural networks with multi-granularity cascaded fusing for protein-protein interaction prediction*, Expert Syst. Appl., 257 (2024), p. 125030.

[59] J. TAN, S. GENG, Z. FU, Y. GE, S. XU, Y. LI, AND Y. ZHANG, *Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning*, in WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022, ACM, 2022, pp. 1018–1027.

[60] S. TAN AND M. L. MAYROVOUNIOTIS, *Reducing data dimensionality through optimizing neural network inputs*, AIChE Journal, 41 (1995), pp. 1471–1480.

[61] E. TJOA AND C. GUAN, *A survey on explainable artificial intelligence (xai): Toward medical xai*, IEEE Transactions on Neural Networks and Learning Systems, (2020).

[62] D. ULYANOV, A. VEDALDI, AND V. S. LEMPITSKY, *Deep image prior*, in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 9446–9454.

[63] A. VAN DEN OORD, N. KALCHBRENNER, L. ESPEHOLT, K. KAVUKCUOGLU, O. VINYALS, AND A. GRAVES, *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, 2016, pp. 4790–4798.

[64] F. VAN MOURIK, A. JUTTE, S. E. BERENDSE, F. A. BUKHSH, AND F. AHMED, *Tertiary review on explainable artificial intelligence: Where do we stand?*, Machine Learning and Knowledge Extraction, 6 (2024), pp. 1997–2017.

[65] C. VONDRICK, H. PIRSIAVASH, AND A. TORRALBA, *Generating videos with scene dynamics*, in Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, eds., 2016, pp. 613–621.

[66] A. WAHEED, M. GOYAL, D. GUPTA, A. KHANNA, F. M. AL-TURJMAN, AND P. R. PINHEIRO, *Covidgan: Data augmentation using auxiliary classifier GAN for improved covid-19 detection*, CoRR, abs/2103.05094 (2021).

[67] N. WALE, I. A. WATSON, AND G. KARYPIS, *Comparison of descriptor spaces for chemical compound retrieval and classification*, Knowl. Inf. Syst., 14 (2008), pp. 347–375.

[68] C. WANG, Z. LIN, X. YANG, J. SUN, M. YUE, AND C. SHAHABI, *HAGEN: homophily-aware graph convolutional recurrent network for crime forecasting*, in Thirty-

Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, AAAI Press, 2022, pp. 4193–4200.

[69] H. WANG, T. LI, Z. ZHUANG, T. CHEN, H. LIANG, AND J. SUN, *Early stopping for deep image prior*, Transactions on Machine Learning Research, (2023).

[70] X. WANG, Y. WU, A. ZHANG, F. FENG, X. HE, AND T. CHUA, *Reinforced causal explainer for graph neural networks*, IEEE Trans. Pattern Anal. Mach. Intell., 45 (2023), pp. 2297–2309.

[71] Y. WU, D. LIAN, Y. XU, L. WU, AND E. CHEN, *Graph convolutional networks with markov random field reasoning for social spammer detection*, vol. 34, Apr. 2020, pp. 1054–1061.

[72] J. XIONG, Z. XIONG, K. CHEN, H. JIANG, AND M. ZHENG, *Graph neural networks for automated de novo drug design*, Drug discovery today, 26 (2021), pp. 1382–1393.

[73] L. YANG, Z. LIU, Y. DOU, J. MA, AND P. S. YU, *Consisrec: Enhancing GNN for social recommendation via consistent neighbor aggregation*, in SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, ACM, 2021, pp. 2141–2145.

[74] Z. YANG, W. PEI, M. CHEN, AND C. YUE, *WTAGRAPH: web tracking and advertising detection using graph neural networks*, in 43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022, IEEE, 2022, pp. 1540–1557.

[75] R. YING, D. BOURGEOIS, J. YOU, M. ZITNIK, AND J. LESKOVEC, *GNN explainer: A tool for post-hoc explanation of graph neural networks*, CoRR, abs/1903.03894 (2019).

[76] Z. YING, D. BOURGEOIS, J. YOU, M. ZITNIK, AND J. LESKOVEC, *Gnnexplainer: Generating explanations for graph neural networks*, in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 9240–9251.

[77] H. YUAN, J. TANG, X. HU, AND S. JI, *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, 2020, pp. 430–438.

[78] H. YUAN, J. TANG, X. HU, AND S. JI, *XGNN: Towards Model-Level Explanations of Graph Neural Networks*, in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event CA USA, Aug. 2020, ACM, pp. 430–438.

[79] H. YUAN, H. YU, S. GUI, AND S. JI, *Explainability in graph neural networks: A taxonomic survey*, CoRR, abs/2012.15445 (2020).

[80] H. YUAN, H. YU, J. WANG, K. LI, AND S. JI, *On explainability of graph neural networks via subgraph explorations*, in Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, vol. 139 of Proceedings of Machine Learning Research, PMLR, 2021, pp. 12241–12252.

[81] Y. ZHANG, D. DEFAZIO, AND A. RAMESH, *Relex: A model-agnostic relational model explainer*, arXiv preprint arXiv:2006.00305, (2020).

[82] Y. ZHAO, Y. WANG, AND T. DERR, *Fairness and explainability: Bridging the gap towards fair model explanations*, in Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, AAAI Press, 2023, pp. 11363–11371.

[83] J. ZHOU, A. H. GANDOMI, F. CHEN, AND A. HOLZINGER, *Evaluating the quality of machine learning explanations: A survey on methods and metrics*, Electronics, 10 (2021), p. 593.

[84] Z. ZHOU, X. ZHAI, AND C. TIN, *Fully automatic electrocardiogram classification system based on generative adversarial network with auxiliary classifier*, Expert Syst. Appl., 174 (2021), p. 114809.

[85] Z. ZHUANG, T. LI, H. WANG, AND J. SUN, *Blind Image Deblurring with Unknown Kernel Size and Substantial Noise*, International Journal of Computer Vision, (2023).