Towards Faithful and Consistent Explanations for Graph Neural Networks

Tianxiang Zhao
College of Information Sciences and Technology, The Pennsylvania State University
State College, USA
tkz5084@psu.edu

Dongsheng Luo
Knight Foundation School of Computing and Information Sciences, Florida International University
Miami, USA
dluo@fiu.edu

Xiang Zhang
College of Information Sciences and Technology, The Pennsylvania State University
State College, USA
xzz89@psu.edu

Suhang Wang
College of Information Sciences and Technology, The Pennsylvania State University
State College, USA
szw494@psu.edu

ABSTRACT

Uncovering rationales behind predictions of graph neural networks (GNNs) has received increasing attention over recent years. Instance-level GNN explanation aims to discover critical input elements, like nodes or edges, that the target GNN relies upon for making predictions. Though various algorithms are proposed, most of them formalize this task by searching the minimal subgraph which can preserve original predictions. However, an inductive bias is deep-rooted in this framework: several subgraphs can result in the same or similar outputs as the original graphs. Consequently, they have the danger of providing spurious explanations and fail to provide consistent explanations. Applying them to explain weakly-performed GNNs would further amplify these issues. To address this problem, we theoretically examine the predictions of GNNs from the causality perspective. Two typical reasons of spurious explanations are identified: confounding effect of latent variables like distribution shift, and causal factors distinct from the original input. Observing that both confounding effects and diverse causal rationales are encoded in internal representations, we propose a simple yet effective countermeasure by aligning embeddings. Concretely, concerning potential shifts in the high-dimensional space, we design a distribution-aware alignment algorithm based on anchors. This new objective is easy to compute and can be incorporated into existing techniques with no or little effort. Theoretical analysis shows that it is in effect optimizing a more faithful explanation objective in design, which further justifies the proposed approach.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; *Statistical relational learning*.

WSDM '23, February 27-March 3, 2023, Singapore, Singapore

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570421

KEYWORDS

graph neural networks, explainability

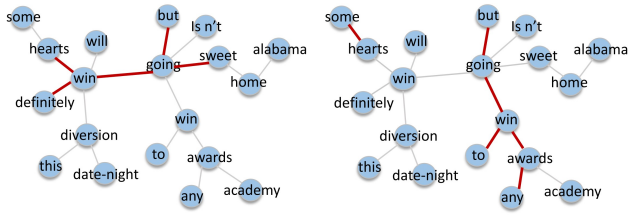
ACM Reference Format:
Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. 2023. Towards Faithful and Consistent Explanations for Graph Neural Networks . In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570421

1 INTRODUCTION

Graph-structured data is ubiquitous in the real world, such as social networks [11, 42], molecular structures [4, 19] and knowledge graphs [25]. With the growing interest in learning from graphs, graph neural networks (GNNs) are receiving more and more attention over the years. Generally, GNNs adopt message-passing mechanisms, which recursively propagate and fuse messages from neighbor nodes on the graphs. GNNs have achieved state-of-the-art performance for many tasks such as node classification [13, 15, 28], graph classification [36], and link prediction [40].

Despite the success of GNNs for various domains, as with other neural networks, GNNs lack interpretability. Understanding the inner working of GNNs can bring several benefits. First, it enhances practitioners' trust in the GNN model by enriching their understanding of the network characteristics. Second, it increases the models' transparency to enable trusted applications in decision-critical fields sensitive to fairness, privacy and safety challenges, such as healthcare and drug discovery [22]. Thus, studying the explainability of GNNs is attracting increasing attention.

Particularly, we focus on post-hoc instance-level explanations. Given a trained GNN and an input graph, this task seeks to discover the substructures that can explain the prediction behavior of the GNN model. Some solutions have been proposed in existing works [14, 29, 37]. For example, in search of important substructures that predictions rely upon, GNNExplainer learns an importance matrix on nodes and edges via perturbation [37]. The identified minimal substructures that preserve original predictions are taken as the explanation. Extending this idea, PGExplainer trains a graph generator to utilize global information in explanation and enable faster inference in the inductive setting [18]. SubgraphX constraint explanations as connected subgraphs and conduct Monte Carlo



(a) Run 1 (b) Run 2

Figure 1: Explanation results achieved by a leading baseline GNNExplainer on the same input graph from Graph-SST5. Red edges formulate explanation substructures.

tree search based on Shapley value [39]. These methods can be summarized in a label preserving framework, i.e., candidate explanation is formed as a masked version of the original graph and is identified as the minimal discriminative substructure that preserves the predicted label.

However, due to the complexity of topology and the combinatory number of candidate substructures, existing label preserving methods are insufficient for a faithful and consistent explanation of GNNs. They are unstable and are prone to give spurious correlations as explanations. A failure case is shown in Fig. 1, where a GNN is trained on Graph-SST5 [38] for sentiment classification. Each node represents a word and each edge denotes syntactic dependency between nodes. Each graph is labeled based on the sentiment of the sentence. In the figure, the sentence "Sweet home alabama isn't going to win any academy awards, but this date-night diversion will definitely win some hearts" is labeled *positive*. In the first run, GNNExplainer [37] identifies the explanation as "definitely win some hearts", and in the second run, it identifies "win academy awards" to be the explanation instead. Different explanations obtained by GNNExplainer break the criteria of **consistency**, i.e., the explanation method should be deterministic and consistent with the same input for different runs [20]. Consequently, explanations provided by the existing methods may fail to faithfully reflect the decision mechanism of the to-be-explained GNN.

Inspecting the inference process of target GNNs, we find that the inconsistency problem and spurious explanations can be understood from the causality perspective. Specifically, existing explanation methods may lead to spurious explanations either as a result of different causal factors or due to the confounding effect of distribution shifts (identified subgraphs may be out of distribution). These failure cases originate from a particular inductive bias that predicted labels are sufficiently indicative for extracting critical input components. This underlying assumption is rooted in optimization objectives adopted by existing works [18, 37, 39]. However, our analysis demonstrates that the label information is insufficient to filter out spurious explanations, leading to inconsistent and unfaithful explanations.

Considering the inference of GNNs, both confounding effects and distinct causal relationships can be reflected in the internal representation space. With this observation, we propose a novel objective that encourages alignment of embeddings of raw graph and identified subgraph in internal embedding space to obtain more faithful and consistent GNN explanations. Concretely, we propose two strategies, a distribution-aware method based on anchors and a simplified one based on absolute distance. Both variants are flexible to be incorporated into various existing GNN explanation methods. Further analysis shows that the proposed method is in fact optimizing a new explanation framework, which is more faithful in design. We summarized our contributions as follows.

- We point out the faithfulness and consistency issues in rationales identified by existing GNN explanation models. These issues arise due to the inductive bias in their label-preserving framework, which only uses predictions as the guiding information.
- We propose an effective and easy-to-apply countermeasure by aligning intermediate embeddings, which is flexible to be incorporated into various GNN explanation techniques. We further conduct theoretical analysis to understand and validate the proposed framework.
- Extensive experiments on real-world and synthetic datasets show that our framework benefits various GNN explanation models to achieve more faithful and consistent explanations.

2 RELATED WORK

2.1 Graph Neural Networks

Graph neural networks (GNNs) are developing rapidly in recent years, with the increasing need for learning on relational data structures [5, 11, 43]. Generally, existing GNNs can be categorized into two categories, i.e., spectral-based approaches [3, 15, 26] based on graph signal processing theory, and spatial-based approaches [1, 8, 34] relying upon neighborhood aggregation. Despite their differences, most GNN variants can be summarized with the message-passing framework, which is composed of pattern extraction and interaction modeling within each layer [12]. Specifically, GNNs model messages from node representations. These messages are then propagated with various message-passing mechanisms to refine node representations, which are then utilized for downstream tasks [13, 40, 43]. Explorations are made by disentangling the propagation process [30, 33, 44] or utilizing external prototypes [16, 35]. Despite their success in network representation learning, GNNs are uninterpretable black box models. It is challenging to understand their behaviors even if the adopted message passing mechanism and parameters are given. Besides, unlike traditional deep neural networks where instances are identically and independently distributed, GNNs consider node features and graph topology jointly, making the interpretability problem more challenging to handle.

2.2 GNN Interpretation Methods

Recently, some efforts have been taken to interpret GNN models and provide explanations for their predictions [31]. Existing methods can be generally grouped into two categories based on the granularity: (1) instance-level explanation [37], which provides explanations on the prediction for each instance by identifying important substructures; and (2) model-level explanation [2, 41], which aims to understand global decision rules captured by the target GNN. From the methodology perspective, existing methods can be categorized as (1) self-explainable GNNs [6, 41], where the GNN can simultaneously give prediction and explanations on the prediction; and (2) post-hoc explanations [18, 37, 39], which adopt another model or strategy to provide explanations of a target GNN. As post-hoc explanations are model-agnostic, i.e., can be applied

for various GNNs, in this work, we focus on post-hoc instance-level explanations [37], i.e., given a trained GNN model, identifying instance-wise critical substructures for each input to explain the prediction. A comprehensive survey can be found in [7].

A variety of strategies for post-hoc instance-level explanations have been explored in the literature, including utilizing signals from gradients [2, 21], perturbed predictions [18, 24, 37, 39], and decomposition [2, 23]. Among these methods, perturbed prediction-based methods are the most popular. The basic idea is to learn a perturbation mask that filters out non-important connections and identifies dominating substructures preserving the original predictions [38]. The identified important substructure is used as an explanation for the prediction. For example, GNNExplainer [37] employs two soft mask matrices on node attributes and graph structure, respectively, which are learned end-to-end under the maximizing mutual information (MMI) framework. PGExplainer [18] extends it by incorporating a graph generator to utilize global information. It can be applied in the inductive setting and prevent the onerous task of re-learning from scratch for each to-be-explained instance. SubgraphX [39] employs Monte Carlo Tree Search (MCTS) to find connected subgraphs that preserve predictions as explanations.

Despite progress in interpreting GNNs, most of these methods discover critical substructures merely upon the change of outputs given perturbed inputs. Due to this underlying inductive bias, existing label-preserving methods are heavily affected by spurious correlations caused by confounding factors in the environment. On the other hand, by aligning intermediate embeddings in GNNs, our method alleviates the effects of spurious correlations on interpreting GNNs, leading to faithful and consistent explanations.

3 PRELIMINARY

3.1 Problem Definition

We use $\mathcal{G} = \{\mathcal{V}, \mathcal{E}; \mathbf{F}, \mathbf{A}\}$ to denote a graph, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is a set of n nodes and $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the set of edges. Nodes are accompanied by an attribute matrix $\mathbf{F} \in \mathbb{R}^{n \times d}$, and $\mathbf{F}[i,:] \in \mathbb{R}^{1 \times d}$ is the d-dimensional node attributes of node v_i. $\mathcal{E}$ is described by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. $A_{ij} = 1$ if there is an edge between node v_i and v_j; otherwise, $A_{ij} = 0$. For *graph classification*, each graph $\mathcal{G}_i$ has a label $Y_i \in C$, and a GNN model f is trained to map $\mathcal{G}$ to its class, i.e., $f : \{\mathbf{F}, \mathbf{A}\} \mapsto \{1, 2, \ldots, C\}$. Similarly, for *node classification*, each graph $\mathcal{G}_i$ denotes a K-hop subgraph centered at node v_i and a GNN model f is trained to predict the label of v_i based on node representation of v_i learned from $\mathcal{G}_i$. The purpose of explanation is to find a subgraph $\mathcal{G}'$, marked with binary importance mask $\mathbf{M}_A \in [0,1]^{n \times n}$ on adjacency matrix and $\mathbf{M}_F \in [0,1]^{n \times d}$ on node attributes, respectively, e.g., $\mathcal{G}' = \{\mathbf{A} \odot \mathbf{M}_A; \mathbf{F} \odot \mathbf{M}_F\}$, where $\odot$ denote elementwise multiplication. These two masks highlight components of $\mathcal{G}$ that are important for f to predict its label. With the notations, the *post-hoc instance-level* GNN explanation task is defined as:

Given a trained GNN model f, for an arbitrary input graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}; \mathbf{F}, \mathbf{A}\}$, find a subgraph $\mathcal{G}'$ that can explain the prediction of f on $\mathcal{G}$. Obtained explanation $\mathcal{G}'$ is depicted by importance mask $\mathbf{M}_F$ on node attributes and importance mask $\mathbf{M}_A$ on adjacency matrix.

3.2 MMI-based Explanation Framework

Many approaches have been designed for post-hoc instance-level GNN explanation. Due to the discreteness of edge existence and non-grid graph structures, most works apply a perturbation-based strategy to search for explanations. Typically, they can be summarized as Maximization of Mutual Information (MMI) between predicted label $\hat{Y}$ and explanation $\mathcal{G}'$, i.e.,

$$\begin{aligned} &\min_{\mathcal{G}'} -I(\hat{Y}, \mathcal{G}'), \\ &\text{s.t.} \quad \mathcal{G}' \sim \mathcal{P}(\mathcal{G}, \mathbf{M}_A, \mathbf{M}_F), \quad \mathcal{R}(\mathbf{M}_F, \mathbf{M}_A) \leq c \end{aligned} \tag{1}$$

where $I()$ represents mutual information and $\mathcal{P}$ denotes the perturbations on original input with importance masks $\{\mathbf{M}_F, \mathbf{M}_A\}$. For example, let $\{\hat{\mathbf{A}}, \hat{\mathbf{F}}\}$ represent the perturbed $\{\mathbf{A}, \mathbf{F}\}$. Then, $\hat{\mathbf{A}} = \mathbf{A} \odot \mathbf{M}_A$ and $\hat{\mathbf{F}} = \mathbf{Z} + (\mathbf{F} - \mathbf{Z}) \odot \mathbf{M}_F$ in GNNExplainer [37], where $\mathbf{Z}$ is sampled from marginal distribution of node attributes $\mathbf{F}$. $\mathcal{R}$ denotes regularization terms on the explanation, imposing prior knowledge into the searching process, like constraints on budgets or connectivity distributions [18]. Mutual information $I(\hat{Y}, \mathcal{G}')$ quantifies consistency between original predictions $\hat{Y} = f(\mathcal{G})$ and prediction of candidate explanation $f(\mathcal{G}')$, which promotes the positiveness of found explanation $\mathcal{G}'$. Since mutual information measures the predictive power of $\mathcal{G}'$ on Y, this framework essentially tries to find a subgraph that can best predict the original output $\hat{Y}$. During training, a relaxed version [37] is often adopted as:

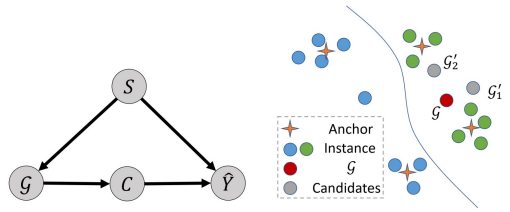
$$\begin{aligned} &\min_{\mathcal{G}'} H_C\big(\hat{Y}, P(\hat{Y}' \mid \mathcal{G}')\big), \\ &\text{s.t.} \quad \mathcal{G}' \sim \mathcal{P}(\mathcal{G}, \mathbf{M}_A, \mathbf{M}_F), \quad \mathcal{R}(\mathbf{M}_F, \mathbf{M}_A) \leq c \end{aligned} \tag{2}$$

where H_C denotes cross-entropy. With this same objective, existing methods mainly differ from each other in searching strategies.

Different aspects regarding the quality of explanations can be evaluated [20]. Among them, two most important criteria are **faithfulness** and **consistency**. Faithfulness measures the descriptive accuracy of explanations, indicating how truthful they are compared to behaviors of the target model. Consistency considers explanation invariance, which checks that identical input should have identical explanations. However, as shown in Figure 1, the existing MMI-based framework is sub-optimal in terms of these criteria. The cause of this problem is rooted in its learning objective, which uses prediction alone as guidance in search of explanations. A detailed analysis will be provided in the next section.

4 ANALYZE SPURIOUS EXPLANATIONS

With "spurious explanations", we refer to those explanations lies outside the genuine rationale of prediction on $\mathcal{G}$, making the usage of $\mathcal{G}'$ as explanations anecdotal. As examples in Figure 1, despite rapid developments in explaining GNNs, the problem w.r.t faithfulness and consistency of detected explanations remains. To get a deeper understanding of reasons behind this problem, we can examine behavior of target GNN model from the causality perspective. Figure 2(a) shows the Structural Equation Model (SEM), where variable C denotes discriminative causal factors and variable S represents confounding environment factors. Two paths between $\mathcal{G}$ and the predicted label $\hat{Y}$ can be found.



(a) SCM (b) Alignment

Figure 2: (a) Prediction rules of f, in the form of SCM. (b) An example of anchor-based embedding alignment.

- $\mathcal{G} \rightarrow C \rightarrow \hat{Y}$: This path presents the inference of target GNN model, i.e., critical patterns C that are informative and discriminative for the prediction would be extracted from input graph, upon which the target model is dependent. Causal variables are determined by both the input graph and learned knowledge by the target GNN model.
- $\mathcal{G} \leftarrow S \rightarrow \hat{Y}$: We denote S as the confounding factors, such as depicting the overall distribution of graphs. It is causally related to both the appearance of input graphs and prediction of target GNN models. Masked version of $\mathcal{G}$ could create out-of-distribution (OOD) examples, resulting in spurious causality to prediction outputs. For example in the chemical domain, removing edges (bonds) or nodes (atoms) may obtain invalid molecular graphs that never appear during training. In the existence of distribution shifts, model predictions would be less reliable.

Figure 2(a) provides us with a tool to analyze f's behaviors. From obtained causal structures, we can observe that spurious explanations may arise as a result of failure in recovering original causal rationale. $\mathcal{G}'$ learned from Equation 1 may preserve prediction $\hat{Y}$ due to confounding effect of distribution shift or different causal variables C compared to original $\mathcal{G}$. Weakly-trained GNN $f(\cdot)$ that are unstable or non-robust towards noises would further amplify this problem as the prediction is unreliable.

To further understand the issue, we can build the correspondence from SEM in Figure 2(a) to the inference process of GNN f. Decomposing $f()$ as feature extractor $f_{ext}()$ and classifier $f_{cls}()$, its inference can be summarized as two steps: (1) encoding step with $f_{ext}()$, which takes $\mathcal{G}$ as input and produce its embedding in the representation space E_C; (2) classification step with $f_{cls}()$, which predicts output labels on input's embedding. Connecting these inference steps to SEM in Figure 2(a), we can find that:

- The causal path $\mathcal{G} \rightarrow C \rightarrow \hat{Y}$ lies behind the inference process with representation space E_C to encode critical variables C;
- The confounding effect of distribution shift S works on the inference process via influencing distribution of graph embedding in E_C. When masked input $\mathcal{G}'$ is OOD, its embedding would fail to reflect its discriminative features and deviate from real distributions, hence deviating the classification step on it.

To summarize, we can observe that spurious explanations are usually obtained due to the following two reasons:

(1) The obtained $\mathcal{G}'$ is OOD graph. During inference of target GNN model, the encoded representation of $\mathcal{G}'$ is distant from those seen in the training set, making the prediction unreliable;

(2) The encoded discriminative representation does not accord with that of the original graph. Different causal factors (C) are extracted between $\mathcal{G}'$ and $\mathcal{G}$, resulting in false explanations.

5 METHODOLOGY

Based on the discussion above, in this section, we focus on improving the faithfulness and consistency of GNN explanations and correcting the inductive bias caused by simply relying on prediction outputs. We first provide an intuitive introduction to the proposed countermeasure, which takes the internal inference process into account. We then design two concrete algorithms to align $\mathcal{G}$ and $\mathcal{G}'$ in the latent space, to promote that they are seen and processed in the same manner. Finally, theoretical analysis is provided to justify our strategies.

5.1 Alleviate Spurious Explanations

Instance-level post-hoc explanation dedicates to finding discriminative substructures that the target model f depends upon. The traditional objective in Equation 2 can identify minimal predictive parts of input, however, it is dangerous to directly take them as explanations. Due to diversity in graph topology and combinatory nature of sub-graphs, multiple distinct substructures could be identified leading to the same prediction, as discussed in Section 4.

For an explanation substructure $\mathcal{G}'$ to be faithful, it should follow the same rationale as the original graph $\mathcal{G}$ inside the internal inference of to-be-explained model f. To achieve this goal, the explanation $\mathcal{G}'$ should be aligned to $\mathcal{G}$ w.r.t the decision mechanism, reflected in Figure 2(a). However, it is non-trivial to extract and compare the critical causal variables C and confounding variables S due to the black box nature of the target GNN model to be explained.

Following the causal analysis in Section 4, we propose to take an alternative approach by looking into internal embeddings learned by f. Causal variables C are encoded in representation space extracted by f, and out-of-distribution effects can also be reflected by analyzing embedding distributions. An assumption can be safely made: *if two graphs are mapped to embeddings near each other by a GNN layer, then these graphs are seen as similar by it and would be processed similarly by following layers.* With this assumption, a proxy task can be designed by aligning internal graph embeddings between $\mathcal{G}'$ and $\mathcal{G}$. This new task can be incorporated into Framework 1 as an auxiliary optimization objective.

Let $\mathbf{h}_v^l$ be the representation of node v at the l-th GNN layer with $\mathbf{h}_v^0 = \mathbf{F}[v,:]$. Generally, the inference process inside GNN layers can be summarized as a message-passing framework:

$$\begin{aligned} \mathbf{m}_v^{l+1} &= \sum_{u \in \mathcal{N}(v)} \text{Message}_l(\mathbf{h}_v^l, \mathbf{h}_u^l, A_{v,u}), \\ \mathbf{h}_v^{l+1} &= \text{Update}_l(\mathbf{h}_v^l, \mathbf{m}_v^{l+1}), \end{aligned} \tag{3}$$

where Message_l and Update_l are the message function and update function at l-th layer, respectively. $\mathcal{N}(v)$ is the set of node v's neighbors. Without loss of generality, the graph pooling layer can also be presented as:

$$\mathbf{h}_{v'}^{l+1} = \sum_{v \in \mathcal{V}} P_{v,v'} \cdot \mathbf{h}_v^l. \tag{4}$$

where $P_{v,v'}$ denotes mapping weight from node v in layer l to node v' in layer $l+1$ inside the myriad of GNN for graph classification.

We propose to align embedding $\mathbf{h}_v^{l+1}$ at each layer, which contains both node and local neighborhood information.

5.2 Distribution-Aware Alignment

Achieving alignment in the embedding space is not straightforward. It has several distinct difficulties. (1) It is difficult to evaluate the distance between $\mathcal{G}$ and $\mathcal{G}'$ in this embedding space. Different dimensions could encode different features and carry different importance. Furthermore, $\mathcal{G}'$ is a substructure of the original $\mathcal{G}$ and a shift on unimportant dimensions would naturally exist. (2) Due to the complexity of graph/node distributions, it is non-trivial to design a measurement of alignments that is both computation-friendly and can correlate well to distance on the distribution manifold.

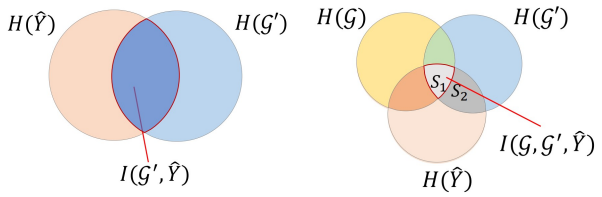
To address these challenges, we design a strategy to identify explanatory substructures and preserve their alignment with original graphs in a distribution-aware manner. The basic idea is to utilize other graphs to obtain a global view of the distribution density of embeddings, providing a better measurement of alignment. Concretely, we obtain representative node/graph embeddings as anchors and use distances to these anchors as the distribution-wise representation of graphs. Alignment is conducted on obtained representation of graph pairs. Next, we go into details of this strategy.

- First, using graphs $\{\mathcal{G}_i\}_{i=1}^m$ from the same dataset, a set of node embeddings can be obtained as $\{\{\mathbf{h}_{v,i}^l\}_{v\in\mathcal{V}_i'}\}_{i=1}^m$ for each layer l, where $\mathbf{h}_{v,i}$ denotes embedding of node v in graph $\mathcal{G}_i$. For node-level tasks, we set $\mathcal{V}_i'$ to contain only the center node of graph $\mathcal{G}_i$. For graph-level tasks, $\mathcal{V}_i'$ contains nodes set after graph pooling layer, and we process them following $\{\sum_{v\in\mathcal{V}_i'}\mathbf{h}_{v,i}^{l+1}/|\mathcal{V}_i'|\}_{i=1}^m$ to get global graph representation.
- Then, a clustering algorithm is applied to obtained embedding set to get K groups. Clustering centers of these groups are set to be anchors, annotated as $\{\mathbf{h}^{l+1,k}\}_{k=1}^K$. In experiments, we select DBSCAN [9] as the clustering algorithm, and tune its hyper-parameters to get around 20 groups.
- At l-th layer, $\mathbf{h}_v^{l+1}$ is represented in terms of relative distances to those K anchors, as $\mathbf{s}_v^l \in \mathbb{R}^{1\times K}$ with the k-th element calculated as $\mathbf{s}_{v,k}^l = \|\mathbf{h}_v^{l+1} - \mathbf{h}_v^{l+1,k}\|_2$.

Alignment between $\mathcal{G}'$ and $\mathcal{G}$ can be achieved by comparing their representations at each layer. The alignment loss is computed as:

$$\mathcal{L}_{align}(f(\mathcal{G}), f(\mathcal{G}')) = \sum_l \sum_{v\in\mathcal{V}'} \|\mathbf{s}_v^l - \hat{\mathbf{s}}_v^l\|_2^2. \tag{5}$$

This metric provides a lightweight strategy for evaluating alignments in the embedding distribution manifold, by comparing relative positions w.r.t representative clustering centers. This strategy can naturally encode the varying importance of each dimension. Fig. 2(b) gives an example, where $\mathcal{G}$ is the graph to be explained and the red stars are anchors. $\mathcal{G}_1'$ and $\mathcal{G}_2'$ are both similar to $\mathcal{G}$ w.r.t absolute distances; while it is easy to see $\mathcal{G}_1'$ is more similar to $\mathcal{G}$ w.r.t to the anchors. In other words, the anchors can better measure the alignment to filter out spurious explanations.



(a) Previous Objective (b) Our Proposed New Objective

Figure 3: Illustration of our proposed new objective.

This alignment loss is used as an auxiliary task incorporated into MMI-based framework in Equation 2 to get faithful explanation as:

$$\begin{aligned} &\min_{\mathcal{G}'} H_C\big(\hat{Y}, P(\hat{Y}' \mid \mathcal{G}')\big) + \lambda \cdot \mathcal{L}_{Align}, \\ &\text{s.t.} \quad \mathcal{G}' \sim \mathcal{P}(\mathcal{G}, \mathbf{M}_A, \mathbf{M}_F), \quad \mathcal{R}(\mathbf{M}_F, \mathbf{M}_A) \le c \end{aligned} \tag{6}$$

where λ controls the balance between prediction preservation and embedding alignment. $\mathcal{L}_{Align}$ is flexible to be incorporated into various existing explanation methods.

As a simpler and more direct implementation, we also design a variant based on absolute distance. For layers without graph pooling, the objective can be written as $\sum_l \sum_{v\in\mathcal{V}} \|\mathbf{h}_v^l - \hat{\mathbf{h}}_v^l\|_2^2$. For layers with graph pooling, as the structure could be different, we conduct alignment on global representation $\sum_{v\in\mathcal{V}'} \mathbf{h}_v^{l+1}/|\mathcal{V}'|$, where $\mathcal{V}'$ denotes node set after pooling.

5.3 Theoretical Analysis

5.3.1 New Explanation Objective. From previous discussions, it is shown that $\mathcal{G}'$ obtained via Equation 1 cannot be safely used as explanations. One main drawback of existing GNN explanation methods lies in the inductive bias that the same outcomes do not guarantee the same causes, leaving existing approaches vulnerable towards spurious explanations. An illustration is given in Figure 3. Objective proposed in Equation 1 optimizes the mutual information between explanation candidate $\mathcal{G}'$ and $\hat{Y}$, corresponding to maximize the overlapping between $H(\mathcal{G}')$ and $H(\hat{Y})$ in Figure 3(a), or region $S_1 \cup S_2$ in Figure 3(b). Here, H denotes information entropy. However, this learning target cannot prevent the danger of generating spurious explanations. Provided $\mathcal{G}'$ may fall into the region S_2, which cannot faithfully represent graph $\mathcal{G}$. Instead, a more sensible objective should be maximizing region S_1 in Figure 3(b). The intuition behind this is that in the search input space that causes the same outcome, identified $\mathcal{G}'$ should account for both representative and discriminative parts of original $\mathcal{G}$, to prevent spurious explanations that produce the same outcomes due to different causes. Concretely, finding $\mathcal{G}'$ that maximize S_1 can be formalized as:

$$\begin{aligned} &\min_{\mathcal{G}'} -I(\mathcal{G}, \mathcal{G}', \hat{Y}), \\ &\text{s.t.} \quad \mathcal{G}' \sim \mathcal{P}(\mathcal{G}, \mathbf{M}_A, \mathbf{M}_F) \quad \mathcal{R}(\mathbf{M}_F, \mathbf{M}_A) \le c \end{aligned} \tag{7}$$

5.3.2 Connecting to Our Method. $I(\mathcal{G}, \mathcal{G}', \hat{Y})$ is intractable as the latent generation mechanism of $\mathcal{G}$ is unknown. In this part, we expand this objective, connect it to Equation 6, and construct its

proxy optimizable form as:

$$I(\mathcal{G}, \mathcal{G}', \hat{Y}) = \sum_{y\sim\hat{Y}} \sum_{\mathcal{G}} \sum_{\mathcal{G}'} P(\mathcal{G}, \mathcal{G}', y) \cdot \log \frac{P(\mathcal{G}', y)P(\mathcal{G}, \mathcal{G}')P(\mathcal{G}, y)}{P(\mathcal{G}, \mathcal{G}', y)P(\mathcal{G})P(\mathcal{G}')P(y)}$$

$$= \sum_{y\sim\hat{Y}} \sum_{\mathcal{G}} \sum_{\mathcal{G}'} P(\mathcal{G}, \mathcal{G}', y) \cdot \log \left[\frac{P(\mathcal{G}', y)}{P(\mathcal{G}')P(y)} \cdot \frac{P(\mathcal{G}, \mathcal{G}')}{P(\mathcal{G})P(\mathcal{G}')} \cdot \frac{P(\mathcal{G}, y)}{P(\mathcal{G}, y|\mathcal{G}')} \right]$$

$$= \sum_{y\sim\hat{Y}} \sum_{\mathcal{G}'} P(\mathcal{G}', y) \cdot \log \frac{P(\mathcal{G}', y)}{P(\mathcal{G}')P(y)} + \sum_{\mathcal{G}} \sum_{\mathcal{G}'} P(\mathcal{G}, \mathcal{G}') \cdot \log \frac{P(\mathcal{G}, \mathcal{G}')}{P(\mathcal{G})P(\mathcal{G}')}$$

$$- \sum_{\mathcal{G}'} \sum_{y\sim\hat{Y}} \sum_{\mathcal{G}} P(\mathcal{G}, y, \mathcal{G}') \cdot \log \frac{P(\mathcal{G}, y, \mathcal{G}')}{P(\mathcal{G}, y)P(\mathcal{G}')}]$$

$$= I(\mathcal{G}', \hat{Y}) + I(\mathcal{G}, \mathcal{G}') - \sum_{y\sim\hat{Y}} \sum_{\mathcal{G}} P(\mathcal{G}, y) \sum_{\mathcal{G}'} P(\mathcal{G}'|\mathcal{G}, y) \cdot \log P(\mathcal{G}'|\mathcal{G}, y)$$

$$+ \sum_{\mathcal{G}'} \sum_{y\sim\hat{Y}} \sum_{\mathcal{G}} P(\mathcal{G}, y, \mathcal{G}') \cdot \log P(\mathcal{G}')$$

$$= I(\mathcal{G}', \hat{Y}) + I(\mathcal{G}, \mathcal{G}') + H(\mathcal{G}'|\mathcal{G}, \hat{Y}) - H(\mathcal{G}').$$

Since both $H(\mathcal{G}'|\mathcal{G}, \hat{Y})$ and $H(\mathcal{G}')$ depicts entropy of explanation $\mathcal{G}'$ and are closely related to perturbation budgets, we can neglect these two terms and get a surrogate optimization objective for $\max_{\mathcal{G}'} I(\mathcal{G}, \mathcal{G}', \hat{Y})$ as $\max_{\mathcal{G}'} I(\hat{Y}, \mathcal{G}') + I(\mathcal{G}', \mathcal{G})$.

In $\max_{\mathcal{G}'} I(\hat{Y}, \mathcal{G}') + I(\mathcal{G}', \mathcal{G})$, the first term $\max_{\mathcal{G}'} I(\hat{Y}, \mathcal{G}')$ is the same as Eq.(1). Following [37], We relax it as $\min_{\mathcal{G}'} H_C(\hat{Y}, \hat{Y}'|\mathcal{G}')$, optimizing $\mathcal{G}'$ to preserve original prediction outputs. The second term, $\max_{\mathcal{G}'} I(\mathcal{G}', \mathcal{G})$, corresponds to maximizing consistency between $\mathcal{G}'$ and $\mathcal{G}$. Although the graph generation process is latent, with the safe assumption that embedding $\mathbf{E}_{\mathcal{G}}$ extracted by f is representative of $\mathcal{G}$, we can construct a proxy objective $\max_{\mathcal{G}'} I(\mathbf{E}_{\mathcal{G}'}, \mathbf{E}_{\mathcal{G}})$, improving the consistency in the embedding space. In this work, we optimize this objective by aligning their representations, either optimizing a simplified distance metric or conducting distribution-aware alignment.

6 EXPERIMENT

In this section, we conduct a set of experiments to evaluate the benefits of the proposed auxiliary task in providing instance-level post-hoc explanations. Experiments are conducted on 5 datasets, and obtained explanations are evaluated w.r.t both faithfulness and consistency. Particularly, we aim to answer the following questions:

- **RQ1** Can the proposed framework perform strongly in identifying explanatory sub-structures for interpreting GNNs?
- **RQ2** Is the consistency problem severe in existing GNN explanation methods? Could the proposed embedding alignment improve GNN explainers over this criterion?
- **RQ3** Can our proposed strategy prevent spurious explanations and be more faithful to the target GNN model?

6.1 Experiment Settings

6.1.1 Datasets. We conduct experiments on five publicly available benchmark datasets for explainability of GNNs:

- BA-Shapes[37]: A node classification dataset with a Barabasi-Albert (BA) graph of 300 nodes as the base structure. 80 "house" motifs are randomly attached to the base graph. Nodes are labeled based on positions. Edges inside the corresponding motif are ground-truth for explaining those attached nodes.
- Tree-Grid [37]: A node classification dataset created by attaching 80 grid motifs to a single 8-layer balanced binary tree. For nodes in the attached motif, edges inside the same motif are used as ground-truth explanations.
- Infection [10]: A single network initialized with an ER random graph. 5% of nodes are labeled as infected, and other nodes are labeled based on their shortest distances to those infected ones. The graph is pre-processed following [10]. For each node, its shortest path is used as the ground-truth explanation.
- Mutag [37]: A graph classification dataset. Each graph corresponds to a molecule with nodes for atoms and edges for chemical bonds. Chemical groups are identified as explanations using prior domain knowledge. Following PGExplainer [18], chemical groups NH_2 and NO_2 are used as ground-truth explanations.
- Graph-SST5 [38]: A graph classification dataset constructed from text data, with labels from sentiment analysis. In this dataset, there is no ground-truth explanation provided, and heuristic metrics are usually adopted for evaluation.

6.1.2 Configurations. Following existing work [18], a three-layer GCN [15] is trained on 80% instances of each dataset as the target model. All explainers are trained using ADAM optimizer with weight decay set to $5e$-4. For GNNExplainer, learning rate is initialized to 0.01 with training epoch being 100. For PGExplainer, learning rate is initialized to 0.003 and training epoch is set as 30. Hyper-parameter λ, which controls the weight of $\mathcal{L}_{align}$, is tuned via grid search. Explanations are tested on all instances.

To evaluate the effectiveness of the proposed framework, we select a group of representative instance-level post-hoc GNN explanation methods as baselines: GRAD [18], ATT [18], GNNExplainer [37], PGExplaienr [18], Gem [17] and RG-Explainer [24]. Our proposed algorithms in Section 5.2 are implemented and incorporated into two representative GNN explanation frameworks, i.e., GNNExplainer [37] and PGExplainer [18].

6.1.3 Evaluation Metrics. To evaluate *faithfulness* of different methods, following [38], we adopt two metrics: (1) AUROC score on edge importance and (2) Fidelity of explanations. On benchmarks with oracle explanations available, we can compute the AUROC score on identified edges as the well-trained target GNN should follow those predefined explanations. On datasets without ground-truth explanations, we evaluate explanation quality with fidelity measurement following [38]. Concretely, we observe prediction changes by sequentially removing edges following assigned importance weight, and a faster performance drop represents stronger fidelity.

To evaluate *consistency* of explanations, we randomly run each method 5 times, and report average structural hamming distance (SHD) [27] among obtained explanations. A smaller SHD score indicates stronger consistency.

6.2 Explanation Faithfulness

To answer **RQ1**, we compare explanation methods in terms of AUROC score and explanation fidelity.

6.2.1 AUROC on Edges. In this subsection, AUROC scores of different methods are reported by comparing assigned edge importance weight with ground-truth explanations. For baseline methods GRAD, ATT, Gem, and RG-Explainer, their performances reported

Table 1: Explanation Faithfulness in terms of AUC on Edges

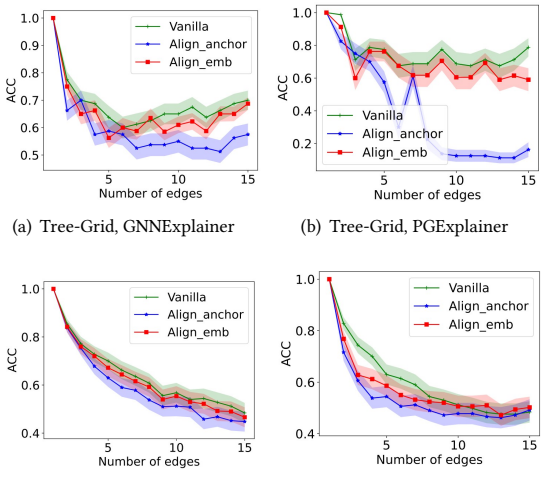
	BA-Shapes	Tree-Grid	Infection	Mutag
GRAD	88.2	61.2	74.0	78.3
ATT	81.5	66.7	–	76.5
Gem	97.1	–	–	83.4
RG-Explainer	98.5	92.7	–	87.3
GNNExplainer	$93.1_{\pm 1.8}$	$86.2_{\pm 2.2}$	$92.2_{\pm 1.1}$	$74.9_{\pm 1.9}$
+ Align_Emb	$95.3_{\pm 1.4}$	$91.2_{\pm 2.3}$	$93.0_{\pm 1.0}$	$76.3_{\pm 1.7}$
+ Align_Anchor	$97.1_{\pm 1.3}$	$92.4_{\pm 1.9}$	$\mathbf{93.1}_{\pm 0.8}$	$78.9_{\pm 1.6}$
PGExplainer	$96.9_{\pm 0.7}$	$92.7_{\pm 1.5}$	$89.6_{\pm 0.6}$	$83.7_{\pm 1.2}$
+ Align_Emb	$97.2_{\pm 0.7}$	$\mathbf{95.8}_{\pm 0.9}$	$90.5_{\pm 0.7}$	$92.8_{\pm 1.1}$
+ Align_Anchor	$\mathbf{98.7}_{\pm 0.5}$	$94.7_{\pm 1.2}$	$91.6_{\pm 0.6}$	$\mathbf{94.5}_{\pm 0.8}$

in their original papers are presented. GNNExplainer and PGExplainer are re-implemented, upon which two alignment strategies are instantiated and tested. Each experiment is randomly conducted 5 times, and we summarize the average performance in Table 1. A higher AUROC score indicates more accurate explanations. From the results, we can make the following observations:

- Across all four datasets, with both GNNExplainer or PGExplainer as the base method, incorporating embedding alignment can improve the quality of obtained explanations;
- In most cases, anchor-based alignment demonstrates stronger improvements, which shows the benefits of utilizing global distribution information;
- On more complex datasets like Mutag, the benefit of introducing embedding alignment is more significant, e.g., the performance of PGExplainer improves from 83.7% to 94.5% with Align Anchor. This result indicates that the problem of spurious explanations is severer with increased dataset complexity.

6.2.2 Explanation Fidelity. In addition to comparing to ground-truths explanations, we also evaluate the obtained explanations in terms of fidelity. Specifically, we sequentially remove edges from the graph by following importance weight learned by the explanation model and test the classification performance. Generally, the removal of really important edges would significantly degrade the classification performance. Thus, a faster performance drop represents stronger fidelity. We conduct experiments on Tree-Grid and Graph-SST5. Each experiment is conducted 3 times and we report results averaged across all instances on each dataset. PGExplainer and GNNExplainer are used as the backbone method. We plot the curves of prediction accuracy concerning the number of removed edges in Fig. 4. From the figure, we can observe that when proposed embedding alignment is incorporated, the classification accuracy from edge removal drops much faster, which shows that the proposed embedding alignment can help to identify more important edges used by GNN for classification, hence providing better explanations. Furthermore, anchor-based alignment is shown to be more effective, validating advantages in distribution-aware alignment.

From these two experiments, we can observe that embedding alignment can obtain explanations of better faithfulness and is flexible to be incorporated into various models such as GNNExplainer and PGExplainer, which answers RQ1.



(a) Tree-Grid, GNNExplainer (b) Tree-Grid, PGExplainer

(c) Graph-SS5, GNNExplainer (d) Graph-SS5, PGExplainer

Figure 4: Explanation Fidelity.

Table 2: Consistency of explanation in terms of average SHD distance across 5 rounds of random running on Tree-Grid.

	Top-K Edges					
Methods	1	2	3	4	5	6
GNNExplainer	0.86	1.85	2.48	3.14	3.77	4.39
+Align_Emb	0.77	1.23	1.28	0.96	1.81	2.72
+Align_Anchor	0.72	1.06	0.99	0.53	1.52	2.21
PGExplainer	0.74	1.23	0.76	0.46	0.78	1.38
+Align_Emb	0.11	0.15	**0.13**	**0.11**	0.24	0.19
+Align_Anchor	**0.07**	**0.12**	**0.13**	0.16	**0.21**	**0.13**

6.3 Explanation Consistency

One problem of spurious explanation is that, due to the randomness in initialization of the explainer, the explanation for the same instance given by a GNN explainer could be different for different runs, which violates the *consistency* of explanations. To test the severity of this problem and answer **RQ2**, we evaluate the proposed framework in terms of explanation consistency. We adopt GNNExplainer and PGExplainer as baselines. Specifically, SHD distance among explanatory edges with top-k importance weights identified each time is computed. Then, the results are averaged for all instances in the test set. Each experiment is conducted 5 times, and the average consistency on dataset Tree-Grid and Mutag are reported in Table 2 and Table 3, respectively. Larger distances indicate inconsistent explanations. From the table, we can observe that existing method following Equation 1 suffers from the consistency problem. For example, average SHD distance on top-6 edges is 4.39 for GNNExplainer. Introducing the auxiliary task of aligning embeddings can significantly improve explainers in terms of this criterion. After incorporating anchor-based alignment on dataset TreeGrid, SHD distance of top-6 edges drops from 4.39 to 2.21 for GNNExplainer and from 1.38 to 0.13 for PGExplainer, which validates effectiveness of our proposal in obtaining consistent explanations.

6.4 Ability in Avoiding Spurious Explanations

Existing graph explanation benchmarks are usually designed to be less ambiguous, containing only one oracle cause of labels, and identified explanatory substructures are evaluated via comparing

Table 3: Consistency of explanation in terms of average SHD distance across 5 rounds of random running on Mutag.

	Top-K Edges					
Methods	1	2	3	4	5	6
GNNExplainer	1.12	1.74	2.65	3.40	4.05	4.78
+Align_Emb	1.05	1.61	2.33	3.15	3.77	4.12
+Align_Anchor	1.06	1.59	2.17	3.06	3.54	3.95
PGExplainer	0.91	1.53	2.10	2.57	3.05	3.42
+Align_Emb	0.55	0.96	1.13	1.31	1.79	2.04
+Align_Anchor	**0.51**	**0.90**	**1.05**	**1.27**	**1.62**	**1.86**

Table 4: Performance on MixMotif. Two GNNs are trained with different γ. We check their performance in graph classification, then compare obtained explanations with the motif.

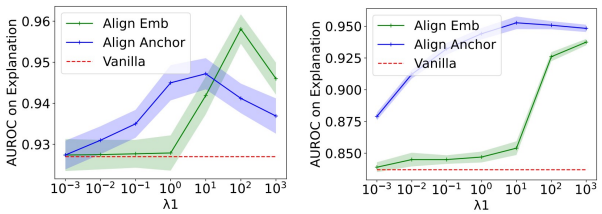
		γ in Training			
Classification		0		0.7	
γ in test	0	0.982		0.765	
	0.7	0.978		0.994	
Explanation		PGExplainer	+Align	PGExplainer	+Align
AUROC on Motif		0.711 (Higher is better)	**0.795**	0.748 (Lower is better)	**0.266**

with the ground-truth explanation. However, this result could be misleading, as faithfulness of explanation methods in more complex scenarios is left untested. Real-world datasets are usually rich in spurious patterns and a trained GNN could contain diverse biases, setting a tighter requirement on explanation methods. Thus, to evaluate if our framework can alleviate the spurious explanation issue and answer **RQ3**, we create a new graph-classification dataset: MixMotif, which enables us to train a biased GNN model, and test whether explanation methods can successfully expose this bias.

Specifically, inspired by [32], we design three types of base graphs, i.e., Tree, Ladder, and Wheel, and three types of motifs, i.e., Cycle, House, and Grid. With a mix ratio γ, motifs are preferably attached to base graphs. Labels are set as the type of motif. When γ is set to 0, each motif has the same probability of being attached to three base graphs. Thus, we consider the dataset with $\gamma = 0$ as clean or bias-free. We would expect GNN trained with $\gamma = 0$ to focus on the motif structure. However, when γ becomes larger, the spurious correlation between base graph and the label would exist, i.e., a GNN might utilize the base graph structure for motif classification instead of relying on the motif structure.

In this experiment, we set γ to 0 and 0.7 separately, and train GNN f_0 and $f_{0.7}$ for each setting. Two models are tested in graph classification performance. Then, explanation methods are applied to and fine-tuned on f_0. Following that, these explanation methods are applied to explain $f_{0.7}$ using found hyper-parameters. Results are summarized in Table 4.

From Table 4, we can observe that (1) f_0 achieves almost perfect graph classification performance during testing. This high accuracy indicates that it captures the genuine pattern, relying on motifs to make predictions. Looking at explanation results, it is shown that our proposal offers more faithful explanations, achieving higher AUROC on motifs. (2) $f_{0.7}$ fails to predict well with $\gamma = 0$, showing that there are biases in it and it no longer depends solely on the motif structure for prediction. Although ground-truth explanations are unknown in this case, a successful explanation should expose this bias. However, PGExplainer would produce similar explanations as the clean model, still highly in accord with motif structures. Instead, for explanations produced by embedding alignment, AUROC score would drop from 0.795 to 0.266, exposing the change in prediction rationales, hence able to expose biases. (3) In summary, our proposal can provide more faithful explanations for both clean and mixed settings, while PGExplainer would suffer from spurious explanations and fail to faithfully explain GNN's predictions, especially in the existence of biases.



(a) Tree-Grid (b) Mutag

Figure 5: Sensitivity of PGExplainer towards weight of embedding alignment loss.

6.5 Hyperparameter Sensitivity Analysis

In this part, we vary the hyper-parameter λ to test method's sensitivity towards its values. λ controls the weight of our proposed embedding alignment task. To keep simplicity, all other configurations are kept unchanged, and λ is varied as scale $[1e-3, 1e-2, 1e-1, 1, 10, 1e2, 1e3\}$. PGExplainer is adopted as the base method. Each experiment is conducted 3 times on Tree-Grid and Mutag. Averaged results are visualized in Figure 5. From the figure, we can observe that increasing λ has a positive effect at first, and the further increase would result in a performance drop. Besides, anchor-based alignment usually requires a smaller weight to show improvements.

7 CONCLUSION

In this work, we study a novel problem of obtaining faithful and consistent explanations for GNNs, which is largely neglected by existing MMI-based explanation framework. With close analysis on inference of GNNs, we propose a simple yet effective approach by aligning internal embeddings of the raw graph and its candidate explainable subgraph. The designed algorithms can be incorporated into existing methods with no effort. Experiments validate its effectiveness, and further theoretical analysis shows that it is more faithful in design.In the future, we will seek more robust explanations. Increased robustness indicates stronger generality, and could provide better class-level interpretation at the same time.

8 ACKNOWLEDGEMENT

This material is based upon work supported by, or in part by, the National Science Foundation under grants number IIS-1707548 and IIS-1909702, the Army Research Office under grant number W911NF21-1-0198, and DHS CINA under grant number E205949D. The findings and conclusions in this paper do not necessarily reflect the view of the funding agency.

REFERENCES

[1] James Atwood and Don Towsley. 2016. Diffusion-convolutional neural networks. In *Advances in neural information processing systems*. 1993–2001.

[2] Federico Baldassarre and Hossein Azizpour. 2019. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686* (2019).

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).

[4] Hryhorii Chereda, A. Bleckmann, F. Kramer, A. Leha, and T. Beißbarth. 2019. Utilizing Molecular Network Information via Graph Convolutional Neural Networks to Predict Metastatic Event in Breast Cancer. *Studies in health technology and informatics* 267 (2019), 181–186.

[5] Enyan Dai, Wei Jin, Hui Liu, and Suhang Wang. 2022. Towards Robust Graph Neural Networks for Noisy Graphs with Sparse Labels. *arXiv preprint arXiv:2201.00232* (2022).

[6] Enyan Dai and Suhang Wang. 2021. Towards Self-Explainable Graph Neural Network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 302–311.

[7] Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Junjie Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. 2022. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. *arXiv preprint arXiv:2204.08570* (2022).

[8] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*. 2224–2232.

[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *kdd*, Vol. 96. 226–231.

[10] Lukas Faber, Amin K. Moghaddam, and Roger Wattenhofer. 2021. When Comparing to Ground Truth is Wrong: On Evaluating GNN Explanation Methods. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 332–341.

[11] Wenqi Fan, Y. Ma, Qing Li, Yuan He, Y. Zhao, Jiliang Tang, and D. Yin. 2019. Graph Neural Networks for Social Recommendation. *The World Wide Web Conference* (2019).

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *ICML*.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[14] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, Dawei Yin, and Yi Chang. 2020. Graphlime: Local interpretable model explanations for graph neural networks. *arXiv preprint arXiv:2001.06216* (2020).

[15] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[16] Shuai Lin, Chen Liu, Pan Zhou, Zi-Yuan Hu, Shuojia Wang, Ruihui Zhao, Yefeng Zheng, Liang Lin, Eric Xing, and Xiaodan Liang. 2022. Prototypical graph contrastive learning. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[17] Wanyu Lin, Hao Lan, and Baochun Li. 2021. Generative causal explanations for graph neural networks. In *International Conference on Machine Learning*. PMLR, 6666–6679.

[18] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. *Advances in neural information processing systems* 33 (2020), 19620–19631.

[19] Elman Mansimov, O. Mahmood, Seokho Kang, and Kyunghyun Cho. 2019. Molecular Geometry Prediction using a Deep Generative Graph Neural Network. *Scientific Reports* 9 (2019).

[20] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. 2022. From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI. *arXiv preprint arXiv:2201.08164* (2022).

[21] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. 2019. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10772–10781.

[22] Jiahua Rao, Shuangjia Zheng, and Yuedong Yang. 2021. Quantitative Evaluation of Explainable Graph Neural Networks for Molecular Property Prediction. *arXiv preprint arXiv:2107.04119* (2021).

[23] Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T Schütt, Klaus-Robert Müller, and Grégoire Montavon. 2020. Higher-order explanations of graph neural networks via relevant walks. *arXiv preprint arXiv:2006.03589* (2020).

[24] Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. 2021. Reinforcement Learning Enhanced Explainer for Graph Neural Networks. *Advances in Neural Information Processing Systems* 34 (2021).

[25] Daniil Sorokin and Iryna Gurevych. 2018. Modeling Semantics with Gated Graph Neural Networks for Knowledge Base Question Answering. *ArXiv* abs/1808.04126 (2018).

[26] S. Tang, Bo Li, and Haijun Yu. 2019. ChebNet: Efficient and Stable Constructions of Deep Neural Networks with Rectified Power Units using Chebyshev Approximations. *ArXiv* abs/1911.05467 (2019).

[27] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine learning* 65, 1 (2006), 31–78.

[28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[29] Minh N Vu and My T Thai. 2020. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *arXiv preprint arXiv:2010.05788* (2020).

[30] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1001–1010.

[31] Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat-seng Chua. 2020. Causal Screening to Interpret Graph Neural Networks. (2020).

[32] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. 2022. Discovering Invariant Rationales for Graph Neural Networks. *arXiv preprint arXiv:2201.12872* (2022).

[33] Teng Xiao, Zhengyu Chen, Zhimeng Guo, Zeyang Zhuang, and Suhang Wang. 2022. Decoupled Self-supervised Learning for Non-Homophilous Graphs. *arXiv e-prints* (2022), arXiv–2206.

[34] Teng Xiao, Zhengyu Chen, Donglin Wang, and Suhang Wang. 2021. Learning how to propagate messages in graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1894–1903.

[35] Junjie Xu, Enyan Dai, Xiang Zhang, and Suhang Wang. 2022. HP-GMN: Graph Memory Networks for Heterophilous Graphs. *arXiv preprint arXiv:2210.08195* (2022).

[36] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[37] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems* 32 (2019), 9240.

[38] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2020. Explainability in graph neural networks: A taxonomic survey. *arXiv preprint arXiv:2012.15445* (2020).

[39] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. In *International Conference on Machine Learning*. PMLR, 12241–12252.

[40] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in neural information processing systems* 31 (2018).

[41] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. 2021. ProtGNN: Towards Self-Explaining Graph Neural Networks. *arXiv preprint arXiv:2112.00911* (2021).

[42] Tianxiang Zhao, Xianfeng Tang, Xiang Zhang, and Suhang Wang. 2020. Semi-Supervised Graph-to-Graph Translation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1863–1872.

[43] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. 2021. GraphSMOTE: Imbalanced Node Classification on Graphs with Graph Neural Networks. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining*.

[44] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. 2022. Exploring edge disentanglement for node classification. In *Proceedings of the ACM Web Conference 2022*. 1028–1036.