A True-to-the-model Benchmark for Edge-level Attributions of GNN Explainers

Francesco Paolo Nerini[1,2][0009−0000−2936−1297] (✉), Francesco Bonchi[2,3][0000−0001−9464−8315], and André Panisson[2][0000−0002−3336−0374]

[1] Sapienza University, Rome, Italy
[2] CENTAI Institute, Turin, Italy {fpn, bonchi, panisson}@centai.eu
[3] Eurecat, Barcelona, Spain

Abstract. Edge-level explainers for Graph Neural Networks (GNNs) aim to identify the most crucial edges that influence the model's predictions in a node classification task. Benchmarking these explainers is particularly challenging due to the extensive search space of potential explanations and the absence of reliable ground truths for edge importance. Moreover, the evaluation methods which are prominent in the literature rely on assumptions about which subgraphs in the input data influence the classification of a node, yet they provide no guarantee that the model has effectively learned the intended behavior.

In this paper, we address these limitations by introducing a white-box GNN model together with a theoretical analysis to identify which edges are truly important, i.e., when removed, they can alter the classification. We demonstrate the effectiveness of this framework on both synthetic and real-world node classification tasks, using metrics that account for the inherent imbalance between the few relevant edges and the many irrelevant ones. Our evaluation reveals two recurring issues in current explainability methods: the frequent misidentification of unimportant edges as important ones, and numerical instability in some attribution techniques. To address these issues, we propose two corrective strategies that significantly enhance the reliability of edge-level attributions: a post-processing method to refine edge rankings and a rescaling of model weights to stabilize numerical outputs.

Our work provides valuable insights into the strengths and weaknesses of existing GNN explainers and presents practical solutions to advance the fine-grained explainability of graph-based models.

Keywords: Explainable AI · Graph Machine Learning · XAI Benchmarking.

1 Introduction

Graph structures are ubiquitous in data science. From chemical bonds to financial transactions, graphs encompass many situations where relations between elements add additional information to the task at hand. In many applications, Graph Neural Networks (GNNs) have been introduced as an effective tool for learning from these relations and performing predictions. However, these methods are inevitably undermined by their black-box nature. The low human understandability of these techniques raises concerns from regulators and practitioners alike about their algorithmic decisions. Due to this reason, a large corpus of works on explainable artificial intelligence has been developed specifically to deal with graphs' peculiarities. Nonetheless, a general and common understanding of what makes a good explanation is still lacking, slowing the adoption of these techniques.

One of the main obstacles to adopting explainability techniques in graph machine learning is the inherent difficulty in validating these methods. Proper validation of a local explanation approach requires: (i) a precise definition of what constitutes a good explanation, i.e., identifying the elements that the explainer should consider as important for the model's decision, and (ii) a robust evaluation framework comprising models, ground truth explanations, and objective evaluation metrics. Prior research has largely focused on validating explanations from a true-to-the-data perspective [1, 14], where ground truths are defined by inherent data structures or by artificially implanting target structures in the data. While this strategy may seem promising initially, it assumes that the model has actually learned the logic behind those inherent or implanted structures. This is a strong assumption, as models often capture spurious correlations with little relation to the underlying data-generation process. In contrast, significantly less attention has been devoted to establishing a *true-to-the-model* evaluation of explanations (e.g. [15]), which assesses an explainer's ability to faithfully capture the intrinsic reasoning underlying a GNN's classification decision.

In this work, we adopt a true-to-the-model perspective by employing a simple yet meaningful toy model of a GNN that is fully interpretable: we call it a white-box model—in contrast with the typical black-box nature of GNNs—to represent the fact that its inner logic in clearly known and thus defines a ground truth for the explanation task. We focus on edge-level explanations, which are the most fine-grained type of explanation available for graph structures. The approach eliminates the uncertainties about the extent to which the GNN has captured the true data-generating process, thereby avoiding many of the pitfalls inherent in a *true-to-the-data* evaluation framework [4] while enabling a rigorous analysis of explanation quality.

First, we define an axiom of importance for the edges. Then, leveraging our knowledge of the model's inner functioning, we can prove that the important edges correspond to a human intuition of importance. We apply the model to synthetic and real datasets in a node classification task. Using metrics that account for the unbalanced proportion between the minority of relevant edges and the majority of irrelevant ones, we observe that the explainers cannot always identify the most relevant edges. In particular, we observe two common problems. The first is due to certain edge patterns, where the explainer alternates unimportant edges with important ones. The second problem is the numerical instability of some of the methods. The first problem can be solved by post-processing the explanations, while the second can be mitigated by rescaling the models' weights. We show how these patches can improve the quality of the produced explanations, enhancing the explainability of GNN models.

The main contributions of this paper are summarized as follows:

1. We introduce a white-box GNN along with a theoretical framework for a true-to-the-model benchmark of edge-level attributions;
2. We examine different metrics for the evaluation of the explanations, and observe that no explainer is capable of always identifying which of the edges are the most important;
3. By analysing the cases where the explainers fail the most, we identify two independent and systematic mistakes in the explanations;
4. Finally, we propose two solutions for overcoming these problems, showing that simple and fast improvements can lead to increased performance of the explainers.

We release all the code of the white-box model and the experimental setup for reproducibility[4].

2 Background and related work

Graph neural networks (GNNs) have become essential in both research and practical applications, with widely used architectures such as Graph Convolutional Networks (GCNs) [11], GraphSAGE [8], and Graph Attention Networks (GATs) [20] relying on message-passing mechanisms. Although all of these message-passing schemes are inherently opaque, our focus is on models like GCNs and GraphSAGE where the message weights are fixed, as opposed to the dynamically learned weights in attention-based approaches like GATs.

Understanding and interpreting GNN decisions is critical for establishing trustworthy models [27]. In this work, we address post-hoc explainability methods—generally referred to as "explainers"—that produce edge-level explanations. Explainers can be divided into three classes, depending on the scope of their explanations: instance-level, class-level, or model-level explainers [12]. We concentrate on instance-level explainers (which consider a single model decision at a time) that provide fine-grained, edge-specific attributions. Edge-level explainers can be further divided into different categories, as discussed in [24, 12]:

- **Mask-based explainers:** These methods, that include GNNExplainer [23] and SubgraphX [25], generate hard or soft masks for the graph's adjacency matrix to highlight important edges.
- **Causal-based explainers:** This category includes explainers that use causal inference techniques OrphicX [13] and policy-based explainers such as ZORRO [5].

[4] https://github.com/FrappaN/EdgeWhiteBoxBench

- **Perturbation-based approaches:** According to [24], both Causal-based explainers and Mask-Based explainers fall into this broader category, where edge importance is inferred by analyzing the effect of perturbations.
- **Gradient-based methods:** Techniques such as Integrated Gradients [19] and Grad-CAM [16] estimate edge relevance using output gradients.
- **Decomposition-based explainers:** These methods decompose the model's output into contributions from individual edges or features. Examples include LRP [3], GStarX [28], and FlowX [7]. However, some are limited to specific tasks (e.g., GStarX for graph classification) or require model-aware implementations (e.g., GNN-LRP [18]).
- **Surrogate-based explainers:** Approaches like GraphLIME [10] and PGM-Explainers [21] build local interpretable models to approximate the decision-making process, though they generally do not provide detailed edge-level explanations.

Early studies such as [4] compared different explainability techniques using synthetic graphs with data-dependent ground truth labels, exposing significant limitations of this evaluation strategy. Subsequent work [17, 1, 14] has assessed explainers via multiple model calls and filtered edge masks. In contrast, the white-box approach proposed in [15] establishes ground truth by leveraging interpretable models but focuses on feature-level explanations. This strategy is closely aligned with our methodology for benchmarking edge-level explanations.

3 Methods

In this section, we present our methodology for evaluating (and enhancing) edge-level explanations in GNNs. First, we introduce a white-box GNN model inspired by the label propagation algorithm, which provides an interpretable and controlled framework for message-passing in graphs (§3.2). Second, we formalize the concept of edge importance through an axiomatic framework that identifies the minimal subgraphs critical for a model's prediction (§3.3). Finally, we propose a post-processing strategy to refine explainer outputs by eliminating spurious attributions and enhancing explanation fidelity (§3.4).

3.1 Notation

Consider a binary node classification task on a directed graph $G = (V, E)$, where V is the set of nodes, and each node is associated with features from the set F. The graph structure is represented by the edges $E \subseteq 2^{V \times V}$ and the node-feature matrix $X \in \mathbb{R}^{|V| \times |F|}$. For a given node $v \in V$, the model is defined as a function:

$$\mathcal{M} : \mathbb{R}^{|V| \times |F|} \times 2^{V \times V} \times V \to [0, 1]\,,$$

which outputs a prediction score in the interval $[0, 1]$.

To facilitate explanation, we discretize the model's output into classes:

$$\widehat{\mathcal{M}}(X,E,v) = \begin{cases} 0, & \text{if } \mathcal{M}(X,E,v) < 0.5 \\ 1, & \text{if } \mathcal{M}(X,E,v) \geq 0.5 \end{cases}$$

We also denote $p_0(X,E,v) = 1 - \mathcal{M}(X,E,v)$ the probability of v being in class 0, and similarly $p_1(X,E,v) = \mathcal{M}(X,E,v)$ the probability of v being in class 1.

Since our focus is on the influence of edges on the model's output, we sometimes simplify the notation by omitting X and writing $\mathcal{M}(E,v)$ (or $p_c(E,v)$). An explainer for the model $\mathcal{M}$ assigns an importance score to each edge in E for the classification of node v. We denote this as $\mathcal{E}_{\mathcal{M}}(E,v) = \beta \in \mathbb{R}^{|E|}$, where β_e is the importance assigned to edge $e \in E$.

3.2 A white-box model inspired by label propagation

To evaluate explanation quality against a well-defined ground truth, we require a realistic white-box model whose edge importance is both interpretable and controllable, with mechanisms that are learnable by a GNN in a real scenario. For this purpose, we propose a white-box GNN that approximates the *label propagation*, a heuristic widely used for community detection and node classification [29, 6], in which, starting from a subset of labeled nodes, at each iteration each node adopts the majority label of its neighbours, until all nodes are labelled.

A single-layer GNN can emulate one iteration of label propagation. For simplicity, consider a binary classification task. The model assigns to each node v an embedding $X_v \in \{(1,0),(0,1),(0,0)\}$, where $(1,0)$ corresponds to a node known to belong to the first class, $(0,1)$ to one belonging to the second class, and $(0,0)$ to any unlabelled node. The model then performs one round of message passing:

$$H_v^{(1)} = \text{ReLU}(\sum_{u \in \mathcal{N}(v)} X_u W), \text{ with } W = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \tag{1}$$

followed by a final transformation:

$$\mathcal{M}(X,E,v) = \sigma\Big(H_v^{(1)}\, w^\top\Big), \tag{2}$$

where σ is the sigmoid function, $w = (1,-1)$, and $\mathcal{N}(v)$ are the nodes neighbours of v. To mimic multiple iterations of label propagation, we generalize the model to L rounds by repeating the propagation step with normalization:

$$H_v^{(l)} = \text{ReLU}(\sum_{u \in \mathcal{N}(v)} \frac{H_u^{(l-1)}}{\|H_u^{(l-1)}\|} W) \text{ for } l = 2, \ldots L; \tag{3}$$

After which, the same final transformation with a sigmoid activation is applied.

Note that scaling the weight matrix W by any scalar factor f does not alter the model's classification. We will later use this property to improve the numerical stability of the explanations, as detailed in Section 4.2.

For multi-class classification, the model can be generalized by encoding node labels as one-hot vectors. In this case, the weight matrix W is a matrix with 1 on the diagonal and -1 everywhere else. The propagation rule remains the same as in Eq. 3, and a softmax activation is applied at the output layer. Under this assumption, a node is assigned to a particular class only if a strict majority of its labelled neighbours belong to that class.

3.3 Defining edge importance

We define the importance of an edge by its contribution to the model's prediction. To formalize this notion, we introduce the concept of *important subgraph*.

Definition 1. *Given a graph $\mathcal{G} = (V, E)$ and a binary node classification model $\mathcal{M}$ such that $\widehat{\mathcal{M}}(E, v) = c$ and the predicted probability $p_c(E, v) > 0.5$ for some node v, a set of edges $A \subseteq E$ is an* ***important subgraph*** *for $\mathcal{M}$ and v iff:*

1. *Removing A from the graph changes the prediction, i.e., $p_c(E \setminus A, v) \leq 0.5$;*
2. *No proper subset $B \subset A$ has this effect, meaning that for all $B \subset A$, the model still predicts class c with $p_c(E \setminus B, v) > 0.5$.*

In other words, an important subgraph is a minimal set of edges whose removal causes the model's confidence to drop below the decision threshold. Different important subgraphs can share multiple edges.

Let $\mathcal{I}$ denote the set of all important subgraphs:

$$\mathcal{I} = \{A \subseteq E | A \text{ is an important subraph for } \mathcal{M}\},$$

and define the union of all such subgraphs as:

$$S = \bigcup_{A \in \mathcal{I}} A.$$

For the evaluation of explainers, we focus on checking whether an edge belongs to S. Accordingly, we propose the following axiom:

Axiom 1 *Let $\mathcal{G} = (V, E)$ be a graph, $\mathcal{M}$ a binary node classification model, and $\mathcal{E}_{\mathcal{M}}$ an explainer that produces an attribution vector $\beta \in \mathbb{R}^{|E|}$. Denote by S the union of all important subgraphs for $\mathcal{M}$. Then, for every edge $e \in S$ and every edge $e' \notin S$,*

$$|\beta_e| > |\beta_{e'}|.$$

In other words, we expect that a faithful explainer assigns higher importance to edges that are critical to the model's prediction.

Defining important subgraphs in this way also allows us to assess whether an explainer can capture redundant structures in the graph. While such redundancy can complicate evaluations based solely on ground-truth data [4], our white-box model ensures that redundant structures are indeed used by the model, thereby enabling a more reliable evaluation.

In our framework, the importance of specific edges is controlled by the initial node labelling. For a binary classification task, the prediction for a node v depends on the difference $n_{c_1,v} - n_{c_2,v}$, where $n_{c,v}$ denotes the number of neighbours of node v with initial label c.

This intuition is formalized in the following proposition for a 2-layer binary label propagation model:

Proposition 1. *Consider a graph $\mathcal{G} = (V, E)$, a node $r \in V$, and a 2-layer (binary) label propagation model $\mathcal{M}$ with initial labels X, such that:*

$$\widehat{\mathcal{M}}(X, E, r) = c \in \{0, 1\}.$$

We define the intermediate (after one layer) and initial labels as:

$$c_u^{(1)} = \arg\max_i H_{u,i}^{(1)}; \qquad c_u^{(0)} = \arg\max_i X_{u,i};$$

with $c_u^{(l)} = 0.5$ if the maximum is not unique. Then, if $\mathcal{M}(X, E, r) \neq 0.5$, the following edges are important according to our definition:

1. *All edges $e = (s, r) \in E$ such that $c_s^{(1)} = c$;*
2. *All edges $e = (t, s) \in E$ such that $s \in \mathcal{N}(r)$ and $c_t^{(0)} = c_s^{(1)} = c$.*

In essence, this proposition shows that all paths from nodes initially labelled as the predicted class of the target node r are important. This result can be easily extended to the multiclass case.

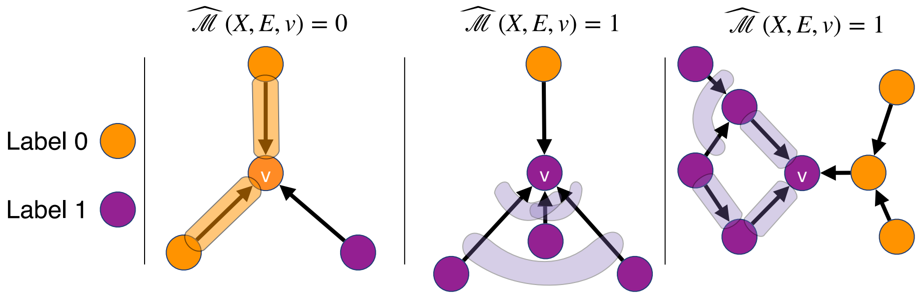

$\widehat{\mathscr{M}}(X, E, v) = 0$
$\widehat{\mathscr{M}}(X, E, v) = 1$
$\widehat{\mathscr{M}}(X, E, v) = 1$
Label 0
Label 1
v
v
v

Fig. 1: Examples of the output of the white-box label-propagation-inspired model: important subgraphs are highlighted with shaded areas.

3.4 Post-processing Edge Importance to Improve Explanations

Our proposed definition of edge importance, which directly ties the relevance of an edge to its role in the message-passing mechanism, allows us to naturally introduce a post-processing step aimed at refining explanation quality. Specifically,

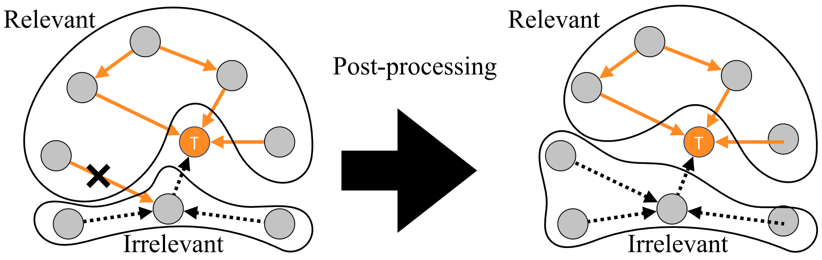


Fig. 2: (left) Example of a misleading attribution pattern for the classification of node "T". In the initial explanation, relevant edges are depicted as continuous orange lines and irrelevant ones as black dotted lines; the false positive edge is marked with a cross. (right) After applying our post-processing procedure, the false positive edge is corrected, resulting in an explanation that accurately reflects the continuous path of relevant edges.

we observed that explainers often assign relevance scores to edges that, by construction, should be considered unimportant for the model's prediction, forming recurring misleading attribution patterns (e.g., Figure 2). These patterns typically involve assigning high importance to edges connecting second-order neighbours to immediate neighbours, even when those edges don't influence the target node's final prediction according to our definitions.

Leveraging our formalization, we identify an intuitive correction rule: if an edge is deemed unimportant, any edges preceding it in the message-passing path should also be considered unimportant. More precisely, in a directed graph, if a node A receives a message from node B along an edge that is considered unimportant by the explainer, then all edges incoming to node B should also be considered unimportant with respect to node A's classification.

To enforce this rule, we define a threshold on edge importance scores to distinguish between important and unimportant edges clearly. For attribution methods producing both positive and negative importance scores (e.g., Integrated Gradients, LRP, and Deconvolution), we set this threshold at zero. For methods generating positive scores in the range $[0, 1]$ (e.g., GNNExplainer), we use a threshold of 0.5. When an edge that is followed by an unimportant edge exceeds this threshold, we reduce its attribution score accordingly. In both cases, applying this threshold results in a noticeable improvement in the quality of the explanations.

4 Experiments

In this section, we present our experimental evaluation of edge-level explainers. We evaluate the performance of these methods using both synthetic and real-world graphs. The goal of our experiments is to quantify how well the explainers capture the edge importance as defined by our white-box label propagation model, and to evaluate the effectiveness of our post-processing strategy in refining the explanations.

4.1 Experimental Setup

Synthetic Graphs: We generate synthetic graphs using the Erdős-Rényi model with a fixed number of 1000 nodes and explore three different edge connection probabilities: $p = 0.005$, $p = 0.01$, and $p = 0.05$. In order to simulate a supervised node classification scenario, we initialize 80% of the nodes with randomly assigned labels. To introduce a higher degree of homophily, which is a common assumption in many GNN models, we rewire 1/3 of the edges originating from labelled nodes to connect them with other nodes sharing the same label. To evaluate the explanations, we select 100 nodes that were not initially labelled and for which the model predicts a class with a probability greater than 0.5.

Real Graphs: Our experiments on real data involve three widely-used citation network datasets: Cora [22], Pubmed [22], and OGBN-ArXiv [9]. In these experiments, we use the multiclass version of the Label Propagation Model. Similarly to the synthetic setup, 80% are initialized with their true labels. To evaluate the explanations, we focus on nodes for which the predicted class probability is strictly higher than that of any other class.

Explainers: Many of the explainers in the literature do not provide edge-level explanations or are tailored for node classification tasks. For example, GraphLIME does not provide edge-level attributions, and GStarX is designed for graph-level tasks rather than node-level ones. Therefore, we consider five explainers that are both relevant and directly applicable to our setting. These methods span three categories:

- Gradient-based explainers: Integrated Gradients (IG) [19].
- Mask-based explainers: GNNExplainer [23] and SubgraphX [25].
- Decomposition-based explainers: LRP [2] and Deconvolution [26] (a hybrid decomposition/gradient method).

For GNNExplainer, we set the number of training epochs to $10,000$, since we observed the performance increases with the number of epochs. We left the other parameters to their default values. In contrast, SubgraphX not only required significantly longer runtimes to generate explanations but also produced results of lower quality. This discrepancy likely stems from how SubgraphX operates: instead of attributing importance directly to edges, it first selects the most relevant nodes and then returns the induced subgraph. As a consequence, its results do not align well with methods that inherently produce edge-level explanations.

Evaluation Metrics: To quantitatively assess the quality of edge-level explanations, we use two evaluation metrics: the Receiver Operating Characteristic Area Under the Curve (ROC-AUC) and the Precision-Recall Area Under the Curve (PR-AUC). These metrics are computed over the set of edges in the 2-hop neighbourhood of each target node, where each edge is labelled as either relevant or irrelevant based on our ground truth.

Due to the inherent class imbalance—where irrelevant edges vastly outnumber relevant ones—the Precision-Recall metric is particularly effective in capturing the performance of edge-level explainers. Our evaluation further considers variations in graph properties by testing on random graphs generated with different edge connection probabilities ($p = 0.005$, 0.01, and 0.05), which influence the overall density and the ratio of relevant to irrelevant edges.

4.2 Results

Impact of Graph Density on Edge-Level Explanation Performance

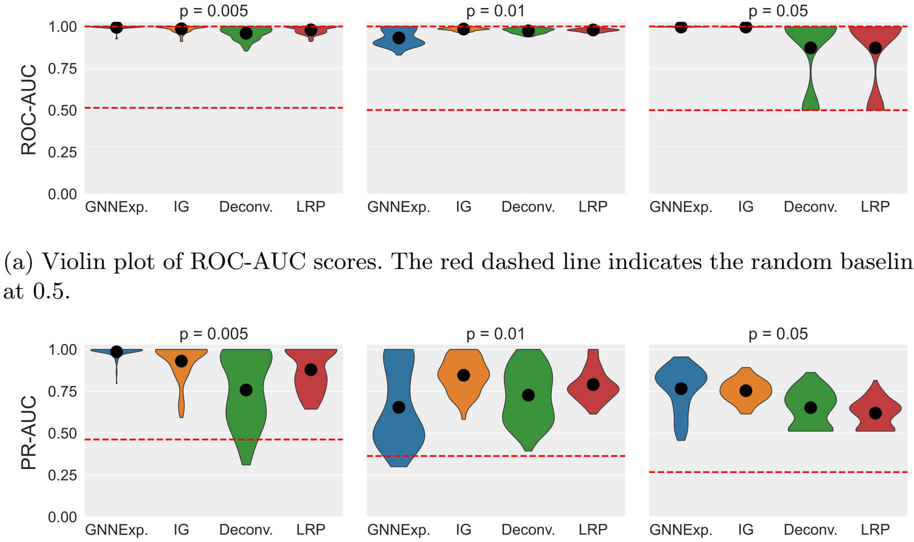
In Figures 3a and 3b we present violin plots of the ROC-AUCs and PR-AUCs, respectively, across random graphs generated with varying edge probabilities (p). For each graph, we evaluate the explainers on a sample of 100 nodes. These metrics are computed based on the ground truth labelling of edges—relevant versus irrelevant—in the 2-hop neighbourhood of each target node.

The main difference between these metrics is in their treatment of false positives. While ROC-AUC uses the false positive rate (FPR), defined as $\text{FPR} = \frac{FP}{FP+TN}$, PR-AUC relies on precision, defined as $\text{Precision} = \frac{TP}{TP+FP}$). As shown in Figure 4, although the FPR remains close to zero beyond a threshold $t = 0$ for all explainers, precision is more sensitive to threshold variations. This observation highlights that even a small number of false positives can significantly affect the ratio of true positives. This sensitivity makes PR-AUC particularly effective for evaluating edge explanations in settings with a high imbalance between relevant and irrelevant edges.

Recurring Misleading Attribution Patterns

In experiments on sparse, small graphs, we observed that explainers consistently assign importance to edges in recurring, misleading patterns, as exemplified by Figure 2.

To address this issue, we implemented the post-processing procedure detailed in Section 3.4 that adjusts the importance scores based on the message-passing paths. This post-processing step significantly improves performance, as shown in Figure 5 and Table 1. In particular, all explainers benefit from this correction except for GNNExplainer, which consistently assigns importance to all edges in the entire misleading pattern rather than just a subset of its edges.



(a) Violin plot of ROC-AUC scores. The red dashed line indicates the random baseline at 0.5.

(b) Violin plot of PR-AUC scores. The red dashed line indicates the random baseline.

Fig. 3: Violin plots displaying the distribution of ROC-AUC (a) and PR-AUC (b) scores for edge-level explanations on a 2-layer label propagation model. Evaluations were performed on 100 nodes sampled from Erdős-Rényi graphs with varying edge connection probabilities. Black points correspond to the mean scores.

Results on empirical graphs

The same problems and behaviours observed on synthetic graphs have also been observed in our experiments on real-world datasets. We evaluate the explainers on three empirical graphs: Cora, PubMed, and OGBN-ArXiv, where 80% of the nodes are initialized with their true labels. For sparser graphs, Cora and PubMed, the explainers achieve near-perfect performance, indicating that the underlying sparsity and homophily help to generate accurate edge-level attributions. In contrast, performance on OGBN-Arxiv, which is both a denser and larger graph, is comparatively lower.

Detailed results are presented in Table 1, Figure 6, and Figure 7. In particular, the post-processing procedure consistently increases the performance of all explainers across these datasets. These results reinforce our observations from synthetic graphs and emphasize the importance of addressing false positives. Moreover, in the next section, we show how weight scaling further benefits the explanations, particularly in denser graph settings.

Numerical Stability of Explanations

Our experimental results from Figure 5 show that as graph density increases, the performance of several explainers degrades, even after applying our post-

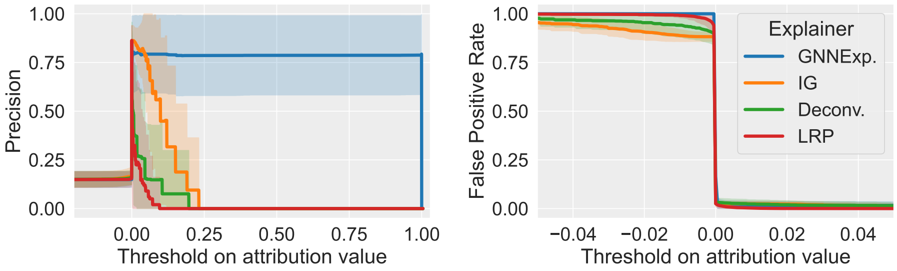


Fig. 4: On the left, the mean precision of the explainer at different attribution value thresholds on a sparse graph with $p = 0.01$; on the right, the mean false positive rate on the same graph; in both plots, the shaded areas correspond to the standard deviation across the explained node predictions.

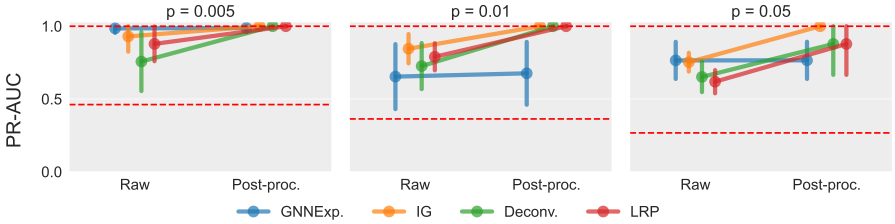


Fig. 5: Mean PR-AUC across nodes before and after post-processing of explanations on the same Erdős-Rényi random graphs of Figure 3; there's an improvement for most explainers, apart from GNNExplainer.

processing procedure. In particular, while GNNExplainer remains relatively stable, both LRP and Deconvolution exhibit improvements in PR-AUC with post-processing but still fall short of the performance achieved by Integrated Gradients on dense graphs.

To gain further insights into this behaviour, we analyzed the cumulative distribution of non-zero edge attributions produced by Integrated Gradients, LRP, and Deconvolution (see Figure 8). The results show that Integrated Gradients covers a broader range of attribution values, as its cumulative curves increase more gradually. In contrast, LRP and Deconvolution tend to concentrate their attributions around small values near zero, particularly in denser graphs, indicating underflow issues when assigning edge importance.

We traced back this problem to the dependency of the explanation scores on the model weights. In our white-box framework, this issue can be mitigated by scaling the weights by a fixed factor. Figure 9 shows that when we multiply the weights in the graph convolution layers by a scaling factor $f \leq 1$ and apply the modified model to our datasets, the performance of Deconvolution (after post-processing) improves significantly, reaching levels comparable to Integrated

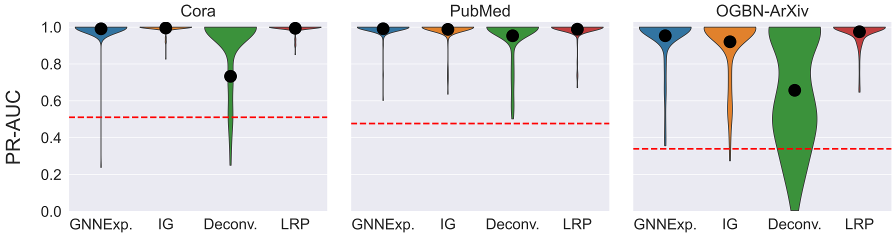


Fig. 6: Violin plot of the PR-AUCs on the three real datasets, across a sample of 100 nodes, with the points corresponding to the mean.

Table 1: Mean PR-AUC across a sample of 100 nodes of the explainers on the real datasets, with standard deviation, we show both the original performance and after the post-processing described in Section 3.4; Darker shades highlight the best results, with different colors used for each dataset.

Dataset	**Post-processing?**	GNNExp.	IG	Deconv.	LRP
Cora	✗	99.1±7.6	99.5±2.2	91.3±18.1	99.4±2.4
	✓	99.5±4.0	100.0±0.0	99.7±2.9	100.0±0.0
PubMed	✗	99.1±4.8	98.8±5.7	95.3±12.1	98.9±5.3
	✓	99.1±4.8	99.8±1.9	98.7±6.6	100.0±0.0
OGBN-ArXiv	✗	95.3±13.3	92.1±16.5	65.7±30.0	97.5±6.5
	✓	95.8±12.6	96.9±10.0	96.2±12.8	99.2±4.1

Gradients. Similar improvements are observed for LRP, not shown because the results overlap those of Deconvolution.

5 Conclusion

This paper introduces a novel, true-to-the-model benchmark for evaluating edge-level explanations of Graph Neural Networks using a white-box model. Our approach provides a controlled and interpretable environment, enabling a systematic comparison of state-of-the-art explainers. Through extensive experiments on both synthetic and real-world graphs, we identify key methodological challenges and offer practical solutions to improve explanation quality.

A critical finding of our work is that some evaluation metrics are more effective in capturing the performance of edge-level explainers. In particular, Precision-Recall curves offer a more sensitive measure of the ability of explainers to distinguish relevant from irrelevant edges, when compared to ROC-AUC scores, particularly for dense input graphs.

Our evaluation revealed two recurrent issues: first, many explainers produce explanations that are inconsistent with the actual computations of the GNN; second, some explainers suffer from numerical instability, which is amplified in

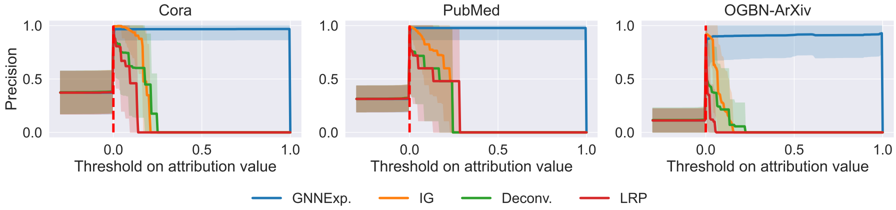


Fig. 7: Plot with precision of wrong difficult negatives at different thresholds for the three real datasets Cora, PubMed, and OGBN-ArXiv

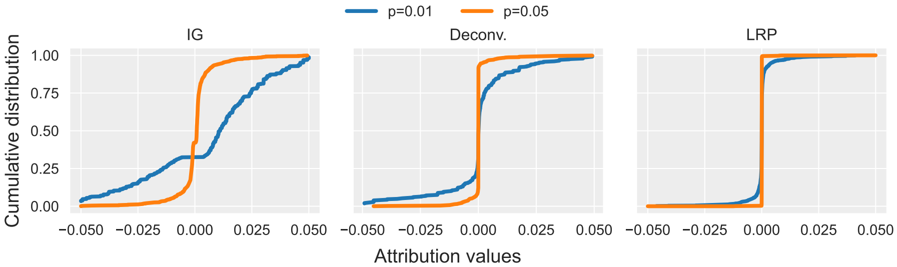


Fig. 8: Cumulative distributions of non-zero edge attributions produced by Integrated Gradients, Deconvolution, and LRP on an Erdős-Rényi graph with 1000 nodes. Results are shown for two edge connection probabilities ($p = 0.01$ and $p = 0.05$), illustrating that Integrated Gradients covers a broader range of attribution values, while Deconvolution and LRP concentrate their attributions near zero, particularly in denser graphs.

denser graphs. To address these issues, we propose two corrective strategies. We introduce a post-processing step that adjusts the attribution scores to better reflect the true message-passing paths, thus mitigating the problem of inconsistent explanations. Additionally, we show that scaling the weights of the model significantly reduces numerical instability, bringing the performance of methods like LRP and Deconvolution closer to that of Integrated Gradients.

Overall, our framework not only benchmarks the strengths and limitations of current edge-level explainers but also provides practical solutions to improve their reliability. Future work may extend this framework to link and graph-level tasks, incorporate other white-box models that implement different interpretable heuristics and mimic attention mechanisms, and inspire the development of new explainers that inherently overcome these issues.

References

1. Amara, K., Ying, Z., Zhang, Z., Han, Z., Zhao, Y., Shan, Y., Brandes, U., Schemm, S., Zhang, C.: Graphframex: Towards systematic evaluation of explainability meth-

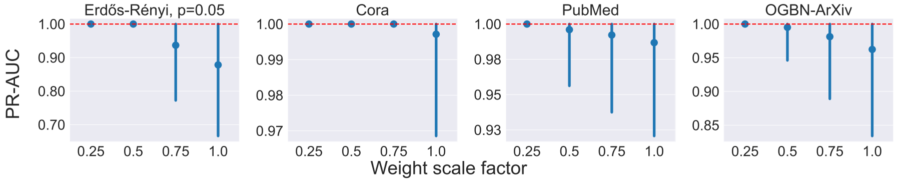


Fig. 9: Effect of weight scaling on the false positive rate of explanations provided by Deconvolution after post-processing. The figure shows how multiplying the graph convolution weights by a scaling factor $f \leq 1$ mitigates underflow issues both for synthetic and real graphs, thereby reducing the false positive rate and improving the numerical stability of the attributions.

ods for graph neural networks. In: Learning on Graphs Conference. pp. 44–1. PMLR (2022)

2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
3. Baldassarre, F., Azizpour, H.: Explainability techniques for graph convolutional networks. arXiv preprint arXiv:1905.13686 (2019)
4. Faber, L., K. Moghaddam, A., Wattenhofer, R.: When comparing to ground truth is wrong: On evaluating GNN explanation methods. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. pp. 332–341 (2021)
5. Funke, T., Khosla, M., Rathee, M., Anand, A.: Zorro: Valid, sparse, and stable explanations in graph neural networks. IEEE Transactions on Knowledge and Data Engineering **35**(8), 8687–8698 (2023)
6. Gregory, S.: Finding overlapping communities in networks by label propagation. New journal of Physics **12**(10), 103018 (2010)
7. Gui, S., Yuan, H., Wang, J., Lao, Q., Li, K., Ji, S.: Flowx: Towards explainable graph neural networks via message flows. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(7), 4567–4578 (2024)
8. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
9. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. arXiv preprint arXiv:2005.00687 (2020)
10. Huang, Q., Yamada, M., Tian, Y., Singh, D., Chang, Y.: Graphlime: Local interpretable model explanations for graph neural networks. IEEE Transactions on Knowledge and Data Engineering **35**(7), 6968–6972 (2023)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
12. Li, X., Wang, J., Yan, Z.: Can graph neural networks be adequately explained? a survey. ACM Comput. Surv. **57**(5) (Jan 2025)
13. Lin, W., Lan, H., Wang, H., Li, B.: Orphicx: A causality-inspired latent variable model for interpreting graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13729–13738 (2022)

14. Longa, A., Azzolin, S., Santin, G., Cencetti, G., Lio, P., Lepri, B., Passerini, A.: Explaining the explainers in graph neural networks: a comparative study. ACM Comput. Surv. **57**(5) (Jan 2025)
15. Monti, C., Bajardi, P., Bonchi, F., Panisson, A., Perotti, A.: A true-to-the-model axiomatic benchmark for graph-based explainers. Transactions on Machine Learning Research (2024)
16. Pope, P.E., Kolouri, S., Rostami, M., Martin, C.E., Hoffmann, H.: Explainability methods for graph convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
17. Rathee, M., Funke, T., Anand, A., Khosla, M.: Bagel: A benchmark for assessing graph neural network explanations. arXiv preprint arXiv:2206.13983 (2022)
18. Schnake, T., Eberle, O., Lederer, J., Nakajima, S., Schütt, K.T., Müller, K.R., Montavon, G.: Higher-order explanations of graph neural networks via relevant walks. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(11), 7581–7596 (2022)
19. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International conference on machine learning. pp. 3319–3328. PMLR (2017)
20. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
21. Vu, M., Thai, M.T.: Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. Advances in neural information processing systems **33**, 12225–12235 (2020)
22. Yang, Z., Cohen, W., Salakhudinov, R.: Revisiting semi-supervised learning with graph embeddings. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 40–48. PMLR, New York, New York, USA (20–22 Jun 2016)
23. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems **32** (2019)
24. Yuan, H., Yu, H., Gui, S., Ji, S.: Explainability in graph neural networks: A taxonomic survey. IEEE transactions on pattern analysis and machine intelligence **45**(5), 5782–5799 (2022)
25. Yuan, H., Yu, H., Wang, J., Li, K., Ji, S.: On explainability of graph neural networks via subgraph explorations. In: International conference on machine learning. pp. 12241–12252. PMLR (2021)
26. Zeiler, M.: Visualizing and understanding convolutional networks. In: European conference on computer vision/arXiv. vol. 1311 (2014)
27. Zhang, H., Wu, B., Yuan, X., Pan, S., Tong, H., Pei, J.: Trustworthy graph neural networks: Aspects, methods, and trends. Proceedings of the IEEE (2024)
28. Zhang, S., Liu, Y., Shah, N., Sun, Y.: Gstarx: Explaining graph neural networks with structure-aware cooperative games. Advances in Neural Information Processing Systems **35**, 19810–19823 (2022)
29. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical Report. CMU-CALD-02-107, Carnegie Mellon University (2002)