Faithful and Consistent Graph Neural Network Explanations with Rationale Alignment

TIANXIANG ZHAO, The Pennsylvania State University, USA
DONGSHENG LUO, Florida International University, USA
XIANG ZHANG and SUHANG WANG, The Pennsylvania State University, USA

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, such as 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 failing 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 for 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 new explanation framework with an auxiliary alignment loss, which is theoretically proven to be optimizing a more faithful explanation objective intrinsically. Concretely for this alignment loss, a set of different perspectives are explored: anchor-based alignment, distributional alignment based on Gaussian mixture models, mutual-information-based alignment, and so on. A comprehensive study is conducted both on the effectiveness of this new framework in terms of explanation faithfulness/consistency and on the advantages of these variants. For our codes, please refer to the following URL link: https://github.com/TianxiangZhao/GraphNNExplanation

CCS Concepts: • **Computing methodologies** → **Neural networks**; *Statistical relational learning;*

Additional Key Words and Phrases: Graph neural network, explainable AI, machine learning

ACM Reference format:
Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. 2023. Faithful and Consistent Graph Neural Network Explanations with Rationale Alignment. *ACM Trans. Intell. Syst. Technol.* 14, 5, Article 92 (October 2023), 23 pages.
https://doi.org/10.1145/3616542

This material is based upon work supported by, or in part by, the National Science Foundation under grant numbers IIS-1707548 and IIS-1909702, the Army Research Office under grant number W911NF21-1-0198, and DHS CINA under grant number E205949D.

Authors' addresses: T. Zhao, X. Zhang, and S. Wang, College of Information Sciences and Technology, The Pennsylvania State University, State College, PA16801, USA; emails: {tkz5084, xzz89, szw494}@psu.edu; D. Luo, Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA; email: dluo@fiu.edu.

2157-6904/2023/10-ART92 $15.00
https://doi.org/10.1145/3616542

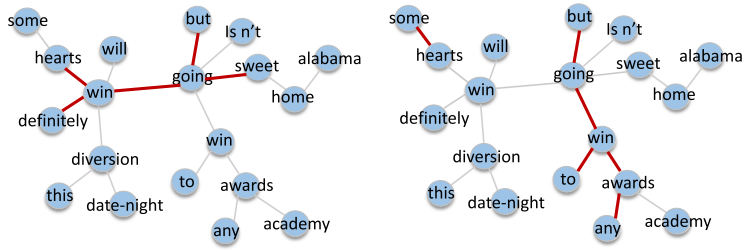
Graph-structured data is ubiquitous in the real world, such as social networks [1–3], molecular structures [4, 5], and knowledge graphs [6, 7]. With the growing interest in learning from graphs, **graph neural networks (GNNs)** are receiving more and more attention over the years. Generally, GNNs adopt message-passing mechanisms, which recursively propagate and fuse messages from neighbor nodes on the graphs. Hence, the learned node representation captures both node attributes and neighborhood information, which facilitates various downstream tasks such as node classification [8–11], graph classification [12], and link prediction [13–15].

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 model characteristics such as if the model is working as desired. Second, it increases the models' transparency to enable trusted applications in decision-critical fields sensitive to fairness, privacy, and safety challenges [16–18]. Thus, studying the explainability of GNNs is attracting increasing attention and many efforts have been taken [19–21].

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 [19, 22, 23]. For example, in search of important substructures that predictions rely upon, GNNExplainer learns an importance matrix on node attributes and edges via perturbation [19]. 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 [20]. SubgraphX constraints explanations into connected subgraphs and conducts Monte Carlo tree search based on Shapley value [24]. These methods can be summarized in a label preserving framework, i.e., the candidate explanation is formed as a masked version of the original graph and is identified as the minimal discriminative substructure.

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 Figure 1, where a GNN is trained on Graph-SST5 [25] 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 [19] 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 [26]. Consequently, explanations provided by 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 [19, 20, 24]. However, our analysis demonstrates that the label information is insufficient to filter out spurious explanations, leading to inconsistent and unfaithful explanations.



(a) Run 1 (b) Run 2

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

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. Specifically, to evaluate the semantic similarity between two inputs and incorporate the alignment objective into explanation, we design and compare strategies with various design choices to measure similarity in the embedding space. These strategies enable the alignment between candidate explanations and original inputs and are flexible to be incorporated into various existing GNN explanation methods. Particularly, aside from directly using Euclidean distance, we further propose three distribution-aware strategies. The first one identifies a set of anchoring embeddings and utilizes relative distances against them. The second one assumes a Gaussian mixture model and captures the distribution using the probability of falling into each Gaussian center. The third one learns a deep neural network to estimate mutual information between two inputs, which takes a data-driven approach with little reliance upon prior domain knowledge. Further analysis shows that the proposed method is in fact optimizing a new explanation framework, which is more faithful in design. Our main contributions are:

— 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. We implement a set of variants with different alignment strategies, which is flexible to be incorporated to various GNN explanation models. We further conduct a 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.

1 RELATED WORK

In this section, we review related works, including graph neural networks and interpretability of GNNs.

1.1 Graph Neural Networks

GNNs are developing rapidly in recent years, with the increasing need for learning on relational data structures [1, 27–30]. Generally, existing GNNs can be categorized into two categories, i.e., spectral-based approaches [31–33] based on graph signal processing theory and spatial-based approaches [34–36] 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 [37]. 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 [10, 13, 28, 38]. Explorations are made by disentangling the propagation process [39–41] or utilizing external prototypes [42, 43]. Research has also been conducted on the expressive power [44, 45] and potential biases introduced by different kernels [46, 47] for the design of more effective GNNs. 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.

1.2 GNN Interpretation Methods

Recently, some efforts have been taken to interpret GNN models and provide explanations for their predictions [48]. Based on the granularity, existing methods can be generally grouped into two categories: (1) instance-level explanation [19], which provides explanations on the prediction for each instance by identifying important substructures; and (2) model-level explanation [49, 50], 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 [50, 51], where the GNN can simultaneously give prediction and explanations on the prediction; and (2) post hoc explanations [19, 20, 24], 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 [19], 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 Reference [52].

A variety of strategies for post hoc instance-level explanations have been explored in the literature, including utilizing signals from gradients-based [49, 53], perturbed predictions-based [19, 20, 24, 54], and decomposition-based [49, 55]. 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 [25]. The identified important substructure is used as an explanation for the prediction. For example, GNNExplainer [19] 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 [20] 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 [24] expects explanations to be in the form of sub-graphs instead of bag-of-edges and employs **Monte Carlo Tree Search (MCTS)** to find connected subgraphs that preserve predictions measured by the Shapley value. To promote faithfulness in identified explanations, some works introduced terminologies from the causality analysis domain, via estimating the individual causal effect of each edge [56] or designing interventions to prevent the discovery of spurious correlations [57]. Reference [58] connects the idea of identifying minimally predictive parts in explanation with the principle of information bottleneck [59] and designs an end-to-end optimization framework for GNN explanation.

Despite the aforementioned 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. However, by aligning intermediate embeddings

in GNNs, our method alleviates the effects of spurious correlations on interpreting GNNs, leading to faithful and consistent explanations.

1.3 Graph Contrastive Learning

In recent years, **contrastive learning (CL)** has garnered significant attention, as it mitigates the need for manual annotations via unsupervised pretext tasks [60–62]. In a typical CL framework, the model is trained in a pairwise manner, promoting attraction between positive sample pairs and repulsion between negative sample pairs [63, 64].

CL techniques have recently been extended to the graph domain, constructing multiple graph views and MMI among semantically similar instances [62, 65–69]. Existing methods primarily vary on graph augmentation techniques and contrastive pretext tasks. Graph augmentations commonly involve node-level attribute masking or perturbation [70–72], edge dropping or rewiring [73, 74], and graph diffusions[75, 76]. Contrastive pretext tasks can be categorized into two branches, *same-scale* contrasts between embeddings of node (graph) pairs [75, 77, 78] and *cross-scale* contrasts between global graph embeddings and local node representations [71, 72]. Recent works have also explored dynamic contrastive objectives [79, 80] via learning the augmentation strategies adaptively.

2 PRELIMINARY

2.1 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 $\odot$ denotes 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:

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 $\mathcal{G}$. The obtained explanation $\mathcal{G}'$ is depicted by importance mask $\mathbf{M}_F$ on node attributes and importance mask $\mathbf{M}_A$ on adjacency matrix.

2.2 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. Generally, they can be summarized as 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 [19], 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 [20]. 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 [19] 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 optimization and searching strategies.

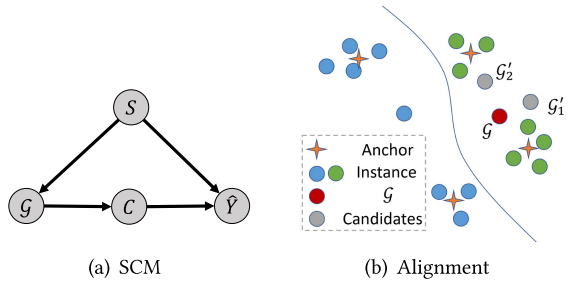
Different aspects regarding the quality of explanations can be evaluated [26]. Among them, two most important criteria are **faithfulness** and **consistency**. Faithfulness measures the descriptive accuracy of explanations, indicating how truthful they are compared to behaviors of the target model. 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. Due to the complex graph structure, the prediction alone as a guide could result in spurious explanations. A detailed analysis will be provided in the next section.

3 ANALYZE SPURIOUS EXPLANATIONS

With "spurious explanations," we refer to those explanations lying 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 will examine the 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.

- $\mathcal{G} \rightarrow C \rightarrow \hat{Y}$: This path presents the inference of target GNN model, i.e., critical patterns C that are informative and discriminative for the prediction $\hat{Y}$ 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 the prediction of target GNN models. A 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 the causal structures, we can observe that spurious explanations may arise as a result of failure in recovering the 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 GNNs



(a) SCM (b) Alignment

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

$f(\cdot)$ that are unstable or non-robust towards noises would further amplify this problem, as the prediction is unreliable.

To further understand the issue, we build the correspondence from SEM in Figure 2(a) to the inference process of GNN f. Specifically, we first decompose $f()$ as a feature extractor $f_{ext}()$ and a classifier $f_{cls}()$. Then, 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.

4 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 four 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.

4.1 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 3.

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 3, 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 lth 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:

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

where Message_l and Update_l are the message function and update function at lth 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 GNNs for graph classification. We propose to align embedding $\mathbf{h}_v^{l+1}$ at each layer, which contains both node and local neighborhood information.

4.2 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 the 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 [81] as the clustering algorithm and tune its hyperparameters to get around 20 groups.
— At lth 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 kth element calculated as $\mathbf{s}_v^{l+1,k} = \|\mathbf{h}_v^{l+1} - \mathbf{h}_v^{l+1,k}\|_2$.

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

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

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. Figure 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.

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

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

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

4.3 Direct Alignment

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 $\sum_l \sum_{v\in\mathcal{V}} \|\mathbf{h}_v^l - \mathbf{h}_v^{'l}\|_2^2$. 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 EXTENDED METHODOLOGY

In this section, we further examine more design choices for the strategy of alignment to obtain faithful and consistent explanations. Instead of using heuristic approaches, we explore two new directions: (1) statistically sound distance measurements based on the Gaussian mixture model, (2) fully utilizing the power of deep neural networks to capture distributions in the latent embedding space. Details of these two alignment strategies will be introduced below.

5.1 Gaussian-mixture-based Alignment

In this strategy, we model the latent embeddings of nodes (or graphs) using a mixture of Gaussian distributions, with representative node/graph embeddings as anchors (Gaussian centers). The produced embedding of each input can be compared with those prototypical anchors, and the semantic information of inputs taken by the target model would be encoded by relative distances from them.

Concretely, we first obtain prototypical representations, annotated as $\{\mathbf{h}^{l,k}\}_{k=1}^{K}$, by running the clustering algorithm on collected embeddings $\{\{\mathbf{h}^{l}_{v,i}\}_{v\in\mathcal{V}'_i}\}_{i=1}^{m}$ from graphs $\{\mathcal{G}_i\}_{i=1}^{m}$, in the same strategy as introduced in Section 4.2. Clustering algorithm DBSCAN [81] is adopted, and we tune its hyperparameters to get around 20 groups.

Next, the probability of encoded representation $\mathbf{h}^{l}$ falling into each prototypical Gaussian centers $\{\mathbf{h}^{l,k}\}_{k=1}^{K}$ can be computed as:

$$p_v^{l,k} = \frac{\exp(-\|\mathbf{h}_v^l - \mathbf{h}^{l,k}\|_2^2/2\sigma^2)}{\sum_{j=1}^{K}\exp(-\|\mathbf{h}_v^l - \mathbf{h}^{l,k}\|_2^2/2\sigma^2)}. \tag{7}$$

This distribution probability can serve as a natural tool for depicting the semantics of the input graph learned by the GNN model. Consequently, the distance between $\mathbf{h}_v^{'l}$ and $\mathbf{h}_v^l$ can be directly measured as the KL-divergence w.r.t. this distribution probability:

$$d(\mathbf{p}_v^{'l}, \mathbf{p}_v^l) = \sum_{k\in[1,\ldots,K]} p_v^{'l,k}\cdot\log\left(\frac{p_v^{'l,k}}{p_v^{l,k}}\right), \tag{8}$$

where $\mathbf{p}_v^{'l}\in\mathbb{R}^K$ denotes the distribution probability of candidate explanation embedding, $\mathbf{h}_v^{'l}$. Using this strategy, the alignment loss between original graph and the candidate explanation is computed as:

$$\mathcal{L}_{align}\Big(f(\mathcal{G}), f(\mathcal{G}')\Big) = \sum_{l}\sum_{v\in\mathcal{V}'} d\left(\mathbf{p}_v^{'l}, \mathbf{p}_v^l\right), \tag{9}$$

which can be incorporated into the revised explanation framework proposed in Equation (6).

Comparison Compared with the alignment loss based on relative distances against anchors in Equation (5), this new objective offers a better strategy in taking distribution into consideration. Specifically, we can show the following two advantages:

— In obtaining the distribution-aware representation of each instance, this variant uses a Gaussian distance kernel (Equation (7)) while the other one uses Euclidean distance in Section 4.2, which may amplify the influence of distant anchors. We can prove this by examining the gradient of changes in representation w.r.t. GNN embeddings $\mathbf{h}_v$. In the lth layer at dimension k, the gradient of the previous variant can be computed as:

$$\frac{\partial s_v^{l,k}}{\partial \mathbf{h}_v^l} = 2\cdot\left(\mathbf{h}_v^l - \mathbf{h}_v^{l,k}\right). \tag{10}$$

However, the gradient of this variant is:

$$\frac{\partial p_v^{l,k}}{\partial \mathbf{h}_v^l} \approx -\frac{\exp(-\|\mathbf{h}_v^l - \mathbf{h}^{l,k}\|_2^2/2\sigma^2)\cdot(\mathbf{h}_v^l - \mathbf{h}_v^{l,k})}{\sigma^2\cdot\sum_{j=1}^{K}\exp(-\|\mathbf{h}_v^l - \mathbf{h}^{l,k}\|_2^2/2\sigma^2)}. \tag{11}$$

It is easy to see that for the previous variant, the magnitude of gradient would grow linearly w.r.t. distances towards corresponding anchors. For this variant, however, the term $\frac{\exp(-\|\mathbf{h}_v^l-\mathbf{h}^{l,k}\|_2^2/2\sigma^2)}{\sigma^2\cdot\sum_{j=1}^{K}\exp(-\|\mathbf{h}_v^l-\mathbf{h}^{l,k}\|_2^2/2\sigma^2)}$ would down-weight the importance of those distant anchors while up-weight the importance of similar anchors, which is more desired in obtaining distribution-aware representations.

— In computing the distance between representations of two inputs, this variant adopts the KL divergence as in Equation (8), which would be scale-agnostic compared to the other one directly using Euclidean distance as in Equation (5). Again, we can show the gradient of

alignment loss towards obtained embeddings that encode distribution information. It can be observed that for the previous variant:

$$\frac{\partial d(\mathbf{s}_v^l, \mathbf{s}_v^{'l})}{\partial s_v^{'l,k}} = 2 \cdot \left(s_v^{l,k} - s_v^{'l,k}\right). \tag{12}$$

For this variant based on Gaussian mixture model, the gradient can be computed as:

$$\frac{\partial d(\mathbf{p}_v^l, \mathbf{p}_v^{'l})}{\partial p_v^{'l,k}} = 1 + \log\left(\frac{p_v^{'l,k}}{p_v^{l,k}}\right). \tag{13}$$

It can be observed that the previous strategy focuses on representation dimensions with a large absolute difference, while would be sensitive towards the scale of each dimension. However, this strategy uses the summation between the logarithm of relative difference with a constant, which is scale-agnostic towards each dimension.

5.2 MI-based Alignment

In this strategy, we further consider the utilization of deep models to capture the distribution and estimate the semantic similarity of two inputs and incorporate it into the alignment loss for discovering faithful and consistent explanations. Specifically, we train a deep model to estimate the **mutual information (MI)** between two input graphs and use its prediction as a measurement of alignment between original graph and its candidate explanation. This strategy circumvents the reliance on heuristic strategies and is purely data-driven, which can be learned in an end-to-end manner.

To learn the mutual information estimator, we adopt a neural network and train it to be a Jensen-Shannon MI estimator [82]. Concretely, we train this JSD-based estimator on top of intermediate embeddings with the learning objective as follows, which offers better stability in optimization:

$$\begin{aligned}\min_{g_{mi}} \mathcal{L}_{mi} = \mathbb{E}_{\mathcal{G}\in\{\mathcal{G}_i\}_{i=1}^m} \mathbb{E}_{v\in\mathcal{G}} \mathbb{E}_l \Big[\mathbb{E}_{\mathbf{h}_v^{l,+}} sp\left(-T^l\left(\mathbf{h}_v^l, \mathbf{h}_v^{l,+}\right)\right) \\ + \mathbb{E}_{\mathbf{h}_v^{l,-}} sp\left(T^l\left(\mathbf{h}_v^l, \mathbf{h}_v^{l,-}\right)\right) \Big],\end{aligned} \tag{14}$$

where $\mathbb{E}$ denotes expectation. In this equation, $T^l(\cdot)$ is a compatibility estimation function in the lth layer, and we denote $\{T^l(\cdot)\}_l$ as the MI estimator g_{mi}. Activation function $sp(\cdot)$ is the *softplus* function, and $\mathbf{h}_v^+$ represents the embedding of augmented node v that is a **positive pair** of v in the original graph. On the contrary, $\mathbf{h}_v^-$ denotes the embedding of augmented node v that is a **negative pair** of original input. A positive pair is obtained through randomly dropping intermediate neurons, corresponding to masking out a ratio of original input, and a negative pair is obtained as embeddings of different nodes. This objective can guide g_{mi} to capture the correlation or similarity between two input graphs encoded by the target model. Note that to further improve the learning of MI estimator, more advanced graph augmentation techniques can be incorporated for obtaining positive/negative pairs, following recent progresses in graph contrastive learning [65, 66]. In this work, we stick to the strategy adopted in Reference [82] for its popularity and leave that for future explorations. With this MI estimator learned, alignment loss between $\mathcal{G}$ and $\mathcal{G}^{'}$ can be readily computed:

$$\mathcal{L}_{align}\Big(f(\mathcal{G}), f(\mathcal{G}')\Big) = \sum_l \sum_{v\in\mathcal{V}'} sp(-T^l(\mathbf{h}_v^{'l}, \mathbf{h}_v^l)), \tag{15}$$

which can be incorporated into the revised explanation framework proposed in Equation (6).

In this strategy, we design a data-driven approach by capturing the mutual information between two inputs, which circumvents the potential biases of using human-crafted heuristics.

6 THEORETICAL ANALYSIS

With those alignment strategies and our new explanation framework introduced, next, we want to look deeper and provide theoretical justifications for the proposed new loss function in Equation (6). In this section, we first propose a new explanation objective to prevent spurious explanations based Section 3. Then, we theoretically show that it squares with our loss function with mild relaxations.

6.1 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 maximizes S_1 can be formalized as:

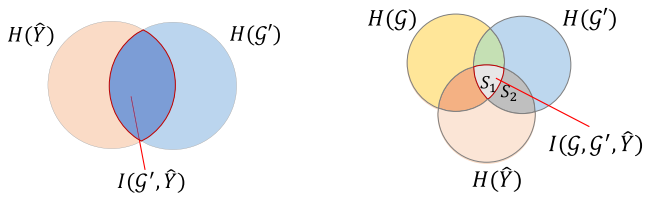
$$\begin{aligned} &\min_{\mathcal{G}'} -I(\mathcal{G},\mathcal{G}',\hat{Y}), \\ &\text{s.t.} \quad \mathcal{G}' \sim \mathcal{P}(\mathcal{G},\mathbf{M}_A,\mathbf{M}_F) \quad \mathcal{R}(\mathbf{M}_F,\mathbf{M}_A) \le c. \end{aligned} \tag{16}$$

6.2 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:

$$\begin{aligned} I(\mathcal{G},\mathcal{G}',\hat{Y}) &= \sum_{y\sim\hat{Y}}\sum_{\mathcal{G}}\sum_{\mathcal{G}'} P(\mathcal{G},\mathcal{G}',y)\cdot\log\frac{P(\mathcal{G}',y)P(\mathcal{G},\mathcal{G}')P(\mathcal{G},y)}{P(\mathcal{G},\mathcal{G}',y)P(\mathcal{G})P(\mathcal{G}')P(y)} \\ &= \sum_{y\sim\hat{Y}}\sum_{\mathcal{G}}\sum_{\mathcal{G}'} P(\mathcal{G},\mathcal{G}',y)\cdot\log\left[\frac{P(\mathcal{G}',y)}{P(\mathcal{G}')P(y)}\cdot\frac{P(\mathcal{G},\mathcal{G}')}{P(\mathcal{G})P(\mathcal{G}')}\cdot\frac{P(\mathcal{G},y)}{P(\mathcal{G},y|\mathcal{G}')}\right] \\ &= \sum_{y\sim\hat{Y}}\sum_{\mathcal{G}'} P(\mathcal{G}',y)\cdot\log\frac{P(\mathcal{G}',y)}{P(\mathcal{G}')P(y)} + \sum_{\mathcal{G}}\sum_{\mathcal{G}'} P(\mathcal{G},\mathcal{G}')\cdot\log\frac{P(\mathcal{G},\mathcal{G}')}{P(\mathcal{G})P(\mathcal{G}')} \\ &\quad - \sum_{\mathcal{G}'}\sum_{y\sim\hat{Y}}\sum_{\mathcal{G}} P(\mathcal{G},y,\mathcal{G}')\cdot\log\frac{P(\mathcal{G},y,\mathcal{G}')}{P(\mathcal{G},y)P(\mathcal{G}')}\Big] \\ &= I(\mathcal{G}',\hat{Y}) + I(\mathcal{G},\mathcal{G}') - \sum_{y\sim\hat{Y}}\sum_{\mathcal{G}} P(\mathcal{G},y)\sum_{\mathcal{G}'} P(\mathcal{G}'|\mathcal{G},y)\cdot\log P(\mathcal{G}'|\mathcal{G},y) \\ &\quad + \sum_{\mathcal{G}'}\sum_{y\sim\hat{Y}}\sum_{\mathcal{G}} P(\mathcal{G},y,\mathcal{G}')\cdot\log P(\mathcal{G}') \\ &= I(\mathcal{G}',\hat{Y}) + I(\mathcal{G},\mathcal{G}') + H(\mathcal{G}'|\mathcal{G},\hat{Y}) - H(\mathcal{G}'). \end{aligned}$$

Since both $H(\mathcal{G}'|\mathcal{G},\hat{Y})$ and $H(\mathcal{G}')$ depict 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 $\max_{\mathcal{G}'} I(\hat{Y},\mathcal{G}') + I(\mathcal{G}',\mathcal{G})$.



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

Fig. 3. Illustration of our proposed new objective.

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 Equation (1). Following Reference [19], 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.

7 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 five datasets, and obtained explanations are evaluated with respect to both faithfulness and consistency. Particularly, we aim to answer the following questions:

- **RQ1:** Can the proposed framework perform strongly in identifying explanatory substructures 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 strategies prevent spurious explanations and be more faithful to the target GNN model?

7.1 Experiment Settings

7.1.1 Datasets. We conduct experiments on five publicly available benchmark datasets for explainability of GNNs. The key statistics of the datasets are summarized in Table 1.

- BA-Shapes[19]: A node classification dataset with a **Barabasi-Albert (BA)** graph of 300 nodes as the base structure. Eighty "house" motifs are randomly attached to the base graph. Nodes in the base graph are labeled as 0 and those in the motifs are labeled based on positions. Explanations are conducted on those attached nodes, with edges inside the corresponding motif as ground-truth.
- Tree-grid [19]: A node classification dataset created by attaching 80 grid motifs to a single 8-layer balanced binary tree. Nodes in the base graph are labeled as 0, and those in the motifs are labeled as 1. Edges inside the same motif are used as ground-truth explanations for nodes from class 1.
- Infection [83]: 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. Labels larger than 4 are clipped. Following Reference [83], infected nodes and nodes with multiple shortest paths are neglected. For each node, its shortest path is used as the ground-truth explanation.

Table 1. Statistics of Datasets

	BA-shapes	Tree-grid	Infection	Mutag	SST-5
Level	Node	Node	Node	Graph	Graph
Graphs	1	1	1	4,337	11,855
Avg.Nodes	700	1,231	1,000	30.3	19.8
Avg.Edges	4,110	3,410	4,001	61.5	18.8
Classes	4	2	5	2	5

— Mutag [19]: A graph classification dataset. Each graph corresponds to a molecule with nodes for atoms and edges for chemical bonds. Molecules are labeled with consideration of their chemical properties, and discriminative chemical groups are identified using prior domain knowledge. Following PGExplainer [20], chemical groups NH_2 and NO_2 are used as ground-truth explanations.
— Graph-SST5 [25]: A graph classification dataset constructed from text, with labels from sentiment analysis. Each node represents a word, and edges denote word dependencies. In this dataset, there is no ground-truth explanation provided, and heuristic metrics are usually adopted for evaluation.

7.1.2 Baselines. To evaluate the effectiveness of the proposed framework, we select a group of representative and state-of-the-art instance-level post hoc GNN explanation methods as baselines. The details are given as follows:

— GRAD [20]: A gradient-based method, which assigns importance weights to edges by computing gradients of GNN's prediction w.r.t. the adjacency matrix.
— ATT [20]: It utilizes average attention weights inside self-attention layers to distinguish important edges.
— GNNExplainer [19]: A perturbation-based method that learns an importance matrix separately for every instance.
— PGExplainer [20]: A parameterized explainer that learns a GNN to predict important edges for each graph and is trained via testing different perturbations;
— Gem [56]: Similar to PGExplainer but from the causal view, based on the estimated individual causal effect.
— RG-Explainer [54]: A **reinforcement learning (RL)**-enhanced explainer for GNN, which constructs $\mathcal{G}'$ by sequentially adding nodes with an RL agent.

Our proposed algorithms in Section 4.2 are implemented and incorporated into two representative GNN explanation frameworks, i.e., GNNExplainer [19] and PGExplainer [20].

7.1.3 Configurations. Following existing work [20], a three-layer GCN [8] is trained on each dataset as the target model, with the train/validation/test data split as 8:1:1. The latent dimension is set to 64 across all datasets, and we use Relu as activation function. For graph classification, we concatenate the outputs of global max pooling and global mean pooling as the graph representation. 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. Hyperparameter λ, which controls the weight of $\mathcal{L}_{align}$, is tuned via grid search within the range $[0.1, 10]$ for different datasets separately. Explanations are tested on all instances.

7.1.4 Evaluation Metrics. To evaluate *faithfulness* of different methods, following Reference [25], we adopt two metrics: (1) AUROC score on edge importance and (2) Fidelity of explanations.

Table 2. 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	$97.1_{\pm 1.3}$	$92.4_{\pm 1.9}$	$93.1_{\pm 0.8}$	$78.9_{\pm 1.6}$
+ Align_MI	$97.4_{\pm 1.6}$	$92.2_{\pm 2.5}$	$93.2_{\pm 1.0}$	$78.2_{\pm 1.8}$
+ Align_Gaus	$96.7_{\pm 1.3}$	$92.5_{\pm 1.9}$	$93.1_{\pm 0.9}$	$79.3_{\pm 1.5}$
PGExplainer	$96.9_{\pm 0.7}$	$92.7_{\pm 1.5}$	$89.6_{\pm 0.6}$	$83.7_{\pm 1.2}$
+ Align_Emb	$97.2_{\pm 0.7}$	$95.8_{\pm 0.9}$	$90.5_{\pm 0.7}$	$92.8_{\pm 1.1}$
+ Align_Anchor	$98.7_{\pm 0.5}$	$94.7_{\pm 1.2}$	$91.6_{\pm 0.6}$	$94.5_{\pm 0.8}$
+ Align_MI	$\mathbf{99.3}_{\pm 0.8}$	$96.2_{\pm 1.3}$	$92.0_{\pm 0.4}$	$94.3_{\pm 1.1}$
+ Align_Gaus	$99.2_{\pm 0.3}$	$\mathbf{96.4}_{\pm 1.1}$	$\mathbf{92.5}_{\pm 0.8}$	$\mathbf{95.9}_{\pm 1.2}$

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 Reference [25]. Concretely, we observe prediction changes by sequentially removing edges following assigned importance weight, and a faster performance drop represents stronger fidelity.

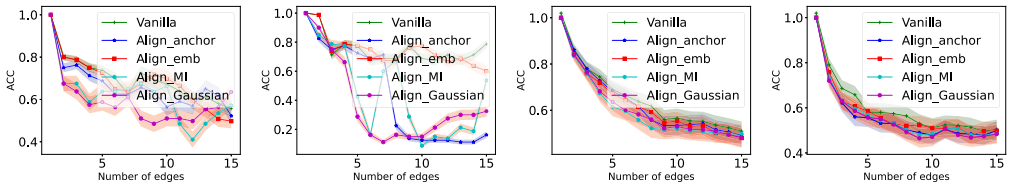
To evaluate *consistency* of explanations, we randomly run each method five times and report average **structural Hamming distance (SHD)** [84] among obtained explanations. A smaller SHD score indicates stronger consistency.

7.2 Explanation Faithfulness

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

7.2.1 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 in their original papers are presented. GNNExplainer and PGExplainer are re-implemented, upon which four alignment strategies are instantiated and tested. Each experiment is conducted five times, and we summarize the average performance in Table 2. 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;
— Among proposed alignment strategies, those distribution-aware approaches, particularly the variant based on Gaussian mixture models, achieve the best performance. In most cases, the variant utilizing latent Gaussian distribution demonstrates stronger improvements, showing the best results on three out of four datasets;
— 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 95.9% with Align_Gaus. This result also indicates that spurious explanations are severer with increased dataset complexity.



(a) Tree-Grid, GNNExplainer (b) Tree-Grid, PGExplainer (c) Graph-SS5, GNNExplainer (d) Graph-SS5, PGExplainer

Fig. 4. Explanation fidelity (best viewed in color).

7.2.2 Explanation Fidelity. In addition to comparing to ground-truth explanations, we 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 three 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 Figure 4. From the figure, we can observe that when the 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, distribution-aware alignment strategies like the variant based on Gaussian mixture models demonstrate stronger fidelity in most cases. Besides, it can be noted that on Tree-grid the fidelity of mutual-information-based alignment is dependent on the number of edges and achieves better results with edge number within [8, 15].

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.

7.3 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 five times, and the average consistencies on datasets Tree-grid and Mutag are reported in Tables 3 and 4, 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-six edges is 4.39 for GNNExplainer. Introducing the auxiliary task of aligning embeddings can significantly improve explainers in terms of this criterion. Of these different alignment strategies, the variant based on Gaussian mixture model shows the strongest performance in most cases. After incorporating Align_Gaus on dataset TreeGrid, SHD distance of top-six edges drops from 4.39 to 2.13 for GNNExplainer and from 1.38 to 0.13 for PGExplainer. After incorporating it on dataset Mutag, SHD distance of top-six edges drops from 4.78 to 3.85 for GNNExplainer and from

Table 3. Consistency of Explanation in Terms of Average SHD across Five 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
+Align_MI	0.74	1.11	1.08	1.32	1.69	2.27
+Align_Gaus	0.68	1.16	1.13	0.72	1.39	2.13
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**
+Align_MI	0.28	0.19	0.27	0.15	0.20	0.16
+Align_Gaus	**0.05**	**0.08**	**0.10**	0.12	**0.19**	**0.13**

Table 4. Consistency of Explanation in Terms of Average SHD Distance across Five 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
+Align_MI	1.11	1.68	2.42	3.23	3.96	4.37
+Align_Gaus	1.03	1.51	2.19	3.02	3.38	3.85
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
+Align_MI	0.95	1.21	1.73	2.25	2.67	2.23
+Align_Gaus	0.59	1.34	1.13	**0.84**	**1.25**	**1.15**

3.42 to 1.15 for PGExplainer. These results validate the effectiveness of our proposal in obtaining consistent explanations.

7.4 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 with the ground-truth explanation. However, this result could be misleading, as faithfulness of explanation 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 Reference [57], 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. For example, Cycle is attached to Tree with probability

Table 5. Performance on MixMotif

		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 **0.795** (Higher is better)	0.748 **0.266** (Lower is better)

Two GNNs are trained with different γ. We check their performance in graph classification, then compare obtained explanations with the motif.

$\frac{2}{3}\gamma + \frac{1}{3}$, and to others with probability $\frac{1-\gamma}{3}$. So are the cases for House to Ladder and Grid to Wheel. Labels of obtained graphs are set as type of the motif. When γ is set to 0, each motif has the same probability of being attached to the three base graphs. In other words, there is no bias on which type of base graph to attach for each type of motif. Thus, we consider the dataset with $\gamma = 0$ as clean or bias-free. We would expect GNN trained on data with $\gamma = 0$ to focus on the motif structure for motif classification. 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. For each setting, the created dataset contains 3,000 graphs, and train:evaluation:test are split as $5:2:3$.

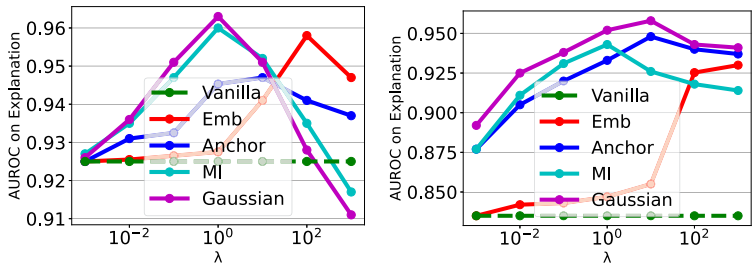
In this experiment, we set γ to 0 and 0.7 separately and train GNN f_0 and $f_{0.7}$ for each setting. Two models are tested in graph classification performance. Then, explanation methods are applied to and fine-tuned on f_0. Following that, these explanation methods are applied to explain $f_{0.7}$ using found hyperparameters. Results are summarized in Table 5.

From Table 5, 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.

7.5 Hyperparameter Sensitivity Analysis

In this part, we vary the hyperparameter λ to test the sensitivity of the proposed framework toward 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 within the scale $[1e-3, 1e-2, 1e-1, 1, 10, 1e2, 1e3\}$. PGExplainer is adopted as the base method. Experiments are randomly conducted three times on datasets Tree-grid and Mutag. Averaged results are visualized in Figure 5. From the figure, we can make the following observations:

— For all four variants, increasing λ has a positive effect at first, and the further increase would result in a performance drop. For example, on the Tree-grid dataset, best results of variants



(a) Tree-Grid (b) Mutag

Fig. 5. Sensitivity of PGExplainer towards weight of embedding alignment loss.

based on anchors, latent Gaussian mixture models, and mutual information scores are all obtained with λ around 1. When λ is small, the explanation alignment regularization in Equation (6) will be underweighted. However, a too-large λ may underweight the MMI-based explanation framework, which preserves the predictive power of obtained explanations.

— Among these variants, the strategy based on latent Gaussian mixture models shows the strongest performance in most cases. For example, for both datasets Tree-grid and Mutag, this variant achieves the highest AUROC scores on identified explanatory edges. However, the variant directly using Euclidean distances shows inferior performances in most cases. We attribute this to their different ability in modeling the distribution and conducting alignment.

8 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 the inference of GNNs, we propose a simple yet effective approach by aligning internal embeddings. Theoretical analysis shows that it is more faithful in design, optimizing an objective that encourages high MI between the original graph, GNN output, and identified explanation. Four different strategies are designed by directly adopting Euclidean distance, using anchors, KL divergence with Gaussian mixture models, and estimated MI scores. All these algorithms can be incorporated into existing methods with no effort. Experiments validate their effectiveness in promoting the faithfulness and consistency of explanations.

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. Besides, the evaluation of explanation methods also needs further studies. Existing benchmarks are usually clear and unambiguous, failing to simulate complex real-world scenarios.

ACKNOWLEDGMENTS

The findings and conclusions in this article do not necessarily reflect the view of the funding agency.

REFERENCES

[1] Wenqi Fan, Y. Ma, Qing Li, Yuan He, Y. Zhao, Jiliang Tang, and D. Yin. 2019. Graph neural networks for social recommendation. In *Proceedings of the World Wide Web Conference*.

[2] 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.

[3] Linli Xu, Wenjun Ouyang, Xiaoying Ren, Yang Wang, and Liang Jiang. 2018. Enhancing semantic representations of bilingual word embeddings with syntactic dependencies. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'18)*. 4517–4524.

[4] Elman Mansimov, O. Mahmood, Seokho Kang, and Kyunghyun Cho. 2019. Molecular geometry prediction using a deep generative graph neural network. *Sci. Rep.* 9 (2019).
[5] 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. *Stud. Health Technol. Inform.* 267 (2019), 181–186.
[6] Daniil Sorokin and Iryna Gurevych. 2018. Modeling semantics with gated graph neural networks for knowledge base question answering. *ArXiv* abs/1808.04126 (2018).
[7] Zhiwei Liu, Liangwei Yang, Ziwei Fan, Hao Peng, and Philip S. Yu. 2022. Federated social recommendation with graph neural network. *ACM Trans. Intell. Syst. Technol.* 13, 4 (2022), 1–24.
[8] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[9] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* 30 (2017).
[11] Weijieying Ren, Lei Zhang, Bo Jiang, Zhefeng Wang, Guangming Guo, and Guiquan Liu. 2017. Robust mapping learning for multi-view multi-label classification with missing labels. In *Proceedings of the 10th International Conference on Knowledge Science, Engineering and Management (KSEM'17)*. Springer, 543–551.
[12] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).
[13] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Adv. Neural Inf. Process. Syst.* 31 (2018).
[14] Zhilong Lu, Weifeng Lv, Zhipu Xie, Bowen Du, Guixi Xiong, Leilei Sun, and Haiquan Wang. 2022. Graph sequence neural network with an attention mechanism for traffic speed prediction. *ACM Trans. Intell. Syst. Technol.* 13, 2 (2022), 1–24.
[15] Xiaoying Ren, Linli Xu, Tianxiang Zhao, Chen Zhu, Junliang Guo, and Enhong Chen. 2018. Tracking and forecasting dynamics in crowdfunding: A basis-synthesis approach. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'18)*. IEEE, 1212–1217.
[16] 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).
[17] Weijieying Ren, Lei Wang, Kunpeng Liu, Ruocheng Guo, Lim Ee Peng, and Yanjie Fu. 2022. Mitigating popularity bias in recommendation with unbalanced interactions: A gradient perspective. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'22)*. IEEE, 438–447.
[18] Xiaoying Ren, Jing Jiang, Ling Min Serena Khoo, and Hai Leong Chieu. 2021. Cross-topic rumor detection using topic-mixtures. *Conference of the European Chapter of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:233189630
[19] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating explanations for graph neural networks. *Adv. Neural Inf. Process. Syst.* 32 (2019), 9240.
[20] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. *Adv. Neural Inf. Process. Syst.* 33 (2020), 19620–19631.
[21] Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. 2022. On consistency in graph neural network interpretation. *arXiv preprint arXiv:2205.13733* (2022).
[22] 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).
[23] Minh N. Vu and My T. Thai. 2020. PGMExplainer: Probabilistic graphical model explanations for graph neural networks. *arXiv preprint arXiv:2010.05788* (2020).
[24] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. In *Proceedings of the International Conference on Machine Learning*. PMLR, 12241–12252.
[25] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2020. Explainability in graph neural networks: A taxonomic survey. *arXiv preprint arXiv:2012.15445* (2020).
[26] 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).
[27] 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).
[28] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. 2021. GraphSMOTE: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.

[29] Yu Zhou, Haixia Zheng, Xin Huang, Shufeng Hao, Dengao Li, and Jumin Zhao. 2022. Graph neural networks: Taxonomy, advances, and trends. *ACM Trans. Intell. Syst. Technol.* 13, 1 (2022), 1–54.
[30] Tianxiang Zhao, Wenchao Yu, Suhang Wang, Lu Wang, Xiang Zhang, Yuncong Chen, Yanchi Liu, Wei Cheng, and Haifeng Chen. 2023. Skill disentanglement for imitation learning from suboptimal demonstrations. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
[31] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).
[32] Ke Liang, Jim Tan, Dongrui Zeng, Yongzhe Huang, Xiaolei Huang, and Gang Tan. 2023. ABSLearn: a GNN-based framework for aliasing and buffer-size information retrieval. *Pattern Analysis and Applications* 26 (2023), 1171–1189. https://api.semanticscholar.org/CorpusID:257129854
[33] Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. Reasoning over different types of knowledge graphs: Static, temporal and multi-modal. *arXiv preprint arXiv:2212.05767* (2022).
[34] 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 *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 2224–2232.
[35] James Atwood and Don Towsley. 2016. Diffusion-convolutional neural networks. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 1993–2001.
[36] 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.
[37] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum Chemistry. In *Proceedings of the International Conference on Machine Learning*.
[38] Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. 2022. TopoImb: Toward topology-level imbalance in learning from graphs. In *Proceedings of the Learning on Graphs Conference*. PMLR, 37–1.
[39] 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.
[40] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. 2022. Exploring edge disentanglement for node classification. In *Proceedings of the ACM Web Conference*. 1028–1036.
[41] 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.
[42] Shuai Lin, Pan Zhou, Zi-Yuan Hu, Shuojia Wang, Ruihui Zhao, Yefeng Zheng, Liang Lin, Eric P. Xing, and Xiaodan Liang. 2021. Prototypical graph contrastive learning. *IEEE Transactions on Neural Networks and Learning Systems*. https://api.semanticscholar.org/CorpusID:235457979
[43] Junjie Xu, Enyan Dai, Xiang Zhang, and Suhang Wang. 2022. HP-GMN: Graph memory networks for heterophilous graphs. *arXiv preprint arXiv:2210.08195* (2022).
[44] Muhammet Balcilar, Pierre Héroux, Benoit Gauzere, Pascal Vasseur, Sébastien Adam, and Paul Honeine. 2021. Breaking the limits of message passing graph neural networks. In *Proceedings of the International Conference on Machine Learning*. PMLR, 599–608.
[45] Ryoma Sato. 2020. A survey on the expressive power of graph neural networks. *arXiv preprint arXiv:2003.04078* (2020).
[46] Hoang Nt and Takanori Maehara. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550* (2019).
[47] Muhammet Balcilar, Renton Guillaume, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. 2021. Analyzing the expressive power of graph neural networks in a spectral perspective. In *Proceedings of the International Conference on Learning Representations (ICLR'21)*.
[48] Sheng-Hsuan Lin and Mohammad Arfan Ikram. 2019. On the relationship of machine learning with causal inference. *European Journal of Epidemiology* 35 (2019), 183–185. https://api.semanticscholar.org/CorpusID:203437853
[49] Federico Baldassarre and Hossein Azizpour. 2019. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686* (2019).
[50] 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).
[51] 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.
[52] 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).

[53] 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.
[54] Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. 2021. Reinforcement learning enhanced explainer for graph neural networks. *Adv. Neural Inf. Process. Syst.* 34 (2021).
[55] 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).
[56] Wanyu Lin, Hao Lan, and Baochun Li. 2021. Generative causal explanations for graph neural networks. In *Proceedings of the International Conference on Machine Learning*. PMLR, 6666–6679.
[57] 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).
[58] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. 2020. Graph information bottleneck for subgraph recognition. *arXiv preprint arXiv:2010.05563* (2020).
[59] Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *Proceedings of the IEEE Information Theory Workshop (ITW'15)*. IEEE, 1–5.
[60] Phúc H. Lê Khac, Graham Healy, and Alan F. Smeaton. 2020. Contrastive representation learning: A framework and review. *IEEE Access* 8 (2020), 193907–193934.
[61] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *ArXiv* abs/2004.11362 (2020).
[62] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1597–1607.
[63] Florian Graf, Christoph Hofer, Marc Niethammer, and Roland Kwitt. 2021. Dissecting supervised contrastive learning. *ArXiv* abs/2102.08817 (2021).
[64] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *ArXiv* abs/2005.10242 (2020).
[65] Shengyu Feng, Baoyu Jing, Yada Zhu, and Hanghang Tong. 2022. Adversarial graph contrastive learning with information regularization. In *Proceedings of the ACM Web Conference*. 1362–1371.
[66] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 70–79.
[67] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116* (2021).
[68] Lei Wang, Ee-Peng Lim, Zhiwei Liu, and Tianxiang Zhao. 2022. Explanation guided contrastive learning for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2017–2027.
[69] Kenny Ye Liang, Yue Liu, Sihang Zhou, Wenxuan Tu, Yi Wen, Xihong Yang, Xiang Dong, and Xinwang Liu. 2022. Knowledge graph contrastive learning based on relation-symmetrical structure. *IEEE Transactions on Knowledge and Data Engineering*. https://api.semanticscholar.org/CorpusID:259145427
[70] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. 2019. Strategies for pre-training graph neural networks. *arXiv: Learning* (2019).
[71] Felix L. Opolka, Aaron Solomon, Cătălina Cangea, Petar Velickovic, Pietro Lio', and R. Devon Hjelm. 2019. Spatio-temporal deep graph infomax. *ArXiv* abs/1904.06316 (2019).
[72] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Lio', Yoshua Bengio, and R. Devon Hjelm. 2018. Deep graph infomax. *ArXiv* abs/1809.10341 (2018).
[73] Qikui Zhu, Bo Du, and Pingkun Yan. 2020. Self-supervised training of graph convolutional networks. *ArXiv* abs/2006.02380 (2020).
[74] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *ArXiv* abs/2010.13902 (2020).
[75] Kaveh Hassani and Amir Hosein Khas Ahmadi. 2020. Contrastive multi-view representation learning on graphs. *ArXiv* abs/2006.05582 (2020).
[76] Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan. 2021. Multi-scale contrastive siamese networks for self-supervised graph representation learning. *ArXiv* abs/2105.05682 (2021).
[77] Yanqiao Zhu, Yichen Xu, Feng Yu, Q. Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *ArXiv* abs/2006.04131 (2020).

[78] Liang Zeng, Lanqing Li, Zi-Chao Gao, Peilin Zhao, and Jian Li. 2022. ImGCL: Revisiting graph contrastive learning on imbalanced node classification. *ArXiv* abs/2205.11332 (2022).
[79] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. 2021. Graph contrastive learning automated. In *Proceedings of the International Conference on Machine Learning*.
[80] Yihang Yin, Qingzhong Wang, Siyu Huang, Haoyi Xiong, and Xiang Zhang. 2021. AutoGCL: Automated graph contrastive learning via learnable view generators. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
[81] 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 *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Vol. 96. 226–231.
[82] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the International Conference on Learning Representations*.
[83] 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.
[84] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach. Learn.* 65, 1 (2006), 31–78.

Received 18 March 2023; revised 17 July 2023; accepted 24 July 2023