RegExplainer: Generating Explanations for Graph Neural Networks in Regression Tasks

Jiaxing Zhang[1], **Zhuomin Chen**[2], **Hao Mei**[3], **Longchao Da**[3], **Dongsheng Luo**[2], **Hua Wei**[3]
[1]New Jersey Institute of Technology, [2]Florida International University, [3]Arizona State University
[1]jz48@njit.edu, [2]{zchen051, dluo}@fiu.edu, [3]{hmei7, longchao, hua.wei}@asu.edu

Abstract

Graph regression is a fundamental task that has gained significant attention in various graph learning tasks. However, the inference process is often not easily interpretable. Current explanation techniques are limited to understanding Graph Neural Network (GNN) behaviors in classification tasks, leaving an explanation gap for graph regression models. In this work, we propose a novel explanation method to interpret the graph regression models (XAIG-R). Our method addresses the distribution shifting problem and continuously ordered decision boundary issues that hinder existing methods away from being applied in regression tasks. We introduce a novel objective based on the graph information bottleneck theory (GIB) and a new mix-up framework, which can support various GNNs and explainers in a model-agnostic manner. Additionally, we present a self-supervised learning strategy to tackle the continuously ordered labels in regression tasks. We evaluate our proposed method on three benchmark datasets and a real-life dataset introduced by us, and extensive experiments demonstrate its effectiveness in interpreting GNN models in regression tasks.

1 Introduction

Graph Neural Networks [1] (GNNs) have become a powerful tool for learning knowledge from graph-structure data and achieved remarkable performance in many areas, including social networks [2, 3], molecular structures [4, 5], traffic flows [6–9], and recommendation systems [10–12]. Despite the success, their popularity in sensitive fields such as fraud detection and drug discovery [13, 14] requires an understanding of their decision-making processes. To address this challenge, some efforts have been made to explain GNN's predictions in a post-hoc manner, which aims to find a sub-graph that preserves the information about the predicted label. On top of the intuitive principle, Graph Information Bottleneck (GIB) [15, 16] maximizes the mutual information $I(G^*; Y)$ between the target prediction label Y and the explanation G^* while constraining the size of the explanation.

However, existing methods focus on the explanation of the classification tasks, leaving another fundamental task, explainable regression, unexplored. Graph regression tasks exist widely in nowadays applications, such as predicting the molecular property [17] or traffic flow volume [18]. Explaining the instance-level predictions of graph regression is challenging due to two main obstacles. First, in the routinely adopted GIB framework, the mutual information between the explanation sub-graph and label, $I(G^*; Y)$, is estimated with the Cross-Entropy between the predictions $f(G^*)$ from GNN model f and its prediction label Y. However, in the regression task, the regression label is the continuous value, making the approximation unsuitable. Another challenge is the distribution shifting problem in the usage of $f(G^*)$, where the prediction of the explanation sub-graph made by the GNN model f is unsafe. Usually, explanation sub-graphs have different topology and feature information compared to the original graph. As a result, explanation sub-graphs are out-of-distribution of the

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

original training graph dataset [19–21]. As shown in Figure 1, a GNN model f is trained on the original graph training set and cannot be safely used to make predictions for sub-graphs.

To fill the gap, in this paper, we propose RegExplainer, to generate post-hoc instance-level explanations for graph regression tasks. Specifically, we formulate a theoretical-sound objective for explainable regression based on information theory. To further address the distribution shifting issue, RegExplainer develops a new mix-up approach with self-supervised learning. Our experiments show that RegExplainer provides consistent and concise explanations of GNN's predictions on regression tasks. We achieved up to 48.0% improvement when compared to the alternative baselines in our experiments. Our contributions can be summarized as follows.

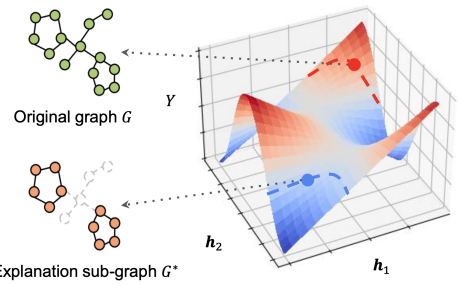


Figure 1: Intuitive illustration of the distribution shifting problem. The 3-dimensional map represents a trained GNN model f, where (h_1, h_2) represents the embedding distribution of the graph in two dimensions, and Y represents the prediction value of the graph through f. The red and blue lines represent the distribution of the original training graph set and the corresponding explanation sub-graph set, respectively. The distribution of G^* shifts away from the original distribution, resulting in shifted prediction values.

• To our best knowledge, we are the first to explain GNN predictions on graph regression tasks. We addressed two challenges in explaining the graph regression task: the mutual information estimation in the GIB objective and the distribution shifting problem with continuous decision boundaries.

• We proposed a novel model with self-supervised learning and the mix-up approach, which can address the two challenges more effectively, and better explain the graph model on the regression tasks compared to other baselines.

• We designed three synthetic datasets, namely BA-Motif-Volume, BA-Motif-Counting and Triangles, as well as a real-world dataset called Crippen, which can also be used in future works, to evaluate the effectiveness of our regression task explanations. Comprehensive empirical studies on both synthetic and real-world datasets demonstrate that our method can provide consistent and concise explanations for graph regression tasks.

2 Related Work and Further Discussions

GNN Explainability The explanation methods for GNN models can be categorized into two types based on their granularity: instance-level [22–25] and model-level [26], where the former methods explain the prediction for each instance by identifying important sub-graphs, and the latter method aims to understand the global decision rules captured by the GNN. These methods can also be classified into two categories based on their methodology: self-explainable GNNs [27, 28] and post-hoc explanation methods [23–25], where the former methods provide both predictions and explanations, while the latter methods use an additional model or strategy to explain the target GNN. Additionally, CGE [29] (cooperative explanation) generates the sub-graph explanation with the sub-network simultaneously, by using cooperative learning. However, it has to treat the GNN model as a white box, which is usually unavailable in the post-hoc explanation. Existing methods have only partially addressed the explanation of graph regression tasks and have not fully considered two important challenges: the distribution shifting problem and the limitations of the GIB objective, both of which are key areas our work aims to tackle.

GIB Objective The Information Bottleneck (IB) [30, 31] provides an intuitive principle for learning dense representations that an optimal representation should contain *sufficient* information for the downstream prediction task with a *minimal* size. Based on IB, a recent work [32] unifies the most existing post-hoc explanation methods for GNN, such as GNNExplainer [23], PGExplainer [24], with the graph information bottleneck (GIB) principle [15, 16, 32]. Formally, the objective of explaining

the prediction of f on G can be represented by

$$\arg\min_{G^*} I(G; G^*) - \alpha I(G^*; Y), \tag{1}$$

where G is the to-be-explained original graph, G^* is the explanation sub-graph of G, Y is the original ground-truth label of G, and α is a hyper-parameter to get the trade-off between minimal and sufficient constraints. GIB uses the mutual information $I(G; G^*)$ to select the minimal explanation that inherits only the most indicative information from G to predict the label Y by maximizing $I(G^*; Y)$, where $I(G; G^*)$ avoids imposing potentially biased constraints, such as the size or the connectivity of the selected sub-graphs [15]. Through the optimization of the sub-graph, G^* provides model interpretation. In graph classification task, a widely-adopted approximation to Eq. (1) in previous methods [23, 24] is:

$$\arg\min_{G^*} I(G; G^*) + \alpha H(Y|G^*) \approx \arg\min_{G^*} I(G; G^*) + \alpha \text{CE}(Y, Y^*),$$

where Y and Y^*, approximated by $f(G)$ and $f(G^*)$, is the predicted label of G and G^* made by the to-be-explained model f, and the cross-entropy $\text{CE}(Y, Y^*)$ between Y and Y^* is used to approximate $-I(G^*; Y)$. The approximation is based on the definition of mutual information $I(G^*; Y) = H(Y) - H(Y|G^*)$: with entropy $H(Y)$ being static and independent of the explanation process, minimizing the mutual information between the explanation sub-graph G^* and Y can be reformulated as maximizing the conditional entropy of Y given G^*, which can be approximated by $\text{CE}(Y, Y^*)$.

3 Preliminary

Notation and Problem Formulation We use $G = (\mathcal{V}, \mathcal{E}; \boldsymbol{X}, \boldsymbol{A})$ to represent a graph from an alphabet $\mathcal{G}$, where $\mathcal{V}$ equals to $\{v_1, v_2, ..., v_n\}$ represents a set of n nodes and $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ represents the edge set. Each graph has a feature matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ for the nodes, wherein $\boldsymbol{X}$, $X_i \in \mathbb{R}^{1 \times d}$ is the d-dimensional node feature of node v_i. $\mathcal{E}$ is described by an adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{n \times n}$, where $A_{ij} = 1$ means that there is an edge between node v_i and v_j; otherwise, $A_{ij} = 0$. For the graph prediction task, each graph G_k has a label $Y_k \in \mathcal{C}$, where $k \in \{1, ..., N\}$, N represents the number of graphs in the dataset, $\mathcal{C}$ is the set of the classification categories or regression values in $\mathbb{R}$, with a GNN model f trained to make the prediction, i.e., $f : (\boldsymbol{X}, \boldsymbol{A}) \mapsto \mathcal{C}$.

Problem 1 (Post-hoc Instance-level GNN Explanation)**.** *Given a trained GNN model f, for an arbitrary input graph $G = (\mathcal{V}, \mathcal{E}; \boldsymbol{X}, \boldsymbol{A})$, the goal of post-hoc instance-level GNN explanation is to find a sub-graph G^* that can explain the prediction of f on G.*

In non-graph structured data, the informative feature selection has been well studied [33], as well as in traditional methods, such as concrete auto-encoder [34], which can be directly extended to explain features in GNNs. In this paper, we focus on discovering the important sub-graph typologies following the previous work [23, 24]. Specifically, the obtained explanation G^* is depicted by a binary mask $\boldsymbol{M}^* \in \{0, 1\}^{n \times n}$ on the adjacency matrix, e.g., $G^* = (\mathcal{V}, \mathcal{E}; \boldsymbol{X}, \boldsymbol{A} \odot \boldsymbol{M}^*)$, $\odot$ means elements-wise multiplication. The mask highlights components of G which are essential for f to make the prediction.

4 Methodology

In this section, we first introduce a new objective based on GIB for explaining graph regression tasks. Then we showcase the distribution shifting problem in the objective for regression and propose a novel framework with the mix-up approach to solve the distribution shifting problem, by incorporating the mix-up approach with self-supervised contrastive learning.

4.1 GIB for Explaining Graph Regression

As introduced in Section 2, in the classification task, $I(G^*; Y)$ in Eq. (1) is commonly approximated by cross-entropy $\text{CE}(Y^*, Y)$ [35]. However, it is non-trivial to extend it for regression tasks because Y is a continuous variable and it is intractable to compute the cross-entropy $\text{CE}(Y^*, Y)$ or the mutual information $I(G^*; Y)$, where G^* is a graph variable with a continuous variable Y^* as its label.

4.1.1 Optimizing the Lower Bound of $I(G^*;Y)$

To address the challenge of computing the mutual information $I(G^*;Y)$ with a continuous Y, we propose a novel objective for explaining graph regression.

Instead of minimizing $I(G^*;Y)$ directly, we propose to maximize a lower bound for the mutual information by including the prediction label of G^*, denoted by Y^*, and approximate $I(G^*;Y)$ in Eq. (1) with $I(Y^*;Y)$:

$$\arg\min_{G^*} I(G;G^*) - \alpha I(Y^*;Y). \tag{2}$$

$I(Y^*;Y)$ has the following property, upon which we can approximate Eq. (2):

Property 1 *$I(Y^*;Y)$ is a lower bound of $I(G^*;Y)$.*

Intuitively, the property of $I(Y^*;Y)$ is guaranteed by the chain rule for mutual information and the independence between each explanation instance g^* in G^*. An intuitive demonstration is shown in Figure 2. The proof is shown in the Appendix B.1.

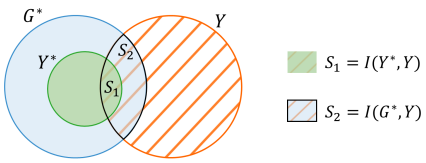


Figure 2: Intuitive illustration about why $I(G^*;Y) \geq I(Y^*;Y)$. G^* contains more mutual information as having more overlapping area with Y than the overlapping area between Y^* and Y.

4.1.2 Estimating $I(Y^*;Y)$ with InfoNCE

Now the challenge becomes the estimation of the mutual information $I(Y^*;Y)$. Inspired by the model of Contrastive Predictive Coding [36], in which InfoNCE loss is interpreted as a mutual information estimator, we further adapt the objective function so that it can be applied with InfoNCE loss in explaining graph regression. In our graph explanation scenario, the InfoNCE Loss defined in Eq. (3) can also be utilized as a lower bound of $I(Y^*;Y)$, as shown in the following property with proofs:

Property 2 InfoNCE Loss is a lower bound of the $I(Y^*;Y)$:

$$I(Y^*;Y) \geq \mathop{\mathbb{E}}_{\mathbb{Y}} \left[\log \frac{\text{sim}\left(Y^*,Y\right)}{\frac{1}{|\mathbb{Y}|}\sum_{Y' \in \mathbb{Y}} \text{sim}\left(Y^*,Y'\right)} \right], \tag{3}$$

where Y' is the prediction label of the randomly sampled graph neighbors, $\mathbb{Y}$ is the set of the neighbors' prediction labels, and sim() estimates the similarity between Y^* and Y. The proof is shown in the Appendix B.2. Therefore, we have the InfoNCE loss $\mathcal{L}_{\text{NCE}}$ as the lower bound of the $I(Y^*;Y)$. We approximate Eq. (2) as:

$$\arg\min_{G^*} I(G;G^*) - \alpha \mathop{\mathbb{E}}_{\mathbb{Y}} \left[\log \frac{\text{sim}\left(Y^*,Y\right)}{\frac{1}{|\mathbb{Y}|}\sum_{Y' \in \mathbb{Y}} \text{sim}\left(Y^*,Y'\right)} \right]. \tag{4}$$

4.2 Distribution Shifting Problem in Graph Regression

We include the prediction label Y^* in Eq. (4) to estimate similarity, which is approximated with $f(G^*)$ in previous work [23, 24]. However, we argue that $f(G^*)$ cannot be safely obtained due to the distribution shift problem [37, 19]. In classification tasks, a small shift may not cross the decision boundaries, which can still lead to a correct prediction. However, due to the continuous decision boundaries in regression, the distribution problem would cause serious prediction errors. Here in this paper, the graph distribution is indicated by its regression label in the regression task.

Figure 3 shows the existence of distribution shifts between $f(G^*)$ and $f(G)$ in graph regression tasks. For each dataset, we sort the indices of the data samples according to the value of their labels, and visualize the label Y, prediction $f(G)$ of the original graph from the trained GNN model f, and prediction $f(G^*)$ of the explanation sub-graph G^* from f. As we can see in Figure 3, in all four graph regression datasets, the red points are well distributed around the ground-truth blue points, indicating that $f(G)$ is close to Y. In comparison, the green points shift away from the red points, indicating the shifts between $f(G^*)$ and $f(G)$. Especially in dataset BA-Motif-Counting, the sub-graph explanation distribution was shifted extremely.

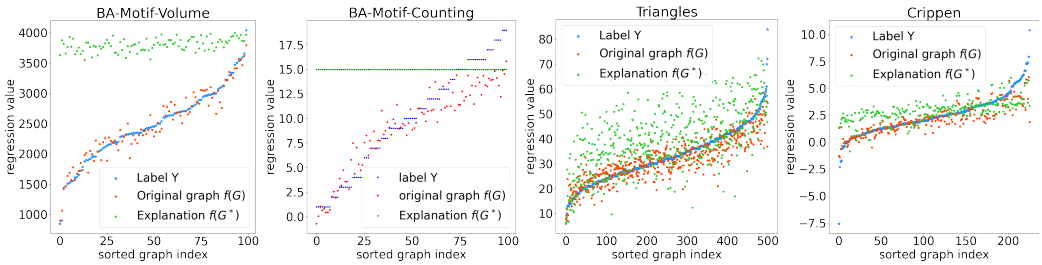


Figure 3: Visualization of distribution shifting problem on four graph regression datasets. The points represent the regression value, where the blue points mean ground truth label Y, red points mean prediction $f(G)$, and the green points mean prediction $f(G^*)$ on the four datasets. The x-axis is the indices of the graph, sorted by the value of the label Y.

Intuitively, this phenomenon indicates the GNN model f can make correct predictions only with the original graph G yet can not predict the explanation sub-graph G^* correctly. This is because the GNN model f is trained with the original graph sets, whereas the explanation G^* as the sub-graph is different from the original graph sets. With the shift between $f(G)$ and $f(G^*)$, the optimal solution in Eq. (4) is unlikely to work well.

4.3 Mix-up Approach with Contrastive Learning

To address this distribution-shifting problem in graph regression, we innovatively incorporate the mix-up approach with a self-supervised contrastive learning strategy. Instead of calculating Y^* with $f(G^*)$ directly, we approximate with $Y^{(\text{mix})}$ from $f(G^{(\text{mix})})$, which contains similar information as G^* but is in the same distribution with G. Specifically, our approach includes the following steps:

• **Step 1 (Neighbor Sampling):** Learning through the triplet instances can effectively reinforce the ability of the explainer to learn the explanation self-supervised. For each target graph G with label Y to be explained, we can define two randomly sampled graphs as positive neighbor G^+ and negative neighbor G^-, where G^+'s label Y^+ is closer to Y than G^-'s label Y^-, i.e., $|Y^+ - Y| < |Y^- - Y|$. Intuitively, the distance between the distributions of the positive pair $\langle G, G^+ \rangle$ should be smaller than the distance between the distributions of the negative pair $\langle G, G^- \rangle$.

• **Step 2 (Mixup for G^*):** Then we generate two mixup graphs $G^{(\text{mix})+}$ and $G^{(\text{mix})-}$ by mixing the sub-graph explanation G^* with the label irrelevant sub-graph $(G^+)^\Delta = G^+ - (G^+)^*$ from its positive neighbor G^+ and the label irrelevant sub-graph $(G^-)^\Delta = G^- - (G^-)^*$ from negative neighbor G^- respectively. Specifically, the label mixup approach is calculated from:

$$G^{(\text{mix})+} = G^* + (G^+)^\Delta = G^* + (G^+ - (G^+)^*), G^{(\text{mix})-} = G^* + (G^-)^\Delta = G^* + (G^- - (G^-)^*).$$

$G^{(\text{mix})+}$ and $G^{(\text{mix})-}$ should have the similar information to G because they have the same label-preserving sub-graphs G^*. Additionally, considering the following two pairs: $(G^{(\text{mix})+}, G^+)$ and $(G^{(\text{mix})-}, G^-)$. The similarity between $(G^{(\text{mix})+}, G^+)$ should be larger than the similarity between $(G^{(\text{mix})-}, G^-)$. Intuitively, since $G^{(\text{mix})+}$ and $G^{(\text{mix})-}$ have the same label-preserving sub-graphs G^* and $|Y^- - Y| > |Y^+ - Y|$, we can have $|f(G^-) - f(G^{(\text{mix})-})| > |f(G^+) - f(G^{(\text{mix})+})|$, where $f(G)$ represents the prediction label of graph G.

• **Step 3 (InfoNCE Loss Approximation):** Then we can safely estimate the similarity with $\text{sim}(Y^{(\text{mix})}, Y)$. To save more information, we use the similarity of representation embedding to approximate the similarity of the graph prediction label, where $\boldsymbol{h}^{(\text{mix})}$ represents the embedding for $G^{(\text{mix})}$ and $\boldsymbol{h}$ represents the embedding for G. We use $\mathbb{H}$ to represent the neighbors set accordingly. Thus, we approximate Eq. (4) as:

$$\arg\min_{G^*} I(G; G^*) - \alpha \mathop{\mathbb{E}}_{\mathbb{H}} \left[\log \frac{\text{sim}\left(\boldsymbol{h}^{(\text{mix})}, \boldsymbol{h}\right)}{\frac{1}{|\mathbb{H}|} \sum_{\boldsymbol{h}' \in \mathbb{H}} \text{sim}\left(\boldsymbol{h}^{(\text{mix})}, \boldsymbol{h}'\right)} \right]. \tag{5}$$

Different between Mix-up Approach in Classification Tasks The mix-up approach in previous work [19] generates a mixed graph by simply mixing explanation sub-graph G^* with a randomly sampled label-irrelevant sub-graph G^Δ [19], which can be formally written as $G_a^{(\text{mix})} = G_a^* + (G_b -$

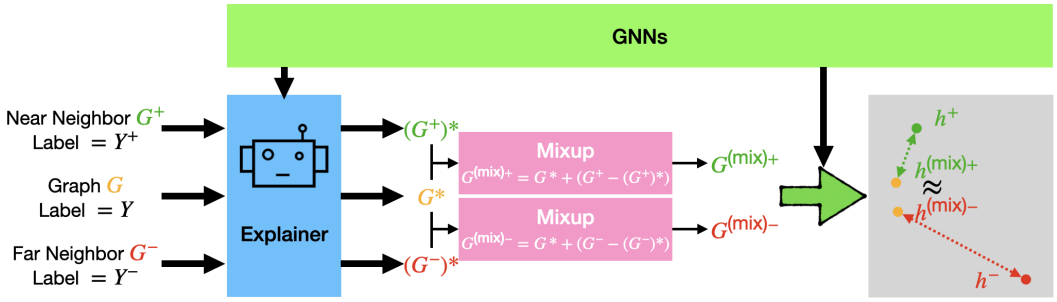


Figure 4: Illustration of RegExplainer. G is the to-be-explained graph, G^+ and G^- are the randomly sampled positive and negative neighbors. The explanation of the graph is produced by the explainer model. Then graph G^* is mixed with $(G^+)^\Delta = G^+ - (G^+)^*$ and $(G^-)^\Delta = G^- - (G^-)^*$ respectively to produce $G^{(\text{mix})+}$ and $G^{(\text{mix})-}$. Then the graphs are fed into the trained GNN model to retrieve the embedding vectors $\boldsymbol{h}^+$, $\boldsymbol{h}^-$, $\boldsymbol{h}^{(\text{mix})+}$ and $\boldsymbol{h}^{(\text{mix})-}$, where $\boldsymbol{h}^{(\text{mix})+} \approx \boldsymbol{h}^{(\text{mix})-}$ due to the same label-preserving sub-graph G^*. We use InfoNCE loss to minimize the distance between $G^{(\text{mix})+}$ and the positive sample and maximize the distance between $G^{(\text{mix})-}$ and the negative sample. The explainer is trained with the GIB objective and self-supervised contrastive loss.

G_b^*). However, it cann't tackle the continuous decision boundaries in graph regression tasks. A detailed description of the mix-up approach can be found in Appendix C.

4.4 Implementation

InfoNCE Loss After generating the mix-up explanation $G^{(\text{mix})}$, we specify the InfoNCE loss to further train the parameterized explainer with a triplet of graphs $\langle G, G^+, G^- \rangle$. In practice, G^+ and G^- are randomly sampled from the graph dataset, upon which we calculate their similarity score with the target graph G. The sample with a higher score would be the positive sample and the other one would be the negative sample. Specifically, we use $\text{sim}(\boldsymbol{h}, \boldsymbol{h}') = \boldsymbol{h}^\top \boldsymbol{h}'$ to compute the similarity score, where G' can be G^+ or G^-. $\boldsymbol{h}$ is generated by feeding G into the GNN model f and retrieving the embedding vector before the dense layers.

Formally, given a target graph G, the sampled positive graph G^+ and negative graph G^-, we formulate the InfoNCE loss in Eq. (5) as the following:

$$\mathcal{L}_{\text{NCE}}(G, G^+, G^-) = -\log \frac{\exp((\boldsymbol{h}^{(\text{mix})+})^\top \boldsymbol{h})}{\exp((\boldsymbol{h}^{(\text{mix})+})^\top \boldsymbol{h}^+) + \exp((\boldsymbol{h}^{(\text{mix})-})^\top \boldsymbol{h}^-)}, \tag{6}$$

where $\exp(\boldsymbol{h}^\top \boldsymbol{h})$ is used to instantiate the function sim, the denominator is a sum over the similarities of both positive and negative samples.

Size Constraints We optimize $I(G; G^*)$ in Eq. (5) to constraint the size of the explanation sub-graph G^*. The upper bound of $I(G; G^*)$ is optimized as the estimation of the KL-divergence between the probabilistic distribution between the G^* and G, where the KL-divergence term can be divided into two parts as the entropy loss and size loss [16]. In practice, we follow the previous work [23, 24, 38] to implement them. Specifically,

$$\mathcal{L}_{\text{size}}(G, G^*) = \gamma \sum_{(i,j)\in\mathcal{E}} (M_{ij}^*) - \log \sigma((\boldsymbol{h}^*)^\top \boldsymbol{h}^*), \tag{7}$$

where $\sum_{(i,j)\in\mathcal{E}} (M_{ij}^*)$ means sum the weights of the existing edges in the edge weight mask $\boldsymbol{M}^*$ for the explanation G^*; $\boldsymbol{h}^*$ is extracted from the embedding of the graph G^* before the GNN model f transforming it into prediction Y^*, σ means the sigmoid function and γ is the weight for the size of the masked graph. In implementation, we set $\gamma = (0.0003, 0.3)$ following previous work [19].

Overall Objective Function In practice, the denominator in Eq. (5) works as a regularization to avoid trivial solutions. Since the label Y is given and independent of the optimization process, we can also employ the MSE loss between Y^* and Y additionally, regarding InfoNCE loss only estimates the

Table 1: Illustration of the graph regression datasets together with the explanation faithfulness in terms of AUC-ROC on edges under four datasets on RegExplainer and other baselines. The original graph row visualizes the structure of the complete graph, the explanation row highlights the explanation sub-graph of the corresponding original graph. In the Crippen dataset, different colors of the node represent different kinds of atoms and the node feature is a one-hot vector to encode the atom type.

Dataset	BA-Motif-Volume	BA-Motif-Counting	Triangles	Crippen
Original Graph G				
Explanation G^*				
Node Feature	Random Float Vector	Fixed Ones Vector	Fixed Ones Vector	One-hot Vector
Regression Label	Sum of Motif Value	Number of Motifs	Number of Triangles	Chemical Property Value
Explanation Type	Fix Size Sub-Graph	Dynamic Size Sub-graph	Dynamic Size Sub-graph	Dynamic Size Sub-graph
Explanation AUC				
GRAD	0.418 ± 0.000	0.527 ± 0.000	0.479 ± 0.000	0.426 ± 0.000
ATT	0.512 ± 0.005	0.521 ± 0.003	0.441 ± 0.004	0.502 ± 0.006
MixupExplainer	0.471 ± 0.0291	0.868 ± 0.127	0.663 ± 0.110	0.499 ± 0.002
GNNExplainer	0.501 ± 0.009	0.505 ± 0.004	0.500 ± 0.002	0.497 ± 0.005
+RegExplainer	0.588 ± 0.017	0.629 ± 0.001	0.537 ± 0.003	0.541 ± 0.011
PGExplainer	0.470 ± 0.057	0.798 ± 0.133	0.511 ± 0.028	0.448 ± 0.005
+RegExplainer	$\mathbf{0.758 \pm 0.177}$	$\mathbf{0.989 \pm 0.003}$	$\mathbf{0.739 \pm 0.008}$	$\mathbf{0.553 \pm 0.013}$

mutual information between the embeddings. Formally, the overall loss function can be implemented as:

$$\mathcal{L} = \mathcal{L}_{\text{GIB}} + \beta \mathcal{L}_{\text{MSE}}(f(G), f(G^{(\text{mix})+})), \text{where } \mathcal{L}_{\text{GIB}} = \mathcal{L}_{\text{size}}(G, G^*) - \alpha \mathcal{L}_{\text{NCE}}(G, G^+, G^-) \quad (8)$$

$G^{(\text{mix})+}$ means mix G^* with the positive sample G^+ and α and β are hyper-parameters. The training algorithm and description of it are put in Appendix D.

5 Experiments

In this section, we conduct experiments to demonstrate the performance of our proposed method[1]. These experiments are mainly designed to explore the following research questions:

• **RQ1:** Can RegExplainer outperforms other baselines in explaining GNNs on regression tasks?

• **RQ2:** How does each part of RegExplainer and hyperparameters impact the overall performance in generating explanations?

• **RQ3:** Does the distribution shifting exist in GNN explanation? Can RegExplainer alleviate it?

5.1 Experiment Settings

We formulate Three synthetic datasets and a real-world dataset, as is shown in Table 1, in order to address the lack of graph regression datasets with ground-truth explanation. The datasets include: **BA-Motif-Volume** and **BA-Motif-Counting**, which are based on BA-shapes [23], **Triangles** [39], and **Crippen** [40]. We compared the proposed RegExplainer against a comprehensive set of baselines in all datasets, including: **GRAD** [23], **ATT** [41], **GNNExplainer** [23], **PGExplainer** [24], and **MixupExplainer** [19]. Detailed information about experiment setting are put in the Appendix E. We elaborate on the measurement metric of methods as follows: (1) **AUC-ROC**: We use the AUC score to evaluate the performance of our proposed methods against baseline methods regarding the ground-truth explanation, which can be treated as a binary classification task. (2) We evaluate the similarity of distribution of the graph with **Cosine Similarity** and **Euclidean Distance**.

5.2 Quantitative Evaluation (RQ1)

In this section, we evaluate the performance of our approach with other baselines. For GRAD and GAT, we use the gradient-based and attention-based explanation, following the setting in the

[1]Our data and code are available at: `https://github.com/jz48/RegExplainer`

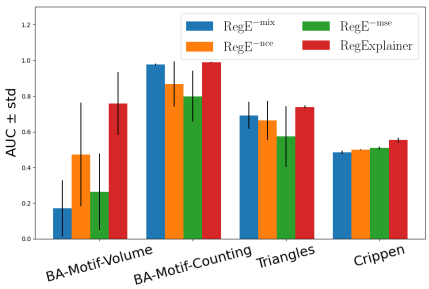
previous work [23]. We take GCN as our to-be-explained model for all post-hoc explainers. For GNNExplainer, PGExplainer, and MixupExplainer, which were previously used for the classification task, we replace the Cross-Entropy loss with the MSE loss. We run and tune all the baselines on our four datasets. We evaluate the explanation from all the methods with the AUC metric, as done in the previous work. As we can see in Table 1, we take GNNExplainer and PGExplainer as backbones and apply our framework as RegExplainer on both of them. The experiment results demonstrate the effectiveness of our methods in explaining graph regression tasks, where our method achieves the best performance compared to the baselines in all four datasets.

In Table 1, RegExplainer based on PGExplainer improves the second best baseline with $0.175/34.3\%$ on average and up to $0.246/48.0\%$. The comparison between RegExplainer and other baselines indicates the advantages of our proposed approach. This improvement indicates the effectiveness of our proposed method, showing that by incorporating the mix-up approach and contrastive learning, we can generate more faithful explanations in the graph regression tasks. In the following sections, we analyze the RegExplainer with PGExplainer as a backbone.

5.3 Ablation Study and Hyper-parameter Sensitivity Study (RQ2)

We conducted an ablation study to show how our proposed components, specifically, the mix-up approach and self-supervised learning, contribute to the final performance of RegExplainer. To this end, we denote RegExplainer as RegE and design three types of variants as follows: (1) RegE$^{-\text{mix}}$: We remove the mix-up processing after generating the explanations and feed the sub-graph G^* into the objective function directly. (2) RegE$^{-\text{nce}}$: We remove the InfoNCE loss term but still maintain the mix-up processing and MSE loss. (3) RegE$^{-\text{mse}}$: We remove the MSE loss computation item from the objective function.

Figure 5: Ablation study of RegExplainer. We evaluated the AUC performance of the original RegExplainer and its variants that exclude the mix-up approach, InfoNCE loss, or MSE loss, respectively. The black solid line shows the standard deviation.



Additionally, we set all variants with the same configurations as original RegExplainer, including learning rate, training epochs, and hyper-parameters η, α, and β. We trained them on all four datasets and conducted the results in Figure 5. We observed that the proposed RegExplainer outperforms its variants in all datasets, which indicates that each component is necessary and the combination of them is effective.

We also investigate the hyper-parameters of our approach, which include α and β, across all four datasets. The hyper-parameter α controls the weight of the InfoNCE loss in the GIB objective while the β controls the weight of the MSE loss. We determined the optimal values of α and β with grid search. The experimental results can be found in Figure 6. We fixed α and β at 1 and changed another parameter to visualize the change in model performance. Figure 6 illustrates that the model's performance is robust to changes in hyper parameters within the scope $[0.001, 1000]$. Our findings indicate that our approach, RegExplainer, is stable and robust when using different hyper-parameter settings, as evidenced by consistent performance across a range.

Figure 6: Hyper-parameters study of α and β on four datasets with RegExplainer. In both figures, the x-axis is the value of different hyper-parameter settings, and the y-axis is the value of the average AUC score over ten runs with different random seeds.

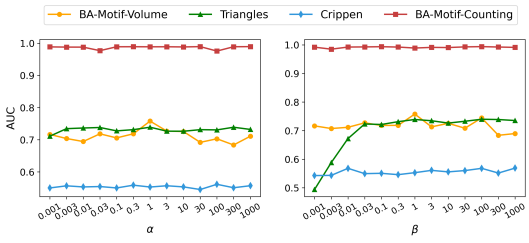


Table 2: Prediction shifting study on the RMSE of $(f(G), Y)$, $(f(G^*), Y)$, $(f(G), f(G^*))$ respectively.

Dataset	$(f(G), Y)$	$(f(G^*), Y)$	$(f(G), f(G^*))$
BA-Motif-Volume	131.42	1432.07	1427.07
BA-Motif-Counting	2.06	7.43	7.22
Triangles	5.28	12.38	12.40
Crippen	1.13	1.54	1.17

5.4 Alleviating Distribution Shifts (RQ3)

In this section, we visualize the regression values of the graphs and calculate the prediction shifting distance for each dataset and analyze their correlations to the distance of the decision boundaries. We put our results into Figure 7 and Table 2.

We observed that in Figure 3, red points surround the blue points but green points are shifted away, which indicates that the explanation sub-graph cann't help GNNs make correct predictions. As shown in Table 2, we calculate the RMSE score between the $f(G)$ and Y, $f(G^*)$ and Y, $f(G)$ and $f(G^*)$ respectively, where $f(G)$ is the prediction the original graph, $f(G^*$ is the prediction of the explanation sub-graph, and Y is the regression label. We can observe that $f(G^*)$ shows a significant prediction shifting from $f(G)$ and Y, indicating that the mutual information calculated with $(f(G^*), Y)$ would be biased.

We further explore the relationship of the prediction shifting against the label value with dataset BA-Motif-Volume, which represents the semantic decision boundary. This additional experiment with Figure 7 can be found in Appendix F.1.

We also design experiments to illustrate how RegExplainer corrects the deviations: we calculate the graph embeddings v and predictions p of the explanation sub-graphs and the mix-up graph. Then we compare them to the ground truth and calculate the Euclidean or Cosine distance between the vectors and RMSE between prediction labels. From the results in Table 3, we can observe that all the performances of COS(v_g, v_m), EUC(v_g, v_m) and prediction errors are better than those of (v_g, v_e), which indicates RegExplainer can effectively fix the distribution of sub-graph explanation G^* and reduce the embedding distance and prediction error.

Table 3: Table for measuring distribution repairing. v_g, v_e and v_m are the embeddings from f of original graph G, explanation subgraph G^* and the mix-up explanation $G^{(\text{mix})+}$. p_g, p_e and p_m are the predicted labels for the original graph, explanation subgraph and the mix-up explanation. EUC means Euclidean distance (↓, the smaller the better) and COS means cosine distance (↑, the larger the better). RMSE means Root Mean Square Error (↓, the smaller the better).

	BA-Motif-Volume	BA-Motif-Counting	Triangles	Crippen
COS(v_g, v_e)	0.95	0.80	0.97	0.89
COS(v_g, v_m)	0.98	0.89	0.99	0.92
EUC(v_g, v_e)	0.46	0.68	0.19	0.67
EUC(v_g, v_m)	0.37	0.52	0.08	0.63
RMSE(p_g, p_e)	1427.07	7.22	12.40	1.17
RMSE(p_g, p_m)	393.26	2.73	8.22	0.68

6 Conclusion

We addressed the challenges in the explainability of graph regression tasks and proposed the RegExplainer, a novel method for explaining the predictions of GNNs with the post-hoc explanation sub-graph on graph regression task without requiring modification of the underlying GNN architecture or re-training. We showed how RegExplainer can leverage the mix-up approach to solve the distribution shifting problem and adopt the GIB objective with the InfoNCE loss to migrate it from graph classification tasks to graph regression tasks, while these existing challenges seriously affect the performances of other explainers. We formulated four new datasets: BA-Motif-Volume, BA-Motif-Counting, Triangles, and Crippen for evaluating the explainers on the graph regression

task, which are aligned with the design of datasets in previous work. They can also benefit future studies on the XAIG-R. While we acknowledge the effectiveness of our method, we also recognize its limitations. Specifically, although our approach can be applied to explainers for graph regression tasks in an explainer-agnostic manner, it cannot be easily applied to explainers built for explaining the spatio-temporal graph due to the dynamic topology and node features of the STG. To overcome this challenge, a potential solution is to incorporate cached dynamic embedding memories into the framework.

7 Ethics Statement

This work is primarily foundational in GNN explainability, focusing on expanding the GIB objective function of the explainer framework from graph classification tasks to graph regression tasks. Its primary aim is to contribute to the academic community by enhancing the explanation in graph regression. We do not foresee any direct, immediate, or negative societal impacts stemming from the outcomes of our research.

8 Acknowledgments

The work was partially supported by NSF awards #2421839 and #2331908. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[1] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[2] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation, 2019.

[3] Shengjie Min, Zhan Gao, Jing Peng, Liang Wang, Ke Qin, and Bo Fang. Stgsn — a spatial–temporal graph neural network framework for time-evolving social networks. *Knowledge-Based Systems*, 214:106746, 2021.

[4] Hryhorii Chereda, Annalen Bleckmann, Frank Kramer, Andreas Leha, and Tim Beissbarth. Utilizing molecular network information via graph convolutional neural networks to predict metastatic event in breast cancer. In *GMDS*, pages 181–186, 2019.

[5] E. Mansimov, O. Mahmood, and S. Kang. Molecular geometry prediction using a deep generative graph neural network, 2019.

[6] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of The Web Conference 2020*, WWW '20, page 1082–1092, New York, NY, USA, 2020. Association for Computing Machinery.

[7] Mengzhang Li and Zhanxing Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4189–4196, May 2021.

[8] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 1907–1913. AAAI Press, 2019.

[9] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3634–3640. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[10] Shakila Shaikh, Sheetal Rathi, and Prachi Janrao. Recommendation system in e-commerce websites: a graph based approached. In *2017 IEEE 7th International Advance Computing Conference (IACC)*, pages 931–934. IEEE, 2017.

[11] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: A survey. *ACM Comput. Surv.*, 55(5), dec 2022.

[12] Kaige Yang and Laura Toni. Graph-based recommendation system. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 798–802. IEEE, 2018.

[13] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry*, 63(16):8749–8760, 2019.

[14] Pietro Bongini, Monica Bianchini, and Franco Scarselli. Molecular generative graph neural networks for drug discovery. *Neurocomputing*, 450:242–252, 2021.

[15] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck. *Advances in Neural Information Processing Systems*, 33:20437–20448, 2020.

[16] Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*, pages 15524–15543. PMLR, 2022.

[17] Marc Brockschmidt. Gnn-film: Graph neural networks with feature-wise linear modulation, 2020.

[18] Krzysztof Rusek, Paul Almasan, José Suárez-Varela, Piotr Chołda, Pere Barlet-Ros, and Albert Cabellos-Aparicio. Fast traffic engineering by gradient descent with learned differentiable routing, 2022.

[19] Jiaxing Zhang, Dongsheng Luo, and Hua Wei. Mixupexplainer: Generalizing explanations for graph neural networks with data augmentation. In *Proceedings of 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2023.

[20] Xu Zheng, Farhad Shirani, Tianchun Wang, Wei Cheng, Zhuomin Chen, Haifeng Chen, Hua Wei, and Dongsheng Luo. Towards robust fidelity for evaluating explainability of graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

[21] Zhuomin Chen, Jiaxing Zhang, Jingchao Ni, Xiaoting Li, Yuchen Bian, Md Mezbahul Islam, Ananda Mondal, Hua Wei, and Dongsheng Luo. Generating in-distribution proxy graphs for explaining graph neural networks. In *Forty-first International Conference on Machine Learning*, 2024.

[22] Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. Reinforcement learning enhanced explainer for graph neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22523–22533. Curran Associates, Inc., 2021.

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

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

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

[26] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. XGNN: towards model-level explanations of graph neural networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 430–438. ACM, 2020.

[27] Federico Baldassarre and Hossein Azizpour. Explainability techniques for graph convolutional networks, 2019.

[28] Enyan Dai and Suhang Wang. Towards self-explainable graph neural network, 2021.

[29] Junfeng Fang, Xiang Wang, An Zhang, Zemin Liu, Xiangnan He, and Tat-Seng Chua. *Cooperative Explanations of Graph Neural Networks*, page 616–624. Association for Computing Machinery, New York, NY, USA, 2023.

[30] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[31] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015.

[32] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. Graph information bottleneck for subgraph recognition. *arXiv preprint arXiv:2010.05563*, 2020.

[33] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Comput. Surv.*, 50(6), dec 2017.

[34] Muhammed Fatih Balın, Abubakar Abid, and James Zou. Concrete autoencoders: Differentiable feature selection and reconstruction. In *International conference on machine learning*, pages 444–453. PMLR, 2019.

[35] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[36] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

[37] Junfeng Fang, Xiang Wang, An Zhang, Zemin Liu, Xiangnan He, and Tat-Seng Chua. Cooperative explanations of graph neural networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 616–624, 2023.

[38] Yaochen Xie, Sumeet Katariya, Xianfeng Tang, Edward Huang, Nikhil Rao, Karthik Subbian, and Shuiwang Ji. Task-agnostic graph explanations, 2022.

[39] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures?, 2020.

[40] John S Delaney. Esol: estimating aqueous solubility directly from molecular structure. *Journal of chemical information and computer sciences*, 44(3):1000–1005, 2004.

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

[42] Scott A Wildman and Gordon M Crippen. Prediction of physicochemical parameters by atomic contributions. *Journal of chemical information and computer sciences*, 39(5):868–873, 1999.

[43] Benjamin Sanchez-Lengeling, Jennifer Wei, Brian Lee, Emily Reif, Peter Wang, Wesley Qian, Kevin McCloskey, Lucy Colwell, and Alexander Wiltschko. Evaluating attribution for graph neural networks. *Advances in neural information processing systems*, 33:5898–5910, 2020.

A Symbols Table

Symbol Name	Symbol Meaning
G	Original to-be-explained graph
G^*	Optimized sub-graph explanation
Y	Prediction label for G
Y^*	Prediction label for G^*
$I(\cdot)$	Mutual Information
$f(\cdot)$	Prediction made by to-be-explained GNN model f
(h_1, h_2)	The 2-dims representation of graph embeddings
α	Hyper parameter for mutual information term in GIB
$H(\cdot)$	Information entropy
$\mathcal{V}$	Node set
$\mathcal{E}$	Edge set
$\boldsymbol{X}$	Feature matrix
$\boldsymbol{A}$	Adjacency matrix
$\mathcal{G}$	Graph set
v	A node in graph
v_i	The i-th node in graph
n	Number of nodes
d	Dimension of feature
i	The i-th node index
j	The j-th node index
k	The k-th graph index
A_{ij}	edge from node i to node j
$\mathcal{C}$	A set of classification categories or $\mathbb{R}$ for regression tasks
$\boldsymbol{M}^*$	Edge mask which denotes G^* in G
$\mathbb{Y}$	Set of neighbors' prediction labels
Y'	Prediction label of sampled graph neighbor
$\boldsymbol{h}$	Graph embedding for G
$\mathbb{H}$	Set of $\boldsymbol{h}$
γ	Hyper parameter for leveraging sum of edge weights
σ	Sigmoid function
β	Hyper parameter for MSE loss
v_g	Embedding vectors for graph G
v_e	Embedding vectors for explanation sub-graph G^*
v_m	Embedding vectors for mix-up graph $G^{(\text{mix})+}$
p_g	Prediction labels for graph G
p_e	Prediction labels for explanation sub-graph G^*
p_m	Prediction labels for mix-up graph $G^{(\text{mix})+}$
$h(\cdot)$	Mapping function from G^* to Y^*
y^*	An instance of Y^*
g^*	An instance of G^*

Table 4: Important notations and symbols table.

B Proof

B.1 Property 1

Proof. From the definition of Y^*, we can make a safe assumption that there is a many-to-one map (function), denoted by h, from G^* to Y^* as Y^* is the prediction label for G^*. For simplicity, we assume a finite number of explanation instances for each label y^*, and each explanation instance, denoted by g^*, is generated independently. Then, we have $p(y^*) = \sum_{g^* \in \mathbb{G}(y^*)} p(g^*)$, where $\mathbb{G}(y^*) = \{g | h(g) = y^*\}$ is the set of explanations whose labels are y^*.

Based on the definition of mutual information, we have:

$$
\begin{aligned}
I(G^*; Y) &= \int_y \int_{g^*} p_{(G^*,Y)}(g^*, y) \log \frac{p_{(G^*,Y)}(g^*, y)}{p_{G^*}(g^*) p_Y(y)} d_{g^*} d_y \\
&= \int_y \int_{g^*} p_{(G^*,Y^*,Y)}(g^*, h(g^*), y) \\
&\qquad \log \frac{p_{(G^*,Y^*,Y)}(g^*, h(g^*), y)}{p_{G^*}(g^*) p_Y(y)} d_{g^*} d_y \\
&= \int_y \int_{g^*} p_{(G^*,Y^*,Y)}(g^*, h(g^*), y) \\
&\qquad \log \frac{p_{(G^*,Y^*,Y)}(g^*, h(g^*), y)}{p_{(G^*,Y^*)}(g^*, h(g^*)) p_Y(y)} d_{g^*} d_y \\
&= \int_y \int_{y^*} \sum_{g^* \in \mathbb{G}(y^*)} p_{(G^*,Y^*,Y)}(g^*, y^*, y) \\
&\qquad \log \frac{p_{(G^*,Y^*,Y)}(g^*, y^*, y)}{p_{(G^*,Y^*)}(g^*, y^*) p_Y(y)} d_{y^*} d_y
\end{aligned}
$$

Based on our many-to-one assumption, while each g^* is generated independently, we know that if $g \notin \mathbb{G}(y^*)$, then we have $p_{(G^*,Y^*,Y)}(g^*, y^*, y) = 0$. Thus, we have:

$$
\begin{aligned}
I(G^*; Y) &= I(G^*; Y) \\
&+ \int_y \int_{y^*} \sum_{g \notin \mathbb{G}(y^*)} p_{(G^*,Y^*,Y)}(g^*, y^*, y) \\
&\qquad \log \frac{p_{(G^*,Y^*,Y)}(g^*, y^*, y)}{p_{(G^*,Y^*)}(g^*, y^*) p_Y(y)} d_{y^*} d_y \\
&= \int_y \int_{y^*} \int_{g*} p_{(G^*,Y^*,Y)}(g^*, y^*, y) \\
&\qquad \log \frac{p_{(G^*,Y^*,Y)}(g^*, y^*, y)}{p_{(G^*,Y^*)}(g^*, y^*) p_Y(y)} d_{g^*} d_{y^*} d_y \\
&= I(G^*, Y^*; Y).
\end{aligned}
$$

With the chain rule for mutual information, we have $I(G^*, Y^*; Y) = I(Y^*; Y) + I(G^*; Y | Y^*)$. Then due to the non-negativity of the mutual information, we have $I(G^*, Y^*; Y) \geq I(Y^*; Y)$. □

B.2 Property 2

Proof. As in the InfoNCE method, the mutual information between Y^* and Y is defined as:

$$
I(Y^*; Y) = \sum_{Y^*, Y} p(Y^*, Y) \log \frac{p(Y | Y^*)}{P(Y)} \tag{9}
$$

However, the ground truth joint distribution $p(Y^*, Y)$ is not controllable, so, we turn to maximize the similarity

$$
\text{sim}\,(Y^*, Y) \propto \frac{p(Y | Y^*)}{p(Y)}. \tag{10}
$$

We want to put the representation function of mutual information into the NCE Loss

$$\mathcal{L}_N = -\mathop{\mathbb{E}}_{\mathbb{Y}} \log \left[\frac{\text{sim}\,(Y^*, Y)}{\sum_{Y' \in \mathbb{Y}} (Y^*, Y')} \right], \tag{11}$$

where $\mathcal{L}_{\text{N}}$ denotes the NCE loss. By inserting the optimal $\text{sim}\,(Y^*, Y)$ into Eq. (11), we can get:

$$\begin{aligned}
\mathcal{L}_{\text{NCE}} &= -\mathop{\mathbb{E}}_{\mathbb{Y}} \log \left[\frac{\frac{p(Y|Y^*)}{p(Y)}}{\frac{p(Y|Y^*)}{p(Y)} + \sum_{Y' \in \mathbb{Y}_{\text{neg}}} \frac{p(Y^*, Y')}{p(Y')}} \right] \\
&= \mathop{\mathbb{E}}_{\mathbb{Y}} \log \left[1 + \frac{p(Y|Y^*)}{p(Y)} \sum_{Y' \in \mathbb{Y}_{\text{neg}}} \frac{p(Y^*, Y')}{p(Y')} \right] \\
&\approx \mathop{\mathbb{E}}_{\mathbb{Y}} \log \left[1 + \frac{p(Y|Y^*)}{p(Y)} (N-1) \mathop{\mathbb{E}}_{Y'} \frac{p(Y^*, Y')}{p(Y')} \right] \\
&= \mathop{\mathbb{E}}_{\mathbb{Y}} \log \left[1 + \frac{p(Y|Y^*)}{p(Y)} (N-1) \right] \\
&\geq \mathop{\mathbb{E}}_{\mathbb{Y}} \log \left[\frac{p(Y|Y^*)}{p(Y)} N \right] \\
&= -I(Y^*, Y) + \log(N)
\end{aligned} \tag{12}$$

□

C Graph Mix-up Approach

To address the distribution shifting issue between $f(G)$ and $f(G^*)$ in the GIB objective, we introduce the mix-up approach to reconstruct a within-distribution graph, $G^{(\text{mix})}$, from the explanation graph G^*. We follow [24] to make a widely-accepted assumption that a graph can be divided by $G = G^* + G^{\Delta}$, where G^* presents the underlying sub-graph that makes important contributions to GNN's predictions, which is the expected explanatory graph, and G^{Δ} consists of the remaining label-independent edges for predictions made by the GNN. Both G^* and G^{Δ} influence the distribution of G. Therefore, we need a graph $G^{(\text{mix})}$ that contains both G^* and G^{Δ}, upon which we use the prediction of $G^{(\text{mix})}$ made by f to approximate Y^* and $\boldsymbol{h}^*$.

Specifically, for a target graph G_a in the original graph set to be explained, we generate the explanation sub-graph $G_a^* = G_a - G_a^{\Delta}$ from the explainer. To generate a graph in the same distribution of original G_a, we can randomly sample a graph G_b from the original set, generate the explanation sub-graph of G_b^* with the same explainer and retrieve its label-irrelevant graph $G_b^{\Delta} = G_b - G_b^*$. Then we can merge G_a^* together with G_b^{Δ} and produce the mix-up explanation $G_a^{(\text{mix})}$. Formally, we can have $G_a^{(\text{mix})} = G_a^* + (G_b - G_b^*)$.

Since we are using the edge weights mask to describe the explanation, we can denote G_a and G_b with the adjacency matrices $\boldsymbol{A}_a$ and $\boldsymbol{A}_b$, their edge weight mask matrices as $\boldsymbol{M}_a$ and $\boldsymbol{M}_b$. If G_a and G_b are aligned graphs with the same number of nodes, we can simply mix them up by $\boldsymbol{M}_a^{(\text{mix})} = \boldsymbol{M}_a^* + (\boldsymbol{I}_b - \boldsymbol{M}_b^*)$, where $\boldsymbol{M}$ denotes the weight of the adjacency matrix and $\boldsymbol{I}_b$ denotes the zero-ones matrix as weights of all edges in the adjacency matrix of G_b, where 1 represents the existing edge and 0 represents there is no edge between the node pair.

If G_a and G_b are not aligned with the same number of nodes, we can use a connection adjacency matrix $\boldsymbol{A}_{\text{conn}}$ and mask matrix $\boldsymbol{M}_{\text{conn}}$ to merge two graphs with different numbers of nodes. Specifically, the mix-up adjacency matrix can be formed as:

$$\boldsymbol{A}_a^{(\text{mix})} = \begin{bmatrix} \boldsymbol{A}_a & \boldsymbol{A}_{\text{conn}} \\ \boldsymbol{A}_{\text{conn}}^T & \boldsymbol{A}_b \end{bmatrix}. \tag{13}$$

And the mix-up mask matrix can be formed as:

$$\boldsymbol{M}_a^{(\text{mix})} = \begin{bmatrix} \boldsymbol{M}_a^* & \boldsymbol{M}_{\text{conn}} \\ \boldsymbol{M}_{\text{conn}}^T & \boldsymbol{M}_b^{\Delta} \end{bmatrix} \tag{14}$$

Finally, we can form $G_a^{(\text{mix})}$ as $(\boldsymbol{X}^{(\text{mix})}, \boldsymbol{A}_a^{(\text{mix})} \odot \boldsymbol{M}_a^{(\text{mix})})$, where $\boldsymbol{X}^{(\text{mix})} = [\boldsymbol{X}_a; \boldsymbol{X}_b]$. The detailed algorithm for mix-up is shown in Algorithm 1. In implantation, $\eta = |\mathcal{E}| * 0.03$.

Algorithm 1 Graph Mix-up Algorithm

Input: Target to-be-explained graph $G_a = (\boldsymbol{X}_a, \boldsymbol{A}_a)$, G_b sampled from a set of graphs $\mathcal{G}$, the number of random connections η, explainer model E.
Output: Graph $G^{(\text{mix})}$.

```
Generate mask matrix M_a = E(G_a)
Generate mask matrix M_b = E(G_b)
Sample η random connections between G_a and G_b as A_conn
Mix-up adjacency matrix A_a^(mix) with Eq. (13)
Mix-up edge mask M_a^(mix) with Eq. (14)
Mix-up node features X^(mix) = [X_a; X_b]
return G^(mix) = (X^(mix), A_a^(mix) ⊙ M_a^(mix))
```

D Training Algorithm

Algorithm 2 Training Explainer

Input: A set of graphs $\mathcal{G}$, trained GNN model f, explainer model E.
Output: Trained explainer E.

```
Initialize explainer model E.
for e ∈ epochs do
    for G ∈ 𝒢 do
        G_b, G_c ← Randomly sample two graphs from 𝒢
        G^+, G^- ← Compare similarity(G_b, G_c) to G
        G^(mix)+ ← Mix-up (G, G^+)
        G^(mix)− ← Mix-up (G, G^−)
        Compute L_NCE(G, G^+, G^−) with Eq. (6)
        Compute L_GIB and overall loss L with Eq. (8)
    end for
    Update E with back propagation.
end for
return Explainer E
```

Algorithm 2 shows the training procedure for our explainer. For each epoch and each to-be-explained graph G, we first randomly sample two neighbors and decide the positive neighbor G^+ and negative neighbor G^- according to the similarity between their embedding vectors respectively. The graph with higher similarity to G is the positive neighbor G^+. We generate the explanation for graphs and mix G with G^+ and G^- respectively. We calculate the InfoNCE for triplet $\langle G, G^+, G^- \rangle$ with Eq. (6) and the GIB loss, which contains the size loss and InfoNCE loss. We also calculate the MSE loss between $f(G^{(\text{mix})+})$ and $f(G)$. The overall loss is the sum of size loss, InfoNCE loss, and MSE loss. We update the trainable parameters in the explainer with the overall loss.

E Implantation details

We provided implementation details for our experiments in this section. Data and code are available in the Anumuous git repo and supplementary.

All experiments are conducted on a Linux machine (Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-210-generic x86_64)) with 4 NVIDIA TITAN Xp (12 GB) GPUs. CUDA version is 11.8 and the Driver version is 520.56.06. All codes are written with the Python version 3.8.13 with PyTorch 1.12.1 and PyTorch Geometric (PyG) 2.1.0.post1, torch-scatter 2.0.9, and torch-sparse 0.6.15. We adopt the Adam optimizer throughout all experiments. Overall, for each dataset, a GCN regression model is well-trained first, where we take a **three-layer GCN** model as the backbone. Then the explainers take the to-be-explained GNN and original graph and generate explanations for its prediction on the

dataset. After that, we evaluate the performance of the explanation. We split the dataset into 8:1:1, where we train the GNN base model with 8 folds, and train and test explainer models with 1 fold respectively. The hyper-parameters are illustrated in the paper correspondingly.

E.1 Datasets

We formulate Three synthetic datasets and a real-world dataset, as is shown in Table 1, in order to address the lack of graph regression datasets with ground-truth explanations. (1) *BA-Motif-Volume*: This dataset is based on the BA-shapes [23] and makes a modification, which is adding random float values from [0.00, 100.00] as the node feature. We then sum the node values on the motif as the regression label of the whole graph, which means the GNNs should recognize the [house] motif and then sum features to make the prediction. (2) *BA-Motif-Counting*: Different from BA-Motif-Volume, where node features are summarized, in this dataset, we attach various numbers of motifs to the base BA random graph and pad all graphs to equal size. The number of motifs is counted as the regression label. Additionally, we pad base graphs to dynamic size to prevent the GNNs from making trivial predictions based on the total number of nodes. (3) *Triangles*: We follow the previous work [39] to construct this dataset. The dataset is a set of 5000 Erdős–Rényi random graphs denoted as $ER(m, p)$, where $m = 30$ is the number of nodes in each graph and $p = 0.2$ is the probability for an edge to exist. The size of 5000 was chosen to match the previous work. The regression label for this dataset is the number of triangles in a graph and GNNs are trained to count the triangles. (4) *Crippen*: The Crippen dataset is a real-life dataset that was initially used to evaluate the graph regression task. The dataset has 1127 graphs reported in the Delaney solubility dataset [40] and has weights of each node assigned by the Crippen model [42], which is an empirical chemistry model predicting the water-actual partition coefficient. We adopt this dataset, firstly shown in the previous work [43], and construct edge weights by taking the average of the two connected nodes' weights.

E.2 Baselines

We compared the proposed RegExplainer against a comprehensive set of baselines in all datasets, including: (1) **GRAD** [23]: GRAD is a gradient-based method that learns weight vectors of edges by computing gradients of the GNN's objective function. (2) **ATT** [41]: ATT is a graph attention network (GAT) that learns attention weights for edges in the input graph. These weights can be utilized as a proxy measure of edge importance. (3) **GNNExplainer** [23]: GNNExplainer is a model-agnostic method that learns an adjacency matrix mask by maximizing the mutual information between the predictions of the GNN and the distribution of possible sub-graph structures. (4) **PGExplainer** [24]: PGExplainer adopts a deep neural network to parameterize the generation process of explanations, which facilitates a comprehensive understanding of the predictions made by GNNs. It also produces sub-graph explanations with edge importance masks. (5) **MixupExplainer** [19]: MixupExplainer adopts the graph mix-up approach with PGExplainer and address the Out-Of-Distribution problem in graph classification tasks.

F Additional Experiments

F.1 Correlation between Prediction Shifting and the Label Value

In Figure 7, each point represents a graph instance, where Y represents the ground-truth label, and Δ represents the absolute value difference. It's clear that both the $\Delta(f(G^*), Y)$ and $\Delta(f(G), f(G^*))$ strongly correlated to Y with statistical significance, indicating the prediction shifting problem is related to the continuous ordered decision boundary, which is present in regression tasks.

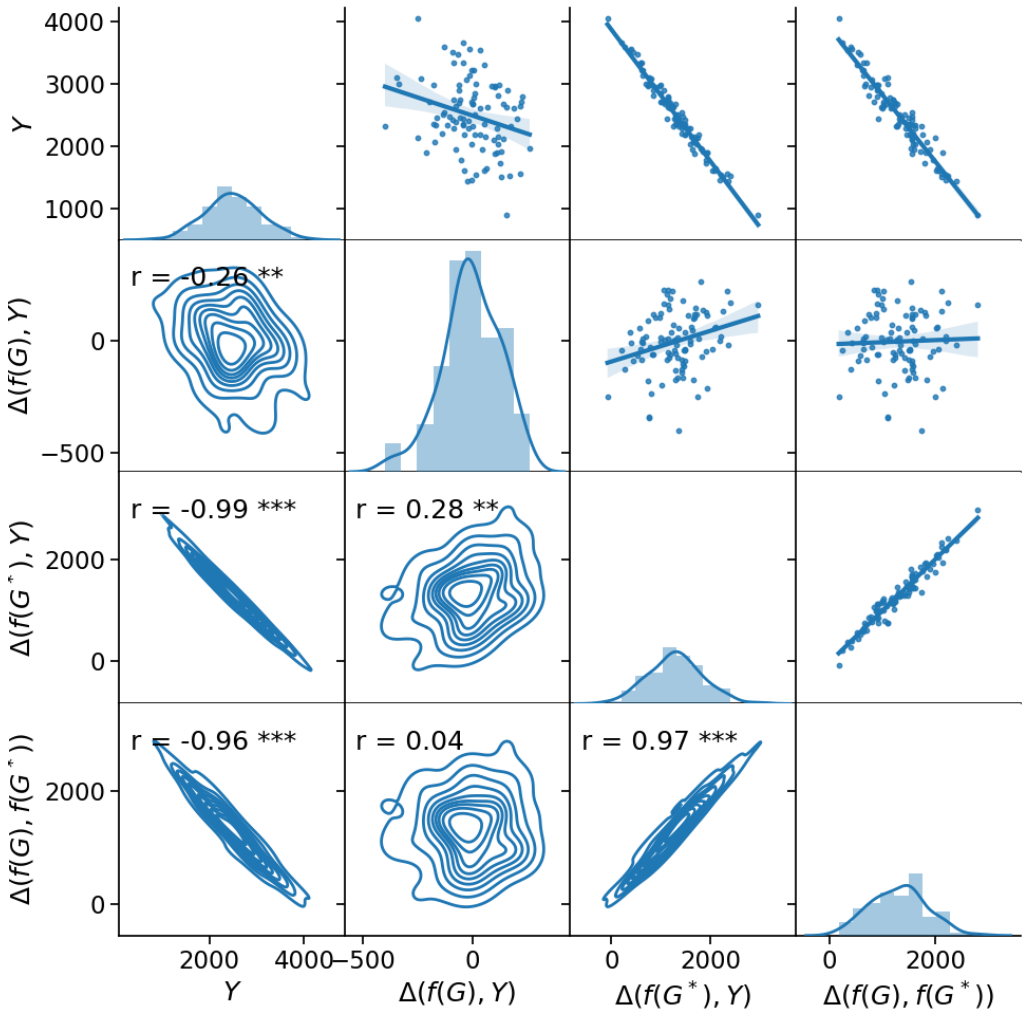


Figure 7: Correlations between the predictions and their shifting on BA-Motif-Volume. The value of r indicates the Pearson Correlation Coefficient, and the values with * indicate statistical significance for correlation, where *** indicates the p-value for testing non-correlation $p \leq 0.001$. Each point represents one graph instance.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We discussed the background scope and introduced our contributions in abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We provided the discussion about the limitations at the end of conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provided assumptions and proofs to our theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provided the hyper-parameters setting.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provided code and data in supplementary.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is presented in the paper and The full details can be found in the appendix and supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We conduct our experiments in 10 random seeds and report the mean and std values in the performance table.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided the information of computing resources in the section of implementation details in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [NA]

   Justification: Our research doesn't have a concern about the ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed, other than helping human beings better understand the black-box GNNs models.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We produce the data by ourselves and the original papers we refer to are cited. The baseline model we used are also cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provided the details about our new proposed datasets in the paper.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with899 human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.