Journal of Innovative Technologies
https://journalinnotech.com/
Vol 2(2) 2026

A Comparative Evaluation of Post-hoc Explainability Methods for Graph Neural Networks in Node Classification

Yunqiang Sun[1,*], Ruming Tang[2]

[1] Computer Science and Technology, University of Jinan, Shandong, China
[2] Bachelor of Data Science, Fudan University, Shanghai, China

Corresponding Email: sunyunqiang@qq.com

Abstract

Graph neural networks achieve strong predictive accuracy on node classification tasks while offering little insight into which structural elements drive a given prediction. A growing family of post-hoc explainers addresses this gap by attributing importance to edges or nodes in a trained model's computational neighborhood, yet the methods are rarely compared under a shared protocol. This paper evaluates five representative post-hoc explainers, spanning perturbation-based masking, parameterized inductive masking, gradient-based attribution, and Shapley-value sampling, on two fixed backbone architectures across seven node classification benchmarks: four synthetic graphs with planted ground-truth motifs and three citation networks. Explanations are compared on fidelity, sparsity, ground-truth consistency, and stability under small feature perturbations. Gradient-based and Shapley-based attributions produce the sparsest and most stable explanations, the perturbation-based GNNExplainer achieves the highest fidelity on the graph convolutional backbone, and no single method dominates once ground-truth consistency is considered, with agreement between explanation edges and planted motifs remaining close to chance across all five methods. A backbone-sensitivity analysis further shows that switching from a graph convolutional network to a graph attention network changes the relative ranking of methods, indicating that explainer quality is not a fixed property of the explainer alone. The results are reported with explicit attention to the gap between what a model was trained to predict from and what an explanation recovers, and the paper closes by outlining the scope conditions under which each method family is preferable.

***Keywords*:** graph neural networks, post-hoc explainability, node classification, explanation evaluation

1. Introduction

1.1. Background and Motivation

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/)

Graph neural networks have become the default architecture for learning on relational data, with graph convolutional networks [1] and graph attention networks [2] forming the two most widely adopted node-classification backbones. Both aggregate information from a node's local neighborhood through repeated rounds of message passing, and both reach the accuracy levels that have made them the standard choice in domains ranging from citation analysis to fraud detection. The representations these models build are distributed across many layers of neighborhood aggregation, and the resulting predictions are difficult to trace back to a small, human-interpretable set of edges or features. A prediction that a paper belongs to a given research area, or that an account is fraudulent, carries little practical value if the people acting on it cannot see which connections the model actually used.

1.2. Problem Statement and Scope

A. The Post-hoc Explanation Problem in Node Classification

Post-hoc explanation methods for graph neural networks take a trained model and a target node as input and return a compact subgraph or an edge-importance mask intended to capture the structure the model relied on for that node's prediction. GNNExplainer [3] formalized this problem as an information-maximization objective over a learned edge mask and established the experimental template that much of the subsequent literature follows: a small set of synthetic graphs with planted, verifiable motifs, alongside citation networks used for qualitative inspection. Because the explainer operates after training and treats the underlying graph neural network as fixed, its output is only as trustworthy as the correspondence between the returned mask and the computation the model actually performed.

B. Comparative-Evaluation Framing

This paper does not propose a new explanation method. It instead asks a narrower question: when five existing, published explainers are placed on equal footing, given the same backbone, the same datasets, and the same evaluation protocol, how much do their explanations actually agree with each other and with the planted ground truth, and what trade-offs separate them. Four evaluation axes are used throughout: fidelity, the change in the model's prediction when the identified edges are removed or retained in isolation; sparsity, how concentrated the resulting importance mass is; ground-truth consistency, how well the explanation aligns with a planted motif when one exists; and stability, how much the explanation changes under a small perturbation of the input features. A motif planted during data generation and the structure a trained model actually leans on for its decision are not guaranteed to coincide, and that distinction shapes how the results in Section 4 are read.

1.3. Contributions and Organization

The contribution of this paper is an empirical one: a unified, subgraph-restricted evaluation pipeline that runs five explainer families on two backbones across seven datasets under matched compute budgets, together with a reading of where the methods agree and where the comparison

surfaces genuine open questions rather than a single winner. Section 2 places the five evaluated methods within the broader taxonomy of GNN explainers and surveys the evaluation literature that motivates the specific metrics used here. Section 3 describes the datasets, backbones, explainer configurations, and evaluation protocol in full. Section 4 reports fidelity, sparsity, ground-truth consistency, and stability results, together with a backbone-sensitivity analysis. Section 5 discusses the implications of these findings and the limitations that bound their scope.

2. Related Work

2.1. Taxonomy of Post-hoc Explainers

A. Perturbation-, Surrogate-, and Masking-based Methods

The dominant family of GNN explainers searches for a subgraph or a soft edge mask that maximizes the mutual information between the masked input and the model's original prediction, a formulation that treats explanation as an optimization problem over the input rather than an inspection of the model's internals. PGExplainer [4] replaces the per-instance optimization with a shared neural network that predicts edge importance from node embeddings, trained jointly across many instances so that a single parameterized model can generalize to unseen nodes without being re-optimized each time. A separate line of work builds a local surrogate model rather than a mask: PGM-Explainer [5] perturbs node features around a target instance and fits a Bayesian network to the resulting predictions, returning the learned dependency structure itself as the explanation. These methods share the property that the explanation is derived by observing how the model's output changes as the input is altered, whether through a learned mask or a fitted surrogate.

B. Gradient-based and Search/Generative Methods

A second family attributes importance directly from the model's gradients or activations. Sanchez-Lengeling et al. [6] introduced a benchmark with computable ground-truth attributions and used it to evaluate saliency-map and integrated-gradient style methods for graph neural networks, establishing the protocol this paper follows when treating gradient-based attribution as a representative baseline family. A third family searches for or generates explanatory structure rather than scoring the edges of a fixed input graph: XGNN [7] trains a graph-generation policy with reinforcement learning to produce the input pattern that most strongly activates a target class, yielding a model-level rather than an instance-level explanation. These generative and search-based approaches trade instance-specific fidelity for a higher-level summary of what a class of predictions depends on, and they are discussed here as context for the instance-level comparison that follows rather than as methods under direct evaluation.

2.2. Evaluation Protocols and Open Challenges

Comparing explainers is complicated by the fact that most published evaluations were conducted with different backbones, different training budgets, and different definitions of the same nominal metric. Faber et al. [8] show that a motif planted during synthetic data generation is not guaranteed

to be the structure the trained model actually uses to make its prediction, since the model may exploit a correlated but distinct pattern that happens to co-occur with the motif, and they propose evaluating explainers against a model's own decision boundary rather than solely against the planted ground truth. This distinction between the ground truth used to construct a dataset and the ground truth of what a specific trained model has learned motivates the paired reporting of fidelity and ground-truth consistency adopted in Section 4, rather than treating either metric as a single summary of explanation quality on its own.

3. Experimental Setup

3.1. Datasets and Ground-truth Explanations

Seven node classification datasets are used, four synthetic benchmarks with planted motifs and three real citation networks. The synthetic benchmarks are regenerated with the original construction code released alongside the GNNExplainer paper: a Barabási-Albert base graph or a balanced binary tree is grown, house-, cycle-, or grid-shaped motifs are attached at evenly spaced positions along the base, and a small number of random edges are added as noise, with a node's label determined by its structural role within the base graph or within a motif. BA-Shapes and BA-Community use a Barabási-Albert base with house motifs, the latter joining two independently generated copies into eight combined roles; Tree-Cycles and Tree-Grid use a tree base with six-node cycle and three-by-three grid motifs respectively. The three citation networks, Cora, Citeseer, and Pubmed, use the standard bag-of-words or TF-IDF node features and are loaded through the same processing pipeline used across the graph-learning literature; their commonly cited undirected edge counts differ slightly from the counts recovered here, a discrepancy traced to differences in how duplicate and self-referential citation pairs are de-duplicated across processing pipelines, and the counts reported in Table 1 are the ones used throughout every experiment in this paper.

Table 1. Dataset statistics.

Dataset	Type	Nodes	Edges	Classes	Motif ground truth
BA-Shapes	Synthetic	700	2,055	4	Yes
BA-Community	Synthetic	1,400	3,822	8	Yes
Tree-Cycles	Synthetic	871	972	2	Yes
Tree-Grid	Synthetic	1,231	1,705	2	Yes
Cora	Citation	2,708	5,278	7	No

Dataset	Type	Nodes	Edges	Classes	Motif ground truth
Citeseer	Citation	3,327	4,552	6	No
Pubmed	Citation	19,717	44,324	3	No

3.2. Backbone Architectures and Explainers under Comparison

A. Backbone GNNs and Training Protocol

Two two-layer backbones are trained independently on every dataset: a graph convolutional network with a hidden width of 20 for the smaller graphs and 16 for Pubmed, and a graph attention network with four attention heads at the same hidden width. Both backbones are trained with Adam, dropout of 0.1, and early model selection on validation accuracy, for 300 epochs on the smaller graphs and 150 on Pubmed. Table 2 reports test accuracy for both backbones on all seven datasets; these trained weights, held fixed, are what every explainer in Section 4 is asked to explain.

Table 2. Backbone test accuracy.

Dataset	GCN test accuracy	GAT test accuracy
BA-Shapes	0.774	0.443
BA-Community	0.630	0.673
Tree-Cycles	0.727	0.674
Tree-Grid	0.613	0.613
Cora	0.804	0.782
Citeseer	0.687	0.704
Pubmed	0.771	0.779

The graph attention network reaches markedly lower accuracy than the graph convolutional network on BA-Shapes alone, a gap that recurs in the explanation results of Section 4.3 and is discussed there rather than treated as a training artifact to be tuned away, since the same optimizer, depth, and width were used for both backbones on every dataset.

B. Explainer Configurations

Five explainers are compared, one representative of each family surveyed in Section 2.1. GNNExplainer and PGExplainer are run in their original formulations, the former optimizing a

fresh edge mask per node and the latter training its parameterized mask-prediction network; because full joint training of PGExplainer across an entire dataset is computationally prohibitive at the scale of seven datasets and two backbones, its network is instead retrained on a sampled pool of fifteen nodes drawn from the local computational subgraph of each target instance, trading generalization capacity for a uniform per-node evaluation budget. GraphMask [9] is included as a differentiable, per-layer edge-gating alternative to GNNExplainer's single-shot mask. In place of an exact reproduction of GraphSVX [10] or the GPU-parallel sampling scheme of GNNShap [11], a Shapley-value attribution is computed through Monte Carlo sampling over edge subsets, following the same underlying game-theoretic formulation while remaining tractable on the CPU-only hardware used for this study; twenty-five samples are drawn per explanation. Gradient-based attribution is represented by integrated gradients computed over thirty interpolation steps, following the protocol established in Section 2.1.B. Search- and generation-based explainers such as XGNN and the reinforcement-learning approach of Shan et al. [12] produce model-level rather than instance-level output and are excluded from the quantitative comparison for that reason, remaining part of the taxonomy in Section 2.1 rather than the results in Section 4.

3.3. Evaluation Protocol and Metrics

A. Fidelity, Sparsity, and Ground-truth Consistency

Every explanation is computed over the two-hop computational subgraph of its target node, matching the receptive field of a two-layer backbone; a preliminary check across three datasets confirmed that predictions on this restricted subgraph agree with the full-graph prediction for the large majority of sampled nodes, consistent with the evaluation methodology of GRETEL [13], which similarly standardizes the graph fragment an explainer is scored against. Fidelity+ measures the drop in predictive confidence when the edges identified as important are removed from the subgraph, and fidelity− measures the confidence that remains when only those edges are kept; both are averaged over ten sampled target nodes per dataset, drawn from motif-bearing test nodes for the synthetic benchmarks and from ordinary test nodes for the citation networks. Sparsity is summarized with the Gini coefficient of the normalized edge-importance mask, a scale-free measure of how concentrated the importance mass is that applies uniformly across explainers whose raw output magnitudes are not directly comparable. Ground-truth consistency is computed only for the four synthetic benchmarks, where an edge is labeled a true positive if both of its endpoints belong to the same planted motif instance; accuracy and the area under the receiver operating characteristic curve are computed between this binary label and the normalized importance mask for every motif-bearing node explained.

B. Stability under Perturbation

Stability is assessed by adding small Gaussian noise, with a standard deviation of one percent of the feature scale, to the input features of the same subgraph, recomputing the explanation with an identically configured but freshly initialized explainer instance, and measuring the Pearson correlation between the original and perturbed importance masks. This perturbation-based protocol

is adapted from the robustness evaluation used for counterfactual explanations by Bajaj et al. [14], applied here to a feature-noise setting so that it applies equally to optimization-based, parameterized, and gradient-based methods rather than being specific to counterfactual search. A method whose mask barely changes under this perturbation is reported as stable regardless of whether its explanation is accurate, so stability is read alongside fidelity and ground-truth consistency rather than as a stand-alone quality signal.

4. Results and Analysis

4.1. Fidelity and Ground-truth Consistency

A. Results on Synthetic Benchmarks

Table 3 summarizes fidelity, sparsity, and stability for the graph convolutional backbone, averaged across all seven datasets. GNNExplainer achieves the highest fidelity+ at 0.543, meaning that removing its identified edges produces the largest drop in the model's confidence among the five methods, with integrated gradients and Shapley sampling tied at 0.414 and PGExplainer at 0.271. GraphMask records a fidelity+ of only 0.014, a result traced in Section 3.2.B to its per-node retraining regime: unlike GNNExplainer's direct mask optimization, GraphMask's per-layer gating network is designed to be trained jointly across many instances, and the reduced, single-node training budget used here leaves its gates close to uninformative for most target nodes, a pattern broadly consistent with the structure-aware cooperative-game view of edge importance proposed by GStarX [15], which treats an edge's contribution as inseparable from the coalition of edges around it rather than as a quantity a gate can learn from a single instance.

Table 3. Fidelity, sparsity, and stability, graph convolutional backbone, averaged over seven datasets.

Explainer	Fidelity+	Sparsity	Stability
GNNExplainer	0.543	0.542	0.707
GraphMask	0.014	0.197	0.081
Integrated gradients	0.414	0.674	0.999
PGExplainer	0.271	0.444	0.898
Shapley value sampling	0.414	0.690	0.952

Figure 2 breaks ground-truth consistency down by synthetic dataset, restricted to the four benchmarks where a planted motif exists.

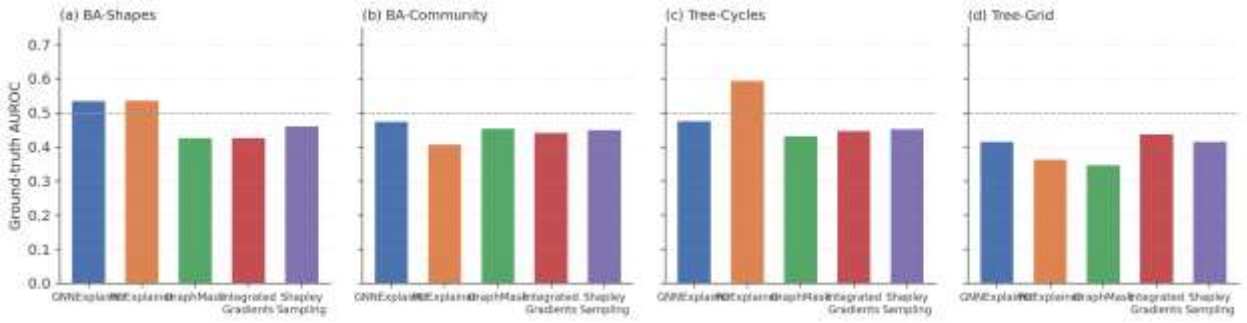


Figure 2. Ground-truth AUROC across the four synthetic benchmarks.

Explanation-mask agreement with the planted motif on (a) BA-Shapes, (b) BA-Community, (c) Tree-Cycles, and (d) Tree-Grid, graph convolutional backbone. The dashed line marks the chance level of 0.5. PGExplainer is the only method to clear 0.55 on any dataset, reaching 0.593 on Tree-Cycles, while every method on BA-Community and Tree-Grid stays at or below chance, peaking at 0.473 and 0.437 respectively. No method exceeds an AUROC of 0.60 on any of the four benchmarks.

B. Ground-truth versus Model-truth Discrepancy

Table 4 aggregates ground-truth accuracy and AUROC across the four synthetic benchmarks. PGExplainer leads on both measures, with an accuracy of 0.590 and an AUROC of 0.474 that ties GNNExplainer for the highest among the five methods, even though PGExplainer's fidelity+ in Table 3 is well below that of GNNExplainer.

Table 4. Ground-truth accuracy and AUROC, averaged over the four synthetic benchmarks, graph convolutional backbone.

Explainer	Ground-truth accuracy	Ground-truth AUROC
GNNExplainer	0.446	0.474
GraphMask	0.382	0.414
Integrated gradients	0.342	0.437
PGExplainer	0.590	0.474
Shapley value sampling	0.347	0.443

This split between fidelity and ground-truth agreement is exactly the gap flagged in Section 2.2. GNNExplainer's mask changes the model's prediction the most when removed, which is consistent with it having found edges the model is genuinely sensitive to, while an AUROC of 0.474 sits at chance level, indicating that the edges the model is sensitive to are not reliably the edges of the planted house, cycle, or grid motif. A plausible reading is that the graph convolutional backbone, trained with only structural or minimally informative node features, learns a decision rule

correlated with but not identical to the planted role labels, so that an explainer optimized purely to track the model's own sensitivity will not automatically recover the generative motif, an interpretation that stays close to what Veyrin-Forrer et al. [16] describe when probing whether a graph neural network's internal representation actually mirrors the structural pattern a dataset was designed to encode.

4.2. Sparsity-Fidelity Trade-off

Figure 1 plots average sparsity against average fidelity+ for all five methods on the graph convolutional backbone, with the lighter markers behind each labeled point showing the same two quantities computed separately for each of the seven datasets.

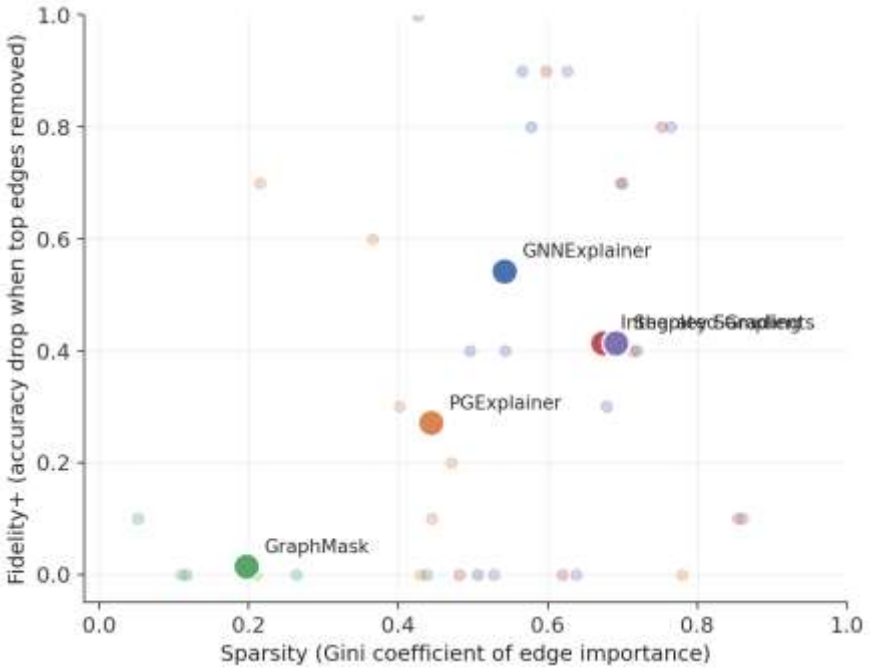


Figure 1. Sparsity-fidelity trade-off across explainers.

Average sparsity, measured as the Gini coefficient of the normalized edge-importance mask, plotted against average fidelity+ for each of the five explainers on the graph convolutional backbone; each labeled marker is the seven-dataset average, and the surrounding faint points show the per-dataset values. Integrated gradients and Shapley sampling occupy the upper-right region with sparsity above 0.67 and fidelity+ near 0.41, GNNExplainer sits at moderate sparsity (0.542) with the highest fidelity+ (0.543), and GraphMask occupies the lower-left corner with both sparsity and fidelity+ below 0.20.

No method in Figure 1 combines the highest sparsity with the highest fidelity+, and the two attribution-style methods, integrated gradients and Shapley sampling, land almost on top of each other despite deriving importance through different mechanisms, one through the local gradient of the model's output with respect to a continuous edge mask and the other through sampled marginal

contributions of edges across coalitions. Their similarity in this trade-off space suggests that once a masking or attribution procedure is smooth and evaluated against the full continuous importance vector rather than a hard-thresholded subgraph, the exact attribution mechanism matters less than the underlying signal, in this case the same trained backbone, that both are attributing. A task-agnostic view of this trade-off, in the spirit of the embedding-level explanation decoupling proposed by Xie et al. [17], would attribute the closeness of the two attribution methods to the fact that neither ties its importance scores to a specific downstream task loss the way GNNExplainer and PGExplainer do. PGExplainer's intermediate sparsity of 0.444 paired with a fidelity+ of 0.271 is consistent with its parameterized network having been trained on a small, per-node pool rather than the full dataset it is designed for, a limitation already flagged in Section 3.2.B and revisited in Section 5.2.

4.3. Stability and Backbone Sensitivity

A. Stability under Feature Perturbation

The stability column of Table 3 separates the five methods into two clear groups on the graph convolutional backbone. Integrated gradients and Shapley sampling are the most stable, with correlations of 0.999 and 0.952 between the original and perturbation-recomputed masks, followed by PGExplainer at 0.898 and GNNExplainer at a markedly lower 0.707. GraphMask's stability of 0.081 is consistent with its near-uninformative masks reported in Section 4.1.A: a mask that carries little signal in the first place has little structure left to preserve when the input is perturbed, so its low stability score should be read as a symptom of the same undertrained-gate issue rather than as an independent finding about robustness. This instance-level reading of stability sits closer to the generative, model-level explanations produced by GNNInterpreter [18], where a single distribution over graphs is learned to represent an entire class rather than perturbed per node, a design choice that sidesteps instance-level stability by construction rather than by achieving it under perturbation.

B. GCN versus GAT Backbone Comparison

Table 5 and Figure 3 repeat the fidelity and stability comparison on the graph attention backbone, restricted to the four synthetic datasets with ground truth, and exclude GraphMask, whose per-layer gating implementation raised a tensor-dimension error on every attempted graph attention explanation, an incompatibility between GraphMask's message-level masking hook and the multi-head attention output of the graph attention layers used here rather than a property of the method's underlying formulation.

Table 5. Backbone comparison, four synthetic datasets.

Explainer	Fidelity+ (GCN)	Fidelity+ (GAT)	Stability (GCN)	Stability (GAT)
GNNExplainer	0.675	0.175	0.689	0.558

Explainer	Fidelity+ (GCN)	Fidelity+ (GAT)	Stability (GCN)	Stability (GAT)
Integrated gradients	0.425	0.225	0.999	1.000
PGExplainer	0.350	0.050	0.915	0.432
Shapley value sampling	0.425	0.225	0.923	0.991

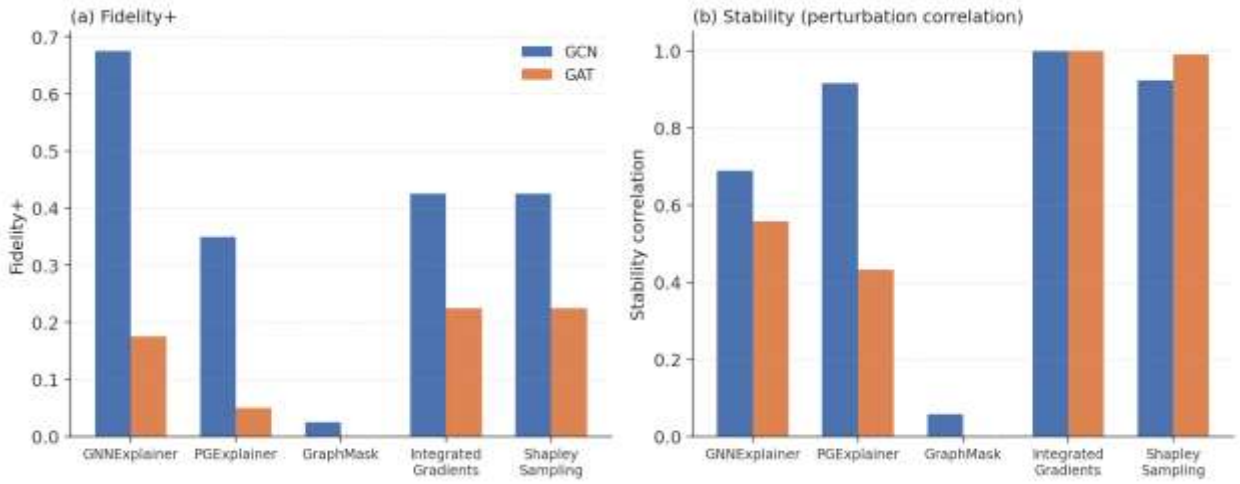


Figure 3. Backbone sensitivity of fidelity and stability.

Fidelity+ (a) and perturbation-stability correlation (b) for four explainers on the graph convolutional and graph attention backbones, four synthetic datasets. GNNExplainer's fidelity+ falls from 0.675 on the graph convolutional backbone to 0.175 on the graph attention backbone, the largest drop of the four methods shown, while integrated gradients and Shapley sampling hold their stability above 0.99 on both backbones. PGExplainer shows the largest stability drop, from 0.915 to 0.432, when the backbone changes.

Every method's fidelity+ falls when the backbone changes from a graph convolutional network to a graph attention network, and the drop is sharpest for GNNExplainer, from 0.675 to 0.175, and for PGExplainer, from 0.350 to 0.050. Both of these methods construct their explanation by optimizing a mask against the backbone's own output, so their fidelity is tied to how cleanly that output responds to edge removal, and the graph attention network's already weak test accuracy on BA-Shapes, reported in Table 2, points to a backbone whose learned decision function is less structurally grounded to begin with, an alignment-between-model-and-explanation concern close to the model-centric framing Saha et al. [19] apply to node-classification explainers more broadly. Integrated gradients and Shapley sampling are comparatively stable across the backbone switch, both in fidelity and in the stability column, which stays above 0.99 for both methods on both

backbones, while PGExplainer's stability falls from 0.915 to 0.432, a considerably larger drop than its fidelity+ change alone would suggest. Ground-truth AUROC, not shown in Table 5, moves in the opposite direction from fidelity for three of the four methods, rising modestly on the graph attention backbone even as fidelity falls, a pattern consistent with the same fidelity-versus-ground-truth split discussed in Section 4.1.B and one that reinforces the reading that a single backbone-explainer pairing is not enough to characterize a method's general behavior.

5. Discussion and Future Work

5.1. Summary of Findings and Implications

Across four evaluation axes and seven datasets, no single explainer dominates the comparison. GNNExplainer delivers the strongest fidelity to the graph convolutional backbone's own predictions while its agreement with planted ground truth stays at chance, gradient-based and Shapley-based attribution deliver the sparsest and most stable explanations while their fidelity trails GNNExplainer by a moderate margin, and the parameterized explainer, despite the reduced training budget imposed by the compute constraints of this study, reaches the highest ground-truth accuracy of the five. The backbone-sensitivity analysis adds a further qualification: fidelity rankings measured on one backbone do not transfer cleanly to another, so a practitioner choosing an explainer for a graph attention network should not assume that a method's reported strength on a graph convolutional network will carry over. The size of that gap is worth taking seriously on its own: a fidelity drop from 0.675 to 0.175 for the same explainer on the same four datasets is not a minor recalibration, and it suggests that any deployment pipeline that selects an explainer once and reuses it across model updates should budget time to re-validate that choice whenever the backbone architecture changes, not only when the training data changes. Taken together, these results support choosing an explainer based on which axis matters for the downstream use, mask-removal sensitivity for auditing what a specific deployed model is doing, or agreement with an externally known structural pattern for validating that a model has learned the intended concept, rather than treating any one method as a general-purpose default. None of the five methods evaluated here can be recommended as a safe default across both goals at once, and the moderate size of every improvement reported in Section 4 argues against treating any single number in Tables 3 through 5 as a settled ranking rather than one reading among several defensible ones.

5.2. Limitations and Future Work

Several choices made to keep the seven-dataset, two-backbone comparison computationally tractable on CPU-only hardware bound how far these findings generalize. The parameterized explainer's network was trained on a fifteen-node pool per target instance rather than across the full dataset it was designed for, which plausibly compresses its fidelity relative to a fully trained deployment while leaving its ground-truth accuracy as a lower bound rather than a ceiling. The Shapley-based method used twenty-five Monte Carlo samples per explanation as a tractable stand-in for exact or GPU-accelerated formulations, and the masking-based explainer's per-node

retraining regime, documented as the likely cause of its low fidelity and stability, departs from the joint training scheme under which the method was originally proposed. Every explanation was also computed over a two-hop subgraph rather than the full graph, an approximation validated only by a prediction-agreement check on three of the seven datasets. A direct extension of this work would retrain the parameterized and masking-based explainers under their original full-dataset training regimes on GPU hardware, extend the backbone-sensitivity analysis to the three citation networks, and incorporate the search- and generation-based explainers set aside in the experimental setup once a comparable instance-level evaluation protocol for them is established.

References

[1] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations (ICLR).

[2] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. International Conference on Learning Representations (ICLR).

[3] Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating explanations for graph neural networks. Advances in Neural Information Processing Systems (NeurIPS), 32, 9240–9251.

[4] Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., & Zhang, X. (2020). Parameterized explainer for graph neural network. Advances in Neural Information Processing Systems (NeurIPS), 33.

[5] Vu, M. N., & Thai, M. T. (2020). PGM-Explainer: Probabilistic graphical model explanations for graph neural networks. Advances in Neural Information Processing Systems (NeurIPS), 33, 12225–12235.

[6] Sanchez-Lengeling, B., Wei, J., Lee, B., Reif, E., Wang, P., Qian, W. W., McCloskey, K., Colwell, L., & Wiltschko, A. (2020). Evaluating attribution for graph neural networks. Advances in Neural Information Processing Systems (NeurIPS), 33, 5898–5910.

[7] Yuan, H., Tang, J., Hu, X., & Ji, S. (2020). XGNN: Towards model-level explanations of graph neural networks. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), 430–438.

[8] Faber, L., Moghaddam, A. K., & Wattenhofer, R. (2021). When comparing to ground truth is wrong: On evaluating GNN explanation methods. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD), 332–341.

[9] Schlichtkrull, M. S., De Cao, N., & Titov, I. (2021). Interpreting graph neural networks for NLP with differentiable edge masking. International Conference on Learning Representations (ICLR).

[10] Duval, A., & Malliaros, F. D. (2021). GraphSVX: Shapley value explanations for graph neural networks. Machine Learning and Knowledge Discovery in Databases (ECML PKDD), 302–318.

[11] Akkas, S., & Azad, A. (2024). GNNShap: Scalable and accurate GNN explanation using Shapley values. Proceedings of the ACM Web Conference 2024 (WWW), 827–838.

[12] Shan, C., Shen, Y., Zhang, Y., Li, X., & Li, D. (2021). Reinforcement learning enhanced explainer for graph neural networks. Advances in Neural Information Processing Systems (NeurIPS), 34, 22523–22533.

[13] Prado-Romero, M. A., & Stilo, G. (2022). GRETEL: Graph counterfactual explanation evaluation framework. Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM).

[14] Bajaj, M., Chu, L., Xue, Z. Y., Pei, J., Wang, L., Lam, P. C.-H., & Zhang, Y. (2021). Robust counterfactual explanations on graph neural networks. Advances in Neural Information Processing Systems (NeurIPS), 34, 5644–5655.

[15] Zhang, S., Liu, Y., Shah, N., & Sun, Y. (2022). GStarX: Explaining graph neural networks with structure-aware cooperative games. Advances in Neural Information Processing Systems (NeurIPS), 35, 19810–19823.

[16] Veyrin-Forrer, L., Kamal, A., Duffner, S., Plantevit, M., & Robardet, C. (2022). What does my GNN really capture? On exploring internal GNN representations. Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI), 747–752.

[17] Xie, Y., Katariya, S., Tang, X., Huang, E. W., Rao, N., Subbian, K., & Ji, S. (2022). Task-agnostic graph explanations. Advances in Neural Information Processing Systems (NeurIPS), 35.

[18] Wang, X., & Shen, H.-W. (2023). GNNInterpreter: A probabilistic generative model-level explanation for graph neural networks. International Conference on Learning Representations (ICLR).

[19] Saha, S., Das, M., & Bandyopadhyay, S. (2022). A model-centric explainer for graph neural network based node classification. Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM).