
Evaluating Post-hoc Explanations for Graph Neural Networks via Robustness Analysis

Junfeng Fang¹, Wei Liu^{1†}, Yuan Gao¹, Zemin Liu², An Zhang²,
Xiang Wang^{1*}, Xiangnan He^{1*}

¹University of Science and Technology of China,

²National University of Singapore

{fjf, lw0118, yuangao}@ustc.edu.com,

{zeminliu, an_zhang}@nus.edu.sg,

{xiangwang1223, xiangnanhe}@gmail.com

Abstract

This work studies the evaluation of explaining graph neural networks (GNNs), which is crucial to the credibility of post-hoc explainability in practical usage. Conventional evaluation metrics, and even explanation methods — which mainly follow the paradigm of feeding the explanatory subgraph to the model and measuring output difference — mostly suffer from the notorious out-of-distribution (OOD) issue. In this work, we endeavor to confront this issue by introducing a novel evaluation metric, termed OOD-resistant Adversarial Robustness (OAR). Specifically, we draw inspiration from adversarial robustness and evaluate post-hoc explanation subgraphs by calculating their robustness under attack. On top of that, an elaborate OOD reweighting block is inset into the pipeline to confine the evaluation process to the original data distribution. For applications involving large datasets, we further devise a Simplified version of OAR (SimOAR), which achieves a significant improvement in computational efficiency at the cost of a small amount of performance. Extensive empirical studies validate the effectiveness of our OAR and SimOAR. Code is available at https://github.com/MangoKiller/SimOAR_OAR.

1 Introduction

Post-hoc explainability has manifested its extraordinary power to explain graph neural networks (GNNs) [1, 2, 3, 4]. Given a GNN-generated prediction for a graph, it aims to identify an explanatory subgraph, which is expected to best support the prediction and make the decision-making process more credible, fair, and understandable [5, 6, 7]. However, the reliable evaluation of explanation quality remains a key challenge. As a primary solution, *Human supervision* seeks to justify whether the explanations align with human knowledge [8, 9], but it is often too subjective, thus hardly providing quantifiable assessments. Another straightforward solution is quantitatively measuring the agreement between the generated and ground-truth explanations, such as *Precision* and *Recall* [10, 11]. Unfortunately, access to the ground truth is usually unavailable and labor-extensive, thereby limiting the scope of evaluations based on this method.

Recently, a compromised paradigm — *Feature Removal* [12, 13] — has been prevailing to quantitatively evaluate the explanation’s predictive power as compared to the full graph, without exploiting the human supervision and ground truth. The basic idea is to first remove the unimportant features and feed the remaining part (*i.e.*, explanatory subgraph) into the GNN, and then observe how the prediction changes. The prediction discrepancy instantiates *Accuracy* [8] and *Fidelity* [9] of the explanation, reflecting “how accurate and faithful the explanation is to recover the prediction of the

[†]Liu Wei is equal contribution to this paper.

^{*}Corresponding authors. Xiang Wang and Xiangnan He are also affiliated with Institute of Artificial Intelligence, Institute of Dataspace, Hefei Comprehensive National Science Center.

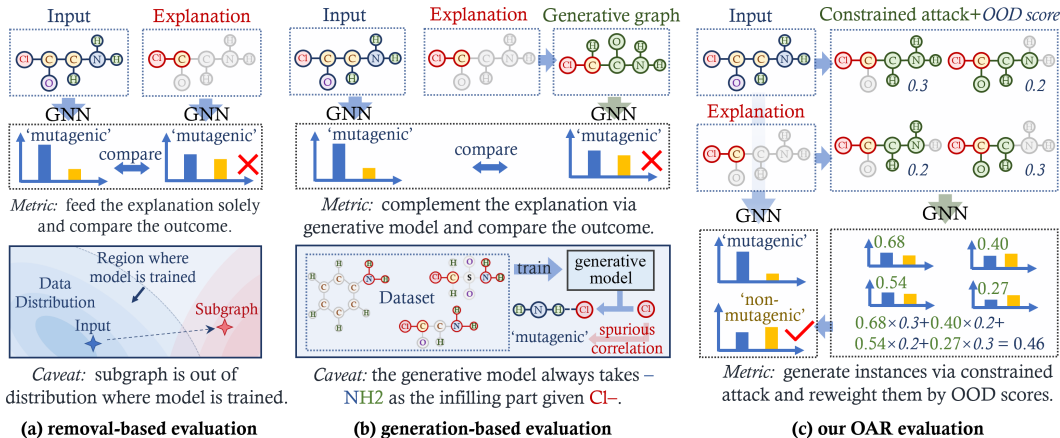


Figure 1: Pipelines and flaws of different evaluation methods. In the “Input” graph, $-\text{NH}_2$ is considered as the ground truth explanation for its mutagenicity. Best viewed in color.

input graph”. Despite the prevalence, these removal-based metrics usually come with the caveat of the out-of-distribution (OOD) issue [14, 13]. Specifically, as the after-removal subgraphs are likely to lie off the distribution of full graphs [15, 16], the GNN is forced to handle these off-manifold inputs and easily gets erroneous predictions [17, 18]. Take Figure 1 (a) as an example. For the full molecular graph, the GNN classifies it as “mutagenic”, which is reasonable due to the presence of mutagenic $-\text{NH}_2$ group; whereas, when taking the subgraph, *i.e.*, non-mutagenic C-Cl group solely as the input, the GNN surprisingly maintains its output “mutagenic”. Clearly, the prediction on the explanatory subgraph might be skeptical, which easily deteriorates the faithfulness of the removal-based evaluations.

In sight of this, recent efforts [18, 17] are beginning to mitigate the OOD issue via the *Generation-based* metrics. Instead of directly feeding the to-be-evaluated subgraph into the target GNN, they use a generative model [19, 20] to imagine and generate a new full graph conditioned on the subgraph. These methods believe that the generation process could infill the subgraph and pull it closer to the original graph distribution. As Figure 1 (b) shows, comparing the predictions on this new graph and the original graph could be the surrogate evaluation. While intuitively appealing, the generative models easily inherit the data bias and inject it into the infilling part. Considering Figure 1 (b) again, since molecules in the Mutagenicity dataset comprising non-mutagenic chloride ($-\text{Cl}$) always carry amino ($-\text{NH}_2$), a generative model is prone to capture this occurrence bias and tend to infill $-\text{NH}_2$ with the $-\text{Cl}$ -involved subgraphs. This bias not only exerts on the generations but also makes the evaluation inconsistent with the GNN behavior: in the generative model, $-\text{Cl}$ is assigned with more “mutagenic” scores as it usually accompanies the mutagenic partner $-\text{NH}_2$; in contrast, the GNN finds no “mutagenic” clues in $-\text{Cl}$. In a nutshell, the generation-based metrics show respect to the data distribution somehow but could be inconsistent with GNNs’ behavior and lose control of the infilling part.

Scrutinizing these removal- and generation-based metrics (as summarized in Figure 1), we naturally raise a question: “Can we design a metric that respects the data distribution and GNN behavior simultaneously?” To this end, we draw inspiration from adversarial robustness [21, 22] and propose a new evaluation framework, OAR (OOD-resistant Adversarial Robustness), to help reliably assess the explanations. As shown in Figure 1 (c), OAR encapsulates two components: constrained attack and OOD reweighting, which respect the GNN behavior and data distribution, respectively. Specifically:

- Intuitively, perturbations on label-irrelevant features should be ineffective to the GNN prediction, while those on label-relevant features are supposed to be impactful and destructive to the prediction [22, 21]. Hence, for the original input graph, the attack model performs perturbations constrained on the complementary part of its explanation. This perturbing game aims to naturally take control of the “infilling” process, making the explanatory subgraph less influenced by its infilling.
- Having obtained a set of perturbed graphs, the reweighting component estimates the OOD score of each perturbed graph, which reflects the degree of distribution shift from the original data

distribution. Then, we feed these graphs into GNN and reweight their predictions with OOD scores. The sum of weighted predictions can quantitatively evaluate the importance of the target subgraph.

We validate the effectiveness of OAR in evaluation tasks across various start-of-the-art explanation methods, datasets, and backbone GNN models. OAR has manifested surprising consistency with the metrics like *Precision*, *Recall* and *Human Supervision*. Furthermore, to better generalize to the large datasets, we also provide a Simplified version of OAR (SimOAR) achieving significant improvements in computational efficiency at the expense of a small amount of performance degradation. Our main contributions can be summarized as follows:

- We propose a novel metric, OAR for evaluating GNNs’ explainability, which tries to resolve the limitation of current removal- and generative-based evaluations by taking both data distribution and GNN behavior into account (Section 2.2).
- We provide a simplified version of OAR, SimOAR for better generalization to the evaluation tasks involving large datasets, which greatly shortens the execution time while only sacrificing a small amount of performance (Section 2.3).
- Experimental results demonstrate that our OAR/SimOAR outperforms the current evaluation metrics by a large margin, and further validate the high efficiency of SimOAR (Section 3).

2 Methodology

In this section, we propose an evaluation method for the explainability of GNNs from the perspective of adversarial robustness. We start with the notation of GNNs and its explainability in Section 2.1. After that, we detail our evaluation metric, OAR via three progressive steps (Section 2.2). In Section 2.3, we provide a simplified version of OAR called SimOAR for applications demanding more efficient execution.

2.1 Problem Formulation

Graph neural networks (GNNs). GNNs have achieved remarkable success due to their powerful representation ability. Without loss of generality, we focus on the graph classification task in this work: a well-trained GNN model f takes a graph \mathcal{G} as the input and outputs its probabilities \mathbf{y} over classes $\{1, \dots, C\}$, i.e., $\mathbf{y} = f(\mathcal{G}) \in \mathbb{R}^C$. Typically, \mathcal{G} is an undirected graph involving the node set \mathcal{V} and the edge set \mathcal{E} . We first introduce the feature of node $v_i \in \mathcal{V}$ as a d -dimensional vector and collect the features of all nodes into $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$. Then we define an adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ to describe graph topology, where $A_{uv} = 1$ if the edge connecting nodes u and v exists, i.e., $(u, v) \in \mathcal{E}$, otherwise $A_{uv} = 0$. Based on these, \mathcal{G} can be alternatively represented as $\mathcal{G} = (\mathbf{A}, \mathbf{X})$.

Explainability for GNNs. Upon the GNN model f , explanation techniques of GNNs generally study the underlying relationships between their outputs \mathbf{y} and inputs \mathcal{G} . They focus on explainability w.r.t. input features, aiming to answer “Which parts of the input graph contribute most to the model prediction?”. Towards this end, explainers typically assign an importance score to each input feature (i.e., node v_i or edge (v_i, v_j)) to trace their contributions. Then they select the salient part (e.g., a subset of nodes or edges with top contributions) as the explanatory subgraph \mathcal{G}_s and delete the complementary part $\mathcal{G}_{\bar{s}} = \mathcal{G} \setminus \mathcal{G}_s$. We formulate the explanation method as h and yield the above process as $\mathcal{G}_s = h(\mathcal{G}, f)$.

2.2 OOD-Resistant Adversarial Robustness

Retrospecting the removal-based and generation-based evaluations, we emphasize that both these classes come with some inherent limitations. Specifically, Removal-based metrics pay less heed to the data distribution thus forcing GNNs to handle off-manifold instances, while generation-based metrics are inconsistent with GNN behavior and lose control of infilling part. Fortunately, in this section, we claim that it is possible to get the best of and avoid the pitfalls of both worlds — removal-based and generation-based metrics — by taking both GNN behavior and data distribution into account.

To meet these challenges, we elaborate our evaluation metric, OOD-resistant adversarial robustness (OAR) via three progressive steps: **in the first step**, we formulate the adversarial robustness tailored for GNNs explanations, which naturally conforms to the GNN behavior; **in the second step**, we introduce a tractable and easy-to-implement objective of above adversarial robustness; **in the third step**, we introduce an elaborate OOD reweighting block which confines the overall evaluation process to the original data distribution.

STEP 1: Formulation of Adversarial Robustness. We prioritize the introduction of *adversarial robustness* in machine learning [22, 21] that motivates our method. Concretely, given a machine

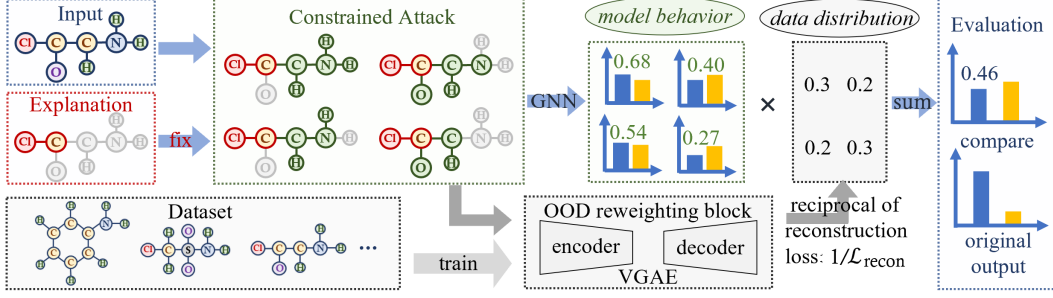


Figure 2: The pipeline of OAR, which takes both model behavior and data distribution into account.

learning model, an input x and a subset of the input $x_s \subseteq x$, the *adversarial robustness* of x_s denotes the minimum perturbation leading to the wrong prediction, on the condition that perturbation is only imposed on x_s [22]. Inspired by this idea, we define the adversarial robustness of GNN explanation \mathcal{G}_s , and formulate it as the minimum adversarial perturbation δ on the structure of complementary subgraph $\mathcal{G}_{\bar{s}}$. More formally,

Definition 1. Given a GNN model f , an input graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ with prediction \mathbf{y} and explanation \mathcal{G}_s , suppose that $\mathcal{G}' = (\mathbf{A}', \mathbf{X}')$ is the graph generated by adding and deleting edges in \mathcal{G} , the adversarial robustness δ of explanation \mathcal{G}_s is defined as:

$$\delta_{\mathcal{G}_s} = \min_{\mathbf{A}'} \sum_{u \in \mathcal{V}} \sum_{v \in \mathcal{V} \setminus u} |A_{uv} - A'_{uv}|$$

$$\text{s.t. } \arg \max_i f(\mathcal{G}')_i \neq \arg \max_i \mathbf{y}_i, \quad \sum_{u \in \mathcal{V}_s} \sum_{v \in \mathcal{V}_s \setminus u} |A_{uv} - A'_{uv}| = 0, \quad (1)$$

where \mathcal{V} and \mathcal{V}_s are the node sets of \mathcal{G} and \mathcal{G}_s , respectively.

Definition 1 identifies the quality of explanation \mathcal{G}_s as the difficulty of reversing the prediction by perturbing features not belonging to \mathcal{G}_s solely. That is, when \mathcal{G}_s is fixed, the more difficult to fool the model by perturbing its complementary, the more important \mathcal{G}_s is. The key intuition behind this inference is: if an explanation comprises most of the label-relevant features, it is conceivably hard to change the prediction by manipulating the remaining features which are label-irrelevant. Thus, according to Definition 1 we can find that: a good explanation would yield high adversarial robustness δ and vice versa, which naturally conforms to the GNN behavior.

It seems that adversarial robustness is the feasible metric to evaluate the GNNs' explanations. However, there are still two matters standing in the way of its adoption: 1) Is its objective (*i.e.*, Equation (1)) tractable and easy to implement? 2) Does it respect the data distribution?

STEP 2: Finding a Tractable Objective. To answer the first question, we argue that Equation (1) is hard to realize and sometimes even intractable owing to two possible reasons:

- The primary reason is that adversarial attacks may fail to find any adversarial example, since a solution satisfying two conditions in Equation (1) simultaneously may not exist. In other words, if the explanation \mathcal{G}_s is precise enough, it is almost impossible to reverse the prediction via manipulating features in the complementary part $\mathcal{G}_{\bar{s}}$ which are mainly label-irrelevant.
- It is notoriously hard to search for the minimum adversarial perturbation δ in most cases. Current attack methods [23, 24] typically turn to find an alternative sub-optimal solution. Thus, leveraging these methods could introduce additional bias and threatens the fairness of evaluation.

To address these issues and make the evaluation objective tractable and easy to implement, we first formulate the inference derived from Definition 1 :

Proposition 1. When the high quality explanation \mathcal{G}_s is anchored (fixed), perturbations restricted to the complementary part $\mathcal{G}_{\bar{s}}$ have a weak influence on the model prediction.

While Definition 1 evaluates the explanations via the adversarial robustness δ , Proposition 1 indicates a more straightforward way to the tractable evaluation objective from its dual perspective. To be more specific, Definition 1 quantifies perturbation on $\mathcal{G}_{\bar{s}}$ causing change of prediction; conversely, Proposition 1 quantifies change of prediction caused by perturbation on $\mathcal{G}_{\bar{s}}$. More formally,

Definition 2. Given a GNN model f , an input graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ with prediction \mathbf{y} and explanation \mathcal{G}_s , suppose that $\mathcal{G}' = (\mathbf{A}', \mathbf{X}')$ is the graph generated by adding and deleting edges in \mathcal{G} , the approximate adversarial robustness δ^* of \mathcal{G}_s is defined as:

$$\begin{aligned} \delta_{\mathcal{G}_s}^* &= \mathbb{E}_{\mathcal{G}'} (f(\mathcal{G}')_c - \mathbf{y}_c) \\ \text{s.t. } c &= \arg \max_i \mathbf{y}_i, \quad \sum_{u \in \mathcal{V}_s} \sum_{v \in \mathcal{V}_s \setminus u} |A_{uv} - A'_{uv}| = 0, \end{aligned} \quad (2)$$

where \mathcal{V}_s refers to the node set of \mathcal{G}_s ; c denotes the predicted class of \mathcal{G} by model f ; $f(\mathcal{G}')_c$ represents the probability value of $f(\mathcal{G}')$ for the given class c .

Stemmed from *Definition 2*, the evaluation method to quantify the adversarial robustness of GNN explanations is more explicit and computationally convenient. As shown in Figure 2, for the to-be-evaluated subgraph \mathcal{G}_s (i.e., C-Cl in red dotted box), we anchor it and randomly perturb the remain part $\mathcal{G}_{\bar{s}}$ to get graphs \mathcal{G}' (i.e., molecules in green dotted box). Then we compare the expectation of the prediction of \mathcal{G}' with the prediction of the original graph \mathcal{G} . If they are close, most features in $\mathcal{G}_{\bar{s}}$ must be label-irrelevant. Hence, we can assign high quality for the explanation \mathcal{G}_s .

So far, there is only one question left: how to ensure that the aforementioned evaluation process respects the data distribution?

STEP 3: OOD Reweighting Block Tailored for GNNs. Before elaborating our OOD block, we first retrospect that in most scenarios, a tiny perturbation would not induce large distribution shifts along the input space, thanks to the approximate continuity of input features (e.g., image pixels in *computer vision*). Unfortunately, the structural features of GNNs’ inputs — adjacency matrix comprising of 0s and 1s — are discrete thus only one perturbation (e.g., adding or deleting an edge) could induce large distribution shifts, and further violate the underlying properties, such as node degree distribution [25], graph size distribution [26] and domain-specific constraints [27].

Thus, it is pivotal to construct an OOD reweighting block for assessing whether the generated graph \mathcal{G}' deviates from the data manifold. This block is expected to assign an “OOD score” — the degree of distribution shift between \mathcal{G}' and original graph \mathcal{G} — to each \mathcal{G}' . However, it is non-trivial to quantify the degree of OOD [28]. Inspired by the great success of *graph anomaly detection* [29, 30, 31, 32, 33], we treat the OOD instance as the “anomaly” since it is isolated from the original data distribution, and naturally employ the common usage module of anomaly detection — *variational graph auto-encoder* (VGAE) [19] containing an encoder and a decoder — to instantiate our OOD reweighting block. Additionally, the great success of diffusion generative models has been recognized recently [34, 35], and thus this domain is deferred for future investigation.

Concretely, as shown in Figure 2, the preparation for evaluations is training the VGAE model on the dataset \mathcal{D} where input graph \mathcal{G} is sampled. After that, we can leverage the reconstruction loss to approximate the degree of OOD for any generated instance \mathcal{G}' . To be more specific:

- Given $\mathcal{G}' = (\mathbf{A}', \mathbf{X}')$, the **encoder** first learns a latent matrix \mathbf{Z} according to \mathbf{A}' and \mathbf{X}' , where row \mathbf{z}_i corresponds to the node v'_i in \mathcal{G}' . Note that \mathbf{z}_i is assumed to follow the independent normal distributions with expectation $\boldsymbol{\mu}_i$ and variance $\boldsymbol{\sigma}_i^2$. Formally:

$$q(\mathbf{Z} | \mathbf{A}', \mathbf{X}') = \prod_{i=1}^{|\mathcal{V}'|} q(\mathbf{z}_i | \mathbf{A}', \mathbf{X}') = \prod_{i=1}^{|\mathcal{V}'|} \mathcal{N}(\mathbf{z}_i | \boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2)), \quad (3)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are parameterized by two two-layer GCNs [36] called $\text{GCN}_{\boldsymbol{\mu}}$ and $\text{GCN}_{\boldsymbol{\sigma}}$.

- Then, the **decoder** recovers the adjacency matrix \mathbf{A}' based on \mathbf{Z} :

$$p(\mathbf{A}' | \mathbf{Z}) = \prod_{i=1}^{|\mathcal{V}'|} \prod_{j=1}^{|\mathcal{V}'|} p(A'_{ij} | \mathbf{z}_i, \mathbf{z}_j), \quad (4)$$

$$\text{with } p(A'_{ij} = 1 | \mathbf{z}_i, \mathbf{z}_j) = \sigma(\mathbf{z}_i^\top \mathbf{z}_j),$$

where $\sigma(\cdot)$ is the logistic sigmoid function.

- The **OOD score** of \mathcal{G}' is given by the normalized reciprocal of the reconstruction loss $\mathcal{L}_{recon}(\mathcal{G}')$,

$$\mathcal{L}_{recon}(\mathcal{G}') = -\log p(\mathbf{A}' | \mathbf{Z}), \text{ with } \mathbf{Z} = \boldsymbol{\mu} = \text{GCN}_{\boldsymbol{\mu}}(\mathbf{A}', \mathbf{X}'). \quad (5)$$

Since VGAE is trained on the dataset \mathcal{D} , \mathcal{G}' straying far from the data distribution of \mathcal{D} would get the high reconstruction loss \mathcal{L}_{recon} . Thus, as the reciprocal of \mathcal{L}_{recon} , the OOD score of \mathcal{G}' must be low. Conversely, if \mathcal{G}' is in distribution, it would gain a high OOD score because it is easy to be reconstructed. Based on this, our OOD block can mitigate the impact of OOD instances. Specifically, the OOD score is utilized as the weight of each prediction when calculating the expectation of the generated graph’s prediction. This allows for the marginalization of instances with low OOD scores, as shown in the gray dotted box of Figure 2.

Overall evaluation process. As the last piece of the OAR puzzle, *i.e.*, OOD reweighting block has been instantiated, let’s revisit Figure 2 and summarize the overall process of OAR:

1. Before we evaluate the explanatory subgraph \mathcal{G}_s , the OOD reweighting block (*i.e.*, VGAE) is trained on the dataset \mathcal{D} where input graph \mathcal{G} is sampled.
2. Then, we fix the \mathcal{G}_s and randomly perturb the structure of the complementary part $\mathcal{G}_{\bar{s}}$ to get \mathcal{G}' .
3. Each \mathcal{G}' is fed into GNN f and VGAE simultaneously to audit prediction and OOD score. Both GNN’s behavior and data distribution are taken into consideration in this step.
4. At last, according to the predictions and their weights (*i.e.*, OOD scores), we calculate the weighted average of the generated graphs’ predictions. The closer this average is to the original prediction of \mathcal{G} , the higher the quality of the explanation \mathcal{G}_s is.

The pseudocode and the tricks to expedite computations are detailed in Appendix A.

2.3 A simplified version of OAR

To better generalize to large datasets and reduce the computational complexity, we provide a simplified version of OAR called SimOAR in this section. Compared with OAR, SimOAR achieves a significant improvement in computational efficiency at the expense of a small amount of performance degradation. Concretely, SimOAR is mainly motivated by three empirical inferences after executing OAR:

- The most time-consuming part of OAR is its preparatory work, *i.e.*, training OOD reweighting block, especially for large datasets. For example, on the dataset *MNIST superpixels* [37] containing 70,000 graphs, the converged process of VGAE occupies 93.7% of OAR’s execution time.
- In the course of generating \mathcal{G}' , the number of perturbation operations is roughly proportional to the degree of distribution shift given by the OOD block. For example, the graph \mathcal{G}'_1 created via deleting one edge typically gets a higher OOD score than the graph \mathcal{G}'_2 created via deleting five edges.
- For two generated graphs generated via the same perturbation times, they generally get similar reconstruction losses and are assigned similar OOD scores.

Based on these, to expedite computations and simplify the OAR, we deactivate the OOD reweighting block (*i.e.*, deleting all the sketches in gray dotted boxes in Figure 2) in OAR. As compensation for data distribution, we restrict the ratio of the number of perturbations to the number of edges in the original graph \mathcal{G} to a pre-defined minor value R . Since the generated graphs typically share similar reconstruction losses and OOD scores while R is fixed, we directly calculate their average prediction to approximate their expected prediction. The pseudocode and more implementation details of SimOAR are provided in Appendix A.

It is worth noting that despite the potential existence of a few generated graphs \mathcal{G}' in SimOAR that fall outside the distribution, the performance of SimOAR still significantly surpasses that of current evaluation methods *w.r.t* consistency with both metrics based on ground truth and human supervision. Hence, in light of the efficiency of the SimOAR, we strongly advocate for its adoption as a predominant alternative to prevalent removal-based evaluation metrics. At the heart of SimOAR – and a **central thesis** of this paper – is the perspective that, during evaluation, rather than deleting all non-explanatory nodes and then gauging the resultant output variations, it is more insightful to **randomly delete a portion of the non-explanatory nodes multiple times and then gauge the average output variations**.

3 Experiments

We present empirical results to demonstrate the effectiveness of our proposed methods OAR and SimOAR. The experiments aim to investigate the following research questions:

- **RQ1:** How is the evaluation quality of OAR and SimOAR compared to that of existing metrics?
- **RQ2:** How is the generalization of OAR and SimOAR compared to that of existing metrics?
- **RQ3:** What is the impact of the designs (*e.g.*, the OOD reweighting block) on the evaluations?

Table 1: Overall evaluation scores and rankings of explainers under different evaluation methods. Symbol (\cdot) indicates the rank of explainers. Our methods, *i.e.*, OAR and SimOAR are bold and the best-performing methods are denoted with the superscript asterisk.

		SA	GradCAM	GNExplainer	PGExplainer	CXPlain	ReFine	$\tau \uparrow$
BA3	Recall	88.12 \pm 0.00 ⁽¹⁾	84.53 \pm 0.00 ⁽²⁾	76.83 \pm 4.64 ⁽³⁾	65.32 \pm 5.21 ⁽⁵⁾	54.77 \pm 4.42 ⁽⁶⁾	72.90 \pm 3.72 ⁽⁴⁾	-
	RM	35.39 \pm 0.00 ⁽³⁾	37.67 \pm 0.00 ⁽⁴⁾	43.32 \pm 1.97 ⁽¹⁾	29.25 \pm 2.26 ⁽⁵⁾	27.78 \pm 0.96 ⁽⁶⁾	41.24 \pm 1.55 ⁽²⁾	0.73
	DSE	43.08 \pm 2.31 ⁽¹⁾	41.38 \pm 1.75 ⁽²⁾	22.25 \pm 2.14 ⁽⁶⁾	37.92 \pm 2.52 ⁽³⁾	24.60 \pm 3.57 ⁽⁵⁾	29.31 \pm 2.96 ⁽⁴⁾	0.73
	OAR	93.12 \pm 4.60 ⁽¹⁾	86.20 \pm 3.76 ⁽²⁾	80.19 \pm 1.68 ⁽³⁾	65.48 \pm 3.75 ⁽⁵⁾	59.69 \pm 2.98 ⁽⁶⁾	71.02 \pm 4.45 ⁽⁴⁾	1.00*
	SimOAR	84.39 \pm 5.70 ⁽¹⁾	83.44 \pm 4.81 ⁽²⁾	62.52 \pm 2.25 ⁽³⁾	50.02 \pm 2.87 ⁽⁶⁾	55.49 \pm 4.22 ⁽⁵⁾	60.42 \pm 3.32 ⁽⁴⁾	0.93
TR3	Recall	82.08 \pm 0.00 ⁽¹⁾	77.00 \pm 0.00 ⁽²⁾	60.09 \pm 4.97 ⁽⁴⁾	55.85 \pm 4.70 ⁽⁵⁾	44.39 \pm 5.57 ⁽⁶⁾	74.19 \pm 3.30 ⁽³⁾	-
	RM	55.08 \pm 0.00 ⁽³⁾	51.15 \pm 0.00 ⁽⁴⁾	79.08 \pm 4.31 ⁽¹⁾	50.45 \pm 2.04 ⁽⁵⁾	47.72 \pm 3.82 ⁽⁶⁾	64.60 \pm 2.27 ⁽²⁾	0.67
	DSE	48.51 \pm 1.00 ⁽¹⁾	37.32 \pm 2.35 ⁽⁴⁾	44.82 \pm 2.90 ⁽²⁾	33.71 \pm 3.72 ⁽⁶⁾	35.65 \pm 1.92 ⁽⁵⁾	39.49 \pm 3.31 ⁽³⁾	0.73
	OAR	95.23 \pm 3.75 ⁽¹⁾	87.51 \pm 6.65 ⁽²⁾	72.63 \pm 5.89 ⁽³⁾	59.06 \pm 4.83 ⁽⁵⁾	51.61 \pm 3.14 ⁽⁶⁾	63.82 \pm 5.43 ⁽⁴⁾	0.93*
	SimOAR	88.45 \pm 6.04 ⁽¹⁾	76.37 \pm 4.54 ⁽²⁾	53.54 \pm 4.46 ⁽⁶⁾	68.58 \pm 5.80 ⁽⁴⁾	62.98 \pm 4.00 ⁽⁵⁾	75.59 \pm 4.27 ⁽³⁾	0.86
MNIST-sp	Recall	43.98 \pm 0.00 ⁽³⁾	44.39 \pm 0.00 ⁽⁴⁾	54.63 \pm 0.96 ⁽¹⁾	30.13 \pm 1.42 ⁽⁶⁾	38.96 \pm 2.62 ⁽⁵⁾	47.88 \pm 1.60 ⁽²⁾	-
	RM	21.34 \pm 0.00 ⁽⁴⁾	19.10 \pm 0.00 ⁽⁵⁾	22.23 \pm 1.02 ⁽³⁾	25.04 \pm 0.35 ⁽²⁾	27.15 \pm 0.69 ⁽¹⁾	17.58 \pm 0.50 ⁽⁶⁾	0.33
	DSE	30.37 \pm 4.06 ⁽²⁾	29.19 \pm 2.19 ⁽³⁾	14.03 \pm 1.77 ⁽⁶⁾	28.45 \pm 2.65 ⁽¹⁾	22.95 \pm 2.44 ⁽⁴⁾	21.32 \pm 1.29 ⁽⁵⁾	0.20
	OAR	66.28 \pm 2.46 ⁽³⁾	64.18 \pm 5.25 ⁽⁴⁾	82.22 \pm 4.13 ⁽¹⁾	63.88 \pm 3.45 ⁽⁵⁾	51.37 \pm 1.76 ⁽⁶⁾	75.43 \pm 4.84 ⁽²⁾	0.93*
	SimOAR	54.72 \pm 3.84 ⁽⁴⁾	69.86 \pm 2.80 ⁽³⁾	79.69 \pm 3.53 ⁽¹⁾	33.27 \pm 2.04 ⁽⁶⁾	52.93 \pm 3.26 ⁽⁵⁾	76.40 \pm 2.24 ⁽²⁾	0.93*
MUTAG	Recall	41.12 \pm 0.00 ⁽⁶⁾	44.44 \pm 0.00 ⁽⁵⁾	55.95 \pm 4.94 ⁽⁴⁾	71.22 \pm 2.54 ⁽²⁾	64.65 \pm 1.50 ⁽³⁾	77.73 \pm 3.67 ⁽¹⁾	-
	RM	75.60 \pm 0.00 ⁽⁵⁾	77.39 \pm 0.00 ⁽⁶⁾	82.32 \pm 4.37 ⁽²⁾	87.11 \pm 2.59 ⁽¹⁾	76.49 \pm 3.51 ⁽⁴⁾	81.76 \pm 2.64 ⁽³⁾	0.73
	DSE	38.03 \pm 3.90 ⁽⁴⁾	32.36 \pm 2.40 ⁽⁵⁾	40.28 \pm 1.32 ⁽³⁾	49.18 \pm 2.57 ⁽¹⁾	29.06 \pm 1.44 ⁽⁶⁾	43.19 \pm 2.14 ⁽²⁾	0.67
	OAR	52.10 \pm 3.58 ⁽⁶⁾	53.19 \pm 3.87 ⁽⁵⁾	63.35 \pm 2.50 ⁽⁴⁾	88.46 \pm 2.43 ⁽²⁾	67.19 \pm 2.08 ⁽³⁾	92.81 \pm 5.24 ⁽¹⁾	1.00*
	SimOAR	75.45 \pm 2.51 ⁽⁵⁾	71.07 \pm 5.40 ⁽⁶⁾	85.65 \pm 4.00 ⁽²⁾	81.48 \pm 2.98 ⁽³⁾	76.32 \pm 3.88 ⁽⁴⁾	89.40 \pm 5.06 ⁽¹⁾	0.80

3.1 Experimental settings

To evaluate the effectiveness of our method, we utilize four benchmark datasets: BA3 [38], TR3 [17], Mutagenicity [39, 40], and MNIST-sp [37], which are publicly accessible and vary in terms of domain and size. Moreover, to generate the explanations of the graphs in the datasets mentioned above, we adopt several state-of-the-art post-hoc explanation methods, *i.e.*, SA [10], GradCAM [41], GNExplainer [11], PGExplainer [42], CXPlain [43] and ReFine [38]. The prevailing metrics — removal-based evaluation (RM for short) and generation-based evaluation, *i.e.*, DSE [17] — are selected as the baselines. Detailed experimental details can be found in Appendix B.

3.2 Measurement metric

We elaborate on the measurement metric of existing evaluation methods in this part since how to fairly define the quality of an evaluation method is critical to our research.

Ground-truth explanations. We first follow the prior studies [11, 44, 42] and treat the subgraphs coherent to data generation procedure or human knowledge as ground truth. Although ground truth might not conform to the decision-making process exactly, they contain sufficient discriminative information to help justify the quality of explanations. Moreover, it’s worth emphasizing again that our method does not depend on ground truth, and gathering ground truth is only for fairly comparison.

Consistency with metric based on ground-truth explanations. Specifically, given a to-be-evaluated subgraph \mathcal{G}_s and its corresponding ground-truth explanation \mathcal{G}_s^{GT} , we use **Recall** as the gold evaluation metric defined as $\text{Recall}(\mathcal{G}_s) = |\mathcal{E}_s \cap \mathcal{E}_s^{GT}| / |\mathcal{E}_s^{GT}|$, where \mathcal{E}_s and \mathcal{E}_s^{GT} are the edge set of \mathcal{G}_s and \mathcal{G}_s^{GT} ; $|\cdot|$ denotes the cardinal function of set. Hence, for any evaluation method, we can calculate its consistency with Recall to quantify its performance via Kendall correlation τ [45] defined as:

$$\tau \left(\{r^i\}_{i=1}^n, \{s^i\}_{i=1}^n \right) = \frac{2}{n(n+1)} \sum_{i < j} I(\text{sgn}(r^i - r^j) = \text{sgn}(s^i - s^j)), \quad (6)$$

where $\{r^i\}_{i=1}^n$ and $\{s^i\}_{i=1}^n$ are a pair of Recall values and evaluation scores; $\text{sgn}(\cdot)$ is the sign function and $I(\cdot)$ is the indicator function. The bigger τ is, the higher the evaluation scores are consistent with Recall values, and thus the better the evaluation method is.

Consistency with human intuition. The consistency between evaluation results and human intuition is also an important reference. In view of the high subjectivity of human intuition, we organized a large-scale user study engaging 100 volunteers. Results are shown in Appendix C.

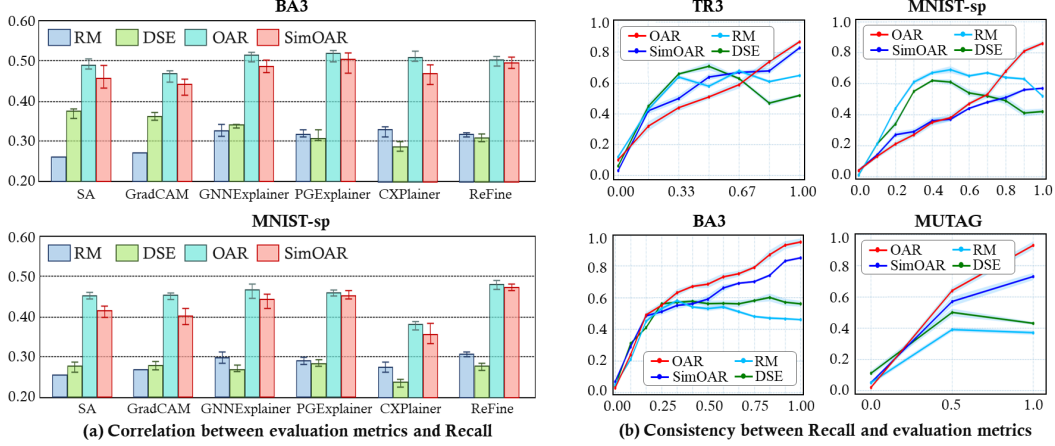


Figure 3: The performance of various evaluation metrics. (a) Correlation between metrics and Recall across various backbone explainers, where the vertical axis represents the normalized Kendall rank correlation. (b) Consistency between Recall and the scores provided by metrics. The more monotonously increasing the curve is, the better the evaluation metric is. Best viewed in color.

Limited by space, we further exhibit and discuss more detailed implementations and results in Appendix B, including but not limited to the generalization of OAR involving the evaluation of the post-hoc explanations for node classification task and the inherent explanations.

3.3 Study of explanation evaluation (RQ1)

As a preparation for the experiments, we first collect the explanatory subgraphs $\{\mathcal{G}_s^i\}$ for a set of graphs $\{\mathcal{G}_s^i\}_{i=1}^N$ and the corresponding well-trained GNN model f . We denote the evaluation score on \mathcal{G}_s^i based on RM, DSE and our OAR and SimOAR by s_{RM}^i , s_{DSE}^i , s_{OAR}^i and s_{SimOAR}^i , respectively. For more faithful comparison, we present both the explainer-level correlation defined as $\tau_* = \tau \left(\left\{ \frac{1}{N} \sum_{i=1}^N \text{Recall}(\mathcal{G}_s^{i,h}) \right\}_{h \in \mathcal{H}}, \left\{ \frac{1}{N} \sum_{i=1}^N s_*^i \right\}_{h \in \mathcal{H}} \right)$ and the instance-level correlation defined as $\tau_* = \tau \left(\left\{ \text{Recall}(\mathcal{G}_s^i) \right\}_{i=1}^N, \left\{ s_*^i \right\}_{i=1}^N \right)$, where $*$ can be RM, DSE, OAR, and SimOAR; $\mathcal{G}_s^{i,h}$ means the subgraph \mathcal{G}_s^i is extracted by explainer h , \mathcal{H} is the set of explainers. The explainer-level results, under all evaluation methods, on all datasets, are presented in Table 1. Moreover, considering the instance-level results share a similar tendency, we presented the representative results on BA3 and MNIST-sp in Figure 3 (a). According to Table 1 and Figure 3 (a) we can find that:

- **Observation 1: OAR outperforms other methods in all cases.** Substantially, Kendall rank correlation greatly improves after leveraging the paradigm of adversarial robustness. The most notable case is the explainers' rank on BA3 and MUTAG, where $\tau^* = 1.00$ achieves a tremendous increase from the RM and the DSE. It demonstrates the effectiveness and universality of OAR/SimOAR and verifies that OAR/SimOAR can be leveraged to boost the quality of evaluations.
- **Observation 2: SimOAR performs a little worse than OAR, but still significantly improves over the strongest baselines.** To be more specific, the average score of SimOAR is 7.83% less than that of OAR, but still, 42.62% higher than RM and 37.45% higher than DSE. It demonstrates that SimOAR is adequate for the task of explanation evaluations in most cases.

Further analysis of the results presented in Table 1 reveals that:

- **Observation 3: OAR/SimOAR presents a more fair and faithful comparison among explainers.** The rankings provided by the OAR and SimOAR are highly consistent (*i.e.*, $\tau^* = 0.928$ on average) with the references, while the removal- and generation-based rankings unsurprisingly pale by comparison. These empirical results give us the courage to leverage OAR and SimOAR to evaluate emerging explanation methods in the future.

3.4 Study of generalization (RQ2)

Although the experiment provided in 3.3 is detailed and fair, we contend that the generalization of OAR and SimOAR is still unexamined. Specifically, a specific explainer always has a preference

for certain patterns, and thus the explanatory subgraphs extracted by it often have similar structures. Therefore, the experimental results based on limited explainers may not generalize well to other existing or future explainers, especially those based on different lines of thought.

As it is impractical for us to cover all these explainers, we resort to directly generalizing the to-be-evaluated subgraphs. That is, we make use of fake explanatory subgraphs which are randomly sampled from the full graph. The detailed sampling algorithm and settings can be found in Appendix C. In these settings, the best case is that the evaluation score is monotonically increasing *w.r.t.* the Recall level, which indeed reaches the best consistency. Average normalized scores under all evaluation methods are shown in Figure 3 (b), which indicate that:

- **Observation 4: OAR/SimOAR greatly improves the consistency between evaluation scores and Recalls, which indicates that our method has tremendous potential to perform well on other explainers.** Conversely, removal- and generation-based methods are negatively correlated with the importance involving the set of to-be-evaluated subgraphs which get high Recalls.

3.5 Study of designs (RQ3)

Effectiveness of OOD block. We first focus on the effectiveness of the OOD block. The most immediate impact of OOD block on OAR can be estimated by comparing the performance of OAR and SimOAR, which can be also deemed as the ablation experiments. Hence, we turn to qualitatively analyze the OOD block via some case studies shown in Figure 4, where all the OOD scores belonging to the same datasets are normalized to the range of 0 to 1. Based on the information conveyed in Figure 4, the following observation can be made:

- **Observation 5: OOD block can assign the lower weights to the subgraph which violent the underlying properties of the full graph.** For example, graph properties of chemical molecules, such as the valency rules, impose some constraints on syntactically valid molecules. Hence, the invalid molecular subgraphs, which destroy the integrity of the carbon ring by simply removing some bonds (edges) or atoms (nodes) are assigned low scores by the OOD block.

Time complexity. To further explore the efficiency of our evaluation method and the designed module in it, we count the running time of the evaluation process on every single graph and average the time over the entire test set to obtain per-graph time consumption. The comparison is provided in Table 2. According to Table 2 we can find that:

- **Observation 6: SimOAR greatly reduces the execution time.** Concretely, the execution speed has nearly doubled after leveraging the metric of SimOAR. This significant improvement in efficiency corresponds to the original intention of SimOAR and verifies the success of SimOAR’s design.

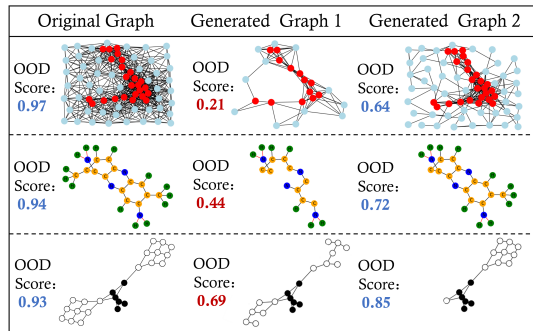


Figure 4: Case studies for OOD reweighting block with the graphs randomly selected from datasets MNIST-sp, MUTAG, and BA3 arranged from top to bottom. Best viewed in color.

Table 2: Per-graph time consumption. (ms)

	BA3	TR3	MNIST-sp	MUTAG
RM	0.11±0.02	0.10±0.01	0.15±0.03	0.13±0.01
DSE	1.73±0.27	1.60±0.14	2.12±0.39	1.88±0.25
OAR	1.16±0.05	1.32±0.06	1.77±0.10	1.78±0.03
SimOAR	0.84±0.07	0.65±0.08	1.03±0.07	0.96±0.06

4 Related Work & Further Discussion

OAR & Contemporary Evaluation Metrics. Apart from the qualitative evaluation methods based on human intuition, recent literature categorizes quantitative metrics into four primary categories: accuracy, faithfulness, stability, and fairness [46, 47, 48, 49, 50, 51]. Notably, Precision and Recall metrics align with the accuracy category, while our proposed OAR and SimOAR fall under the faithfulness category. Among these methods, the most recently proposed faithfulness metric is GEF [48], which however omits quantification of the distribution shift in subgraphs. Nevertheless, we

have exhibited the experimental comparison between GEF and our metrics *w.r.t* the latest dataset SHAPEGGEN [48] in Appendix B.

OOD & GNNs Explainability The OOD issue is one of the most critical challenges in the post-hoc explainability domain currently [16, 52]. To sidestep this challenge, many studies have pivoted towards the development of inherently explainable GNNs [15, 53, 54, 55]. Notwithstanding the complexity of the task, efforts such as FIDO [56] are making significant successes in addressing the OOD problem within post-hoc explanations. Concurrently, CGE [57] leverages the lottery ticket hypothesis [58, 59] to craft the cooperative explanation for both GNNs and graphs, wherein the OOD challenge is potentially mitigated through the EM algorithm. GIBE [60] delves into the intersection of the OOD issue and regularization, viewing it through the lens of information theory. Furthermore, MixupExplainer [61] and CoGE [62] navigate the OOD problem from the generation and recognition stances, respectively.

Simultaneously, the evaluation of inherent explanations encounters the same hurdles as post-hoc explanations: it’s challenging to quantify in the absence of the ground truth. Fortunately, by introducing an additionally well-trained GNN, OAR can be employed to **evaluate inherent explanations** in a similar way to evaluate post-hoc explanations. The experimental results are shown in Appendix B.

Limitations & Concerns. While we acknowledge the effectiveness of our methods, we also recognize their limitations. Concretely, despite utilizing SimOAR to expedite the evaluation process, our paradigm remains more time-intensive compared to the conventional removal-based metric. To overcome this constraint, a probable solution is summarizing the optimal number of perturbations and implementing a self-adaptive extraction module to select the perturbed features.

Furthermore, we recognize potential apprehensions regarding the migration of trust issues from the black-box GNN to the equally non-transparent VGAE. Nevertheless, we posit that the repercussions of this are substantially mitigated in our streamlined method. Specifically, SimOAR bypasses VGAE in favor of employing transparent heuristics for perturbation generation, effectively addressing the aforementioned trust concerns. It’s noteworthy that, while SimOAR’s performance may be marginally below or comparable to OAR’s, it consistently exceeds other benchmarks. This emphasizes VGAE’s restrained impact and reaffirms our recommendation of SimOAR over OAR.

5 Conclusion

In this paper, we explored the evaluation process of GNN explanations and proposed a novel evaluation metric, OOD-resistant adversarial robustness (OAR). OAR get inspiration from the notion of adversarial robustness and evaluate the quality of explanations by calculating their robustness under attack. It addresses the inherent limitations of current removal- and generation-based evaluation metrics by taking both data distribution and GNN behavior into account. For applications involving large datasets, we introduce a simplified version of OAR (SimOAR), which achieves a significant increase in computational efficiency at the cost of a small amount of performance degradation. This work represents an initial attempt to exploit evaluation metrics for post-hoc GNN explainability from the perspective of adversarial robustness and resistance to OOD.

6 Ethics Statement

This work is primarily foundational in GNN explainability, focusing on the development of the more reliable evaluation algorithm. Its primary aim is to contribute to the academic community by enhancing the understanding and implementation of evaluation process. We do not foresee any direct, immediate, or negative societal impacts stemming from the outcomes of our research.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (9227010114, 62121002) and the University Synergy Innovation Program of Anhui Province (GXXT-2022-040).

References

- [1] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *CoRR*, abs/2003.00982, 2020.

- [2] Tianyi Zhao, Yang Hu, Linda R. Valsdottir, Tianyi Zang, and Jiajie Peng. Identifying drug-target interactions based on graph convolutional network and deep neural network. *Briefings Bioinform.*, 22(2):2141–2150, 2021.
- [3] Zhiwei Guo and Heng Wang. A deep graph neural network-based mechanism for social recommendations. *IEEE Trans. Ind. Informatics*, 17(4):2776–2783, 2021.
- [4] Kun Wang, Yuxuan Liang, Pengkun Wang, Xu Wang, Pengfei Gu, Junfeng Fang, and Yang Wang. Searching lottery tickets in graph neural networks: A dual perspective. In *The Eleventh International Conference on Learning Representations*, 2023.
- [5] Minh N. Vu and My T. Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. In *NeurIPS*, 2020.
- [6] Wanyu Lin, Hao Lan, and Baochun Li. Generative causal explanations for graph neural networks. In *ICML*, volume 139, pages 6666–6679, 2021.
- [7] Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. Interpreting graph neural networks for NLP with differentiable edge masking. In *ICLR*, 2021.
- [8] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *CoRR*, abs/2012.15445, 2020.
- [9] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *ICML*, volume 139, pages 12241–12252. PMLR, 2021.
- [10] Federico Baldassarre and Hossein Azizpour. Explainability techniques for graph convolutional networks. *CoRR*, abs/1905.13686, 2019.
- [11] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*, pages 9240–9251, 2019.
- [12] Leo Breiman. Random forests. *Mach. Learn.*, 45:5–32, 2001.
- [13] Giles Hooker and Lucas Mentch. Please stop permuting features: An explanation and alternatives. *CoRR*, abs/1905.03151, 2019.
- [14] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *CoRR*, abs/2202.07987, 2022.
- [15] Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *ICML*, pages 15524–15543, 2022.
- [16] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. *CoRR*, abs/2201.12872, 2022.
- [17] Ying-Xin Wu, Xiang Wang, An Zhang, Xia Hu, Fuli Feng, Xiangnan He, and Tat-Seng Chua. Deconfounding to explanation evaluation in graph neural networks. *CoRR*, abs/2201.08802, 2022.
- [18] Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.
- [19] Thomas N. Kipf and Max Welling. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016.
- [20] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [21] Thomas Fel, Melanie Ducoffe, David Vigouroux, Rémi Cadène, Mikael Capelle, Claire Nicodeme, and Thomas Serre. Don’t lie to me! robust and efficient explainability with verified perturbation analysis. *CoRR*, abs/2202.07728, 2022.

- [22] Cheng-Yu Hsieh, Chih-Kuan Yeh, Xuanqing Liu, Pradeep Kumar Ravikumar, Seungyeon Kim, Sanjiv Kumar, and Cho-Jui Hsieh. Evaluations and methods for explanation through robustness analysis. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.
- [23] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 2847–2856, 2018.
- [24] Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.
- [25] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, pages 177–187, 2005.
- [26] Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. Size-invariant graph representations for graph classification extrapolations. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pages 837–851, 2021.
- [27] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt. Constrained graph variational autoencoders for molecule design. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7806–7815, 2018.
- [28] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. OOD-GNN: out-of-distribution generalized graph neural network. *CoRR*, abs/2112.03806, 2021.
- [29] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Quan Z. Sheng, and Hui Xiong. A comprehensive survey on graph anomaly detection with deep learning. *CoRR*, abs/2106.07178, 2021.
- [30] Yuan Gao, Xiang Wang, Xiangnan He, Zhengguang Liu, Huamin Feng, and Yongdong Zhang. Alleviating structural distribution shift in graph anomaly detection. In *WSDM*, pages 357–365. ACM, 2023.
- [31] Xian Teng, Muheng Yan, Ali Mert Ertugrul, and Yu-Ru Lin. Deep into hypersphere: Robust and unsupervised anomaly discovery in dynamic networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 2724–2730, 2018.
- [32] Yuan Gao, Xiang Wang, Xiangnan He, Zhengguang Liu, Huamin Feng, and Yongdong Zhang. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *WWW*, pages 1528–1538. ACM, 2023.
- [33] Yuan Gao, Xiang Wang, Xiangnan He, Huamin Feng, and Yong-Dong Zhang. Rumor detection with self-supervised learning on texts and social graph. *Frontiers Comput. Sci.*, 17(4):174611, 2023.
- [34] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.
- [35] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, pages 8780–8794, 2021.
- [36] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR (Poster)*, 2017.
- [37] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, pages 5425–5434, 2017.

- [38] Xiang Wang, Ying-Xin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards multi-grained explainability for graph neural networks. In *NeurIPS*, pages 18446–18458, 2021.
- [39] Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of medicinal chemistry*, 48(1):312–320, 2005.
- [40] Kaspar Riesen and Horst Bunke. IAM graph database repository for graph based pattern recognition and machine learning. In *SSPR/SPR*, pages 287–297, 2008.
- [41] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 618–626, 2017.
- [42] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In *NeurIPS*, 2020.
- [43] Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [44] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. XGNN: towards model-level explanations of graph neural networks. In *KDD*, pages 430–438, 2020.
- [45] Eulalia Szmidt and Janusz Kacprzyk. The spearman and kendall rank correlation coefficients between intuitionistic fuzzy sets. In *EUSFLAT Conf*, pages 521–528. Atlantis Press, 2011.
- [46] Anna Himmelhuber, Mitchell Joblin, Martin Ringsquandl, and Thomas A. Runkler. Demystifying graph neural network explanations. In *PKDD/ECML Workshops (1)*, volume 1524, pages 67–75, 2021.
- [47] Benjamín Sánchez-Lengeling, Jennifer N. Wei, Brian K. Lee, Emily Reif, Peter Wang, Wesley Wei Qian, Kevin McCloskey, Lucy J. Colwell, and Alexander B. Wiltchko. Evaluating attribution for graph neural networks. In *NeurIPS*, 2020.
- [48] Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating explainability for graph neural networks. *CoRR*, abs/2208.09339, 2022.
- [49] Chirag Agarwal, Marinka Zitnik, and Himabindu Lakkaraju. Probing GNN explainers: A rigorous theoretical and empirical analysis of GNN explanation methods. In *AISTATS*, volume 151, pages 8969–8996, 2022.
- [50] Lukas Faber, Amin K. Moghaddam, and Roger Wattenhofer. When comparing to ground truth is wrong: On evaluating GNN explanation methods. In *KDD*, pages 332–341, 2021.
- [51] He Zhang, Bang Wu, Xingliang Yuan, Shirui Pan, Hanghang Tong, and Jian Pei. Trustworthy graph neural networks: Aspects, methods and trends. *CoRR*, abs/2205.07424, 2022.
- [52] Yongduo Sui, Xiang Wang, Tianlong Chen, Meng Wang, Xiangnan He, and Tat-Seng Chua. Inductive lottery ticket learning for graph neural networks. *Journal of Computer Science and Technology*, 2023.
- [53] Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph rationalization with environment-based augmentations. In *KDD*, pages 1069–1078, 2022.
- [54] Yongduo Sui, Tianlong Chen, Pengfei Xia, Shuyao Wang, and Bin Li. Towards robust detection and segmentation using vertical and horizontal adversarial training. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.
- [55] Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. Causal attention for interpretable and generalizable graph classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1696–1705, 2022.

- [56] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.
- [57] Junfeng Fang, Xiang Wang, An Zhang, Zemin Liu, Xiangnan He, and Tat-Seng Chua. Cooperative explanations of graph neural networks. In *WSDM*, pages 616–624. ACM, 2023.
- [58] Ari S. Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. In *NeurIPS*, pages 4933–4943, 2019.
- [59] Kun Wang, Yuxuan Liang, Pengkun Wang, Xu Wang, Pengfei Gu, Junfeng Fang, and Yang Wang. Searching lottery tickets in graph neural networks: A dual perspective. In *The Eleventh International Conference on Learning Representations*, 2022.
- [60] Junfeng Fang, Wei Liu, An Zhang, Xiang Wang, Xiangnan He, Kun Wang, and Tat-Seng Chua. On regularization for explaining graph neural networks: An information theory perspective. 2022.
- [61] Jiaxing Zhang, Dongsheng Luo, and Hua Wei. Mixupexplainer: Generalizing explanations for graph neural networks with data augmentation. In *KDD*, pages 3286–3296. ACM, 2023.
- [62] Lukas Faber, Amin K. Moghaddam, and Roger Wattenhofer. Contrastive graph neural network explanation. *CoRR*, abs/2010.13663, 2020.
- [63] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [64] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *CoRR*, abs/1710.10903, 2017.
- [65] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*. OpenReview.net, 2019.

A Algorithms

Algorithm 1 presents the pseudocode of the evaluation process of our proposed method OAR. The pseudocode of SimOAR can be obtained by removing the line 1, 6, and 8 and simply modifying line 9 into “ $s \leftarrow \frac{1}{N_{adv}} \sum_i y^{(i)}$ ”. For clarity, we put the step of feeding adversarial graph $\mathcal{G}'^{(i)}$ into the target GNN and the VGAE under the for-loop. However, when implementing in real code, we can batch all those N_{adv} adversarial graphs and feed them at one time, after the sampling process is finished, to expedite computation. Meanwhile, Algorithm 2 presents the sampling process of fake explanatory subgraphs for general evaluation.

Algorithm 1 Evaluation Process of OAR

Input: Trained GNN f ; To-be-evaluated subgraph \mathcal{G}_s and its corresponding original graph \mathcal{G} and dataset \mathcal{D} ; Perturbation ratio R ; Number of adversarial graphs N_{adv}

Output: Evaluation score s

- 1: Train a standard VGAE on \mathcal{D} according to [19].
 - 2: $c \leftarrow \arg \max_i f(\mathcal{G})_i$.
 - 3: **for** $i = 1, 2, \dots, N_{adv}$ **do**
 - 4: $\mathcal{G}'^{(i)} \leftarrow$ randomly deleting $\lfloor R \cdot |\mathcal{E}_{\mathcal{G}}| \rfloor$ edges from \mathcal{G} while fixing \mathcal{G}_s .
 - 5: $y^{(i)} \leftarrow f(\mathcal{G}'^{(i)})_c$.
 - 6: $\mathcal{L}_{recon}^{(i)} \leftarrow$ calculated according to Equation 5 using the trained VGAE.
 - 7: **end for**
 - 8: $w_{OOD}^{(i)} \leftarrow \frac{1/\mathcal{L}_{recon}^{(i)}}{\sum_j 1/\mathcal{L}_{recon}^{(j)}}, i = 1, 2, \dots, N_{adv}$.
 - 9: $s \leftarrow \sum_i w_{OOD}^{(i)} \cdot y^{(i)}$.
-

Algorithm 2 Sampling Fake Explanatory Subgraphs for General Evaluation

Input: Dataset $\mathcal{D} = \{\mathcal{G}^1, \mathcal{G}^2, \dots, \mathcal{G}^N\}$; Number of Recall levels N_L ; Number of sampled subgraphs per graph N_{sub} ; Size of sampled subgraph K_{sub}

Output: Pairs of Recall level and corresponding subgraphs $(L_k, \{\mathcal{G}_{s,k}^{i,j} \mid i = 1, 2, \dots, N; j = 1, 2, \dots, N_{sub}\})$, $k = 1, 2, \dots, N_L$

- 1: **for** $k = 1, 2, \dots, N_L$ **do**
 - 2: $L_k \leftarrow \frac{k-1}{N_L-1}$
 - 3: **for** $i = 1, 2, \dots, N$ **do**
 - 4: $K^{GT} \leftarrow$ the number of edges in the ground-truth explanation of \mathcal{G}^i
 - 5: $K_{pos} \leftarrow \lfloor L_k \times K^{GT} \rfloor$
 - 6: $K_{neg} \leftarrow K_{sub} - K_{pos}$
 - 7: **for** $j = 1, 2, \dots, N_{sub}$ **do**
 - 8: $\mathcal{G}_{s,k}^{i,j} \leftarrow$ a connected subgraph, randomly sampled from \mathcal{G} , with K_{pos} edges in the ground truth explanation and K_{neg} edges not in it.
 - 9: **end for**
 - 10: **end for**
 - 11: **end for**
-

B Experimentals

B.1 Experimental Details

All experiments are conducted on a Linux machine with 8 NVIDIA GeForce RTX 3090 (24 GB) GPUs. CUDA version is 11.6 and Driver Version is 510.39.01. All codes are written under Python 3.9.13 with PyTorch 1.13.0 and PyTorch Geometric (PyG)[63] 2.2.0. We adopt the Adam optimizer throughout all experiments.

Overall, for each dataset, a target GNN classification model is well-trained first. Then the explainers are built on the GNN and generate explanations for its prediction on the dataset. After that, the explanation evaluation methods evaluate how well the explanations are. Our work stands at the last level.

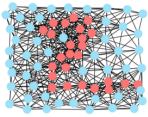

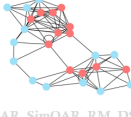
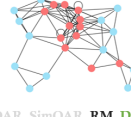
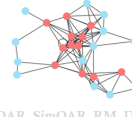
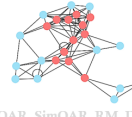
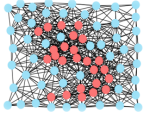

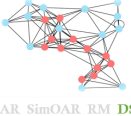
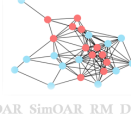
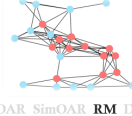
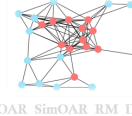
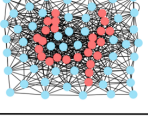
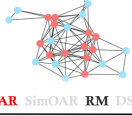




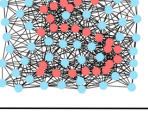
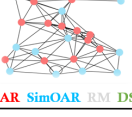

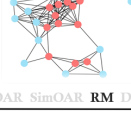


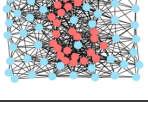

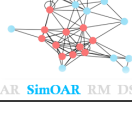



Study Cases from MNIST-sp						
class	Original Graph	Human Intuition				
		High Quality				Low Quality
		Explanation 1	Explanation 2	Explanation 3	Explanation 4	Explanation 5
2		 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE
3		 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE
4		 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE
5		 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE
6		 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE	 OAR SimOAR RM DSE

Figure 5: Study cases. For each row, the explanations are ranked based on average rankings given by volunteers. Highlighted evaluation method below each explanation means that the method has given the explanation the highest score compared to other explanations in that row. Best viewed in color.

Target GNNs. GNNs have garnered significant recognition for their prowess in encoding graph data [36, 64, 65, 33, 59]. Amidst the vast landscape of GNNs, GIN [65] distinguishes itself with its superior encoding aptitude. In sight of this, the target GNNs for BA3, TR3, and Mutagenicity have the same structure, which is a two-layered GIN followed by a two-layered MLP with 32 hidden channels. They are trained with max epochs equal to 20, 200, and 200 respectively, batch size equal to 128, and learning rate equal to 0.001. The target GNN for MNIST-sp is adapted from an example code¹ provided by PyG, trained with the number of epochs equal to 20, batch size equal to 64, and initial learning rate equal to 0.01. Before training, we randomly split BA3, TR3, and Mutagenicity into train and test sets with ratios of 90% and 10%, respectively, while adopting the split provided by PyG for MNIST-sp. During training, we reserve data of the same size as the test set from the train set as the validation set and save the model which reaches the highest classification accuracy on the validation set for later use.

Explainers. We have implemented six state-of-the-art post-hoc explainers, namely, SA, GradCAM, GNNExplainer, PGExplainer, CXPlain, and ReFine, as claimed in Section 3.1, to generate explanatory subgraphs. Here we give a brief introduction to them:

- **SA** [10] captures the gradients *w.r.t.* adjacency matrix of the input features in the process of backpropagation and directly treats them as the importance scores.
- **GradCAM** [41] takes one step further over SA via improving the gradients *w.r.t.* the input features like edges by using context within the graph convolutional layers.
- **GNNExplainer** [11] directly learns an adjacency matrix mask by maximizing the mutual information between a GNN’s prediction and distribution of possible subgraph structures.
- **PGExplainer** [42] adopts a deep neural network to parameterize the generation process of explanations, which makes it a natural approach to explaining multiple instances collectively.

¹Reference code for the target GNN of MNIST-sp: https://github.com/pyg-team/pytorch_geometric/blob/89a54d9454d3832f814f9a574ed421c58f1fce10/examples/mnist_voxel_grid.py

Table 3: Average consistency scores of different evaluation metrics (corresponding to Table 1 in the main paper).

	SHAPEGGEN	TR3	MUTAG	MNIST	BA3
GEF	0.800	0.734	0.800	0.934	0.734
SimOAR	0.800	0.867	0.800	0.934	0.934
OAR	0.867	0.934	1.000	0.934	1.000

Table 4: Correlation between metrics and Recall across explanatory subgraphs in four node classification datasets in [11] (corresponding to Figure 3 (a) in the main paper).

	BA-Shapes	BA-Community	Tree-Cycles	Tree-Grid
RM	0.312	0.321	0.295	0.411
DSE	0.406	0.377	0.343	0.384
OAR	0.542	0.560	0.489	0.440
SimOAR	0.527	0.535	0.471	0.431

- **CXPlain** [43] treats the explanations as a causal learning task and trains causal explanation models that learn to estimate to what degree certain inputs cause outputs in the to-be-explained model.
- **ReFine** [38] leverages the pre-training explanations to exhibit global explanations and the fine-tuning explanations to adapt the global explanations in the local context.

The default hyper-parameters suggested by those papers are adopted, as whether the explainers are at their optimal state is secondary to our work. Among these explainers, SA, GradCAM, and GNNExplainer directly take the to-be-explained graph as input, while the rest three need to be trained on a set of graphs, *i.e.*, the train set in our case, in advance. We only use the explanations extracted from graphs in the test set for evaluation.

Evaluation Methods. Finally, we arrive at the explanation evaluation level. There are four evaluation methods, *i.e.*, removal-based evaluation, DSE, OAR, and SimOAR, to be considered. Removal-based evaluation directly feeds the explanatory subgraph into the target GNN and gets its prediction, on the predicted class of the original graph, as the evaluation score, which does not involve any details. For DSE, we make use of its public source code² and follow its paper to set hyper-parameters. As for our method OAR/SimOAR, we have summarized the entire process of OAR in Algorithm 1. Here we present more details on how the VGAE is trained. The encoder involves two two-layered GCNs for obtaining μ and σ , each of which is realized with hidden channels equal to 256 and output channels equal to 128, while the two GCNs share the first layer. The dataset split process is the same as the training of the target GNNs. We train the VGAE model on the train set with the number of epochs equal to 100, batch size equal to 256, and learning rate equal to 0.001. The model that reaches the lowest loss on the validation set is saved for later OOD reweighting.

B.2 More Quantitative Results

Here, we sequentially present the experimental results of 1) comparison between GEF [48] and our metrics *w.r.t* the latest dataset SHAPEGGEN and the other datasets (Table 3), 2) correlation between metrics and Recall across explanatory subgraphs in four node classification dataset (Table 4), and 3) correlation between metrics and Recall while evaluating the explanations generated by the inherent explainable GNN, GSAT [15] via introducing an additionally well-trained GIN [65] as the model f in Algorithm 1 (Table 5).

Note that while evaluating explanations in node classification tasks, for each node in the input graph, we construct an ego graph for it based on the number of layers in the baseline GNN. Then, the explanation task for node classification can be transferred to the explanation task for graph classification. Furthermore, the remaining hyperparameters and methods in our OAR/SimOAR remain unchanged.

²Source code of DSE: <https://anonymous.4open.science/r/DSE-24BC/>

Table 5: Correlation between metrics and Recall while evaluating the explanations generated by the inherent explainable GNN, GSAT (corresponding to Figure 3 (a) in the main paper).

	BA3	MUTAG	TR3	MINST
RM	0.376	0.381	0.402	0.337
DSE	0.411	0.425	0.417	0.319
OAR	0.613	0.590	0.585	0.606
SimOAR	0.598	0.572	0.552	0.581

C User Study

In order to measure the consistency between evaluation results and human intuition, we organized a large-scale user study engaging 100 volunteers. Each volunteer was asked to check 5 groups of graphs, which contain an instance from MNIST-sp, its predicted class, and 5 randomly sampled subgraphs from this instance, and try to rank these 5 subgraphs according to how well they serve as explanations of the prediction based on intuition. We exhibit partial results in Figure 5. According to these results we can find that our evaluation methods *i.e.* OAR and SimOAR show the highest consistency with human intuition.