Advances in Artificial Intelligence and Machine Learning; Research 3 (2) 1165-1179 Received 17-04-2023; Accepted 15-06-2023; Published 22-06-2023

SEEN: Sharpening Explanations for Graph Neural Networks Using Explanations From Neighborhoods

Hyeoncheol Cho hcheol35.cho@samsung.com
Samsung SDS
Seoul, Republic of Korea

Youngrock Oh * yrock.oh@gmail.com
Mobilint
Seoul, Republic of Korea

Eunjoo Jeon ej85.jeon@samsung.com
Samsung SDS
Seoul, Republic of Korea

Corresponding Author: Hyeoncheol Cho

Abstract

Explaining the foundations for predictions obtained from graph neural networks (GNNs) is critical for credible use of GNN models for real-world problems. Owing to the rapid growth of GNN applications, recent progress in explaining predictions from GNNs, such as sensitivity analysis, perturbation methods, and attribution methods, showed great opportunities and possibilities for explaining GNN predictions. In this study, we propose a method to improve the explanation quality of node classification tasks that can be applied in a post hoc manner through aggregation of auxiliary explanations from important neighboring nodes, named SEEN. Applying SEEN does not require modification of a graph and can be used with diverse explainability techniques due to its independent mechanism. Experiments on matching motif-participating nodes from a given graph show great improvement in explanation accuracy of up to 12.71% and demonstrate the correlation between the auxiliary explanations and the enhanced explanation accuracy through leveraging their contributions. SEEN provides a simple but effective method to enhance the explanation quality of GNN model outputs, and this method is applicable in combination with most explainability techniques.

Keywords: Explainable AI, Explanation Enhancement, Explainability Technique, Graph Neural Network

* This work was done when the author worked at Samsung SDS.

Citation: Hyeoncheol Cho, et al. SEEN: Sharpening Explanations for Graph Neural Networks Using Explanations From Neighborhoods. Advances in Artificial Intelligence and Machine Learning. 2023;3(2):68.

1. INTRODUCTION

Learning and extracting information from graph structures are considered important but challenging, due to the difficulty of modeling relational information, even extending to long-range interactions among nodes [1–3]. Graph neural networks (GNNs), which are specially designed deep neural network for learning topologies and features from graphs, have revolutionized the field of machine learning on graph-structured data and achieved state-of-the-art performances [2, 4–6]. Recent progression on GNN architecture represented by a message-passing scheme, which recursively generates, aggregates, and updates node representations based on a local connectivity with neighborhoods, have generalized existing architectures and extended the applicability of GNNs to complex graph-structured problems [1, 7, 8].

Understanding why such decisions are made by GNNs improves the transparency of the models, helps to identify the failure modes and provides hints to revise the models. Additionally, providing human-understandable explanations for GNN predictions is highly important for reliability and trustworthiness of GNNs, which are essential for critical applications requiring credible predictions, such as medical uses. However, compared to the rapidly growing success of GNNs on graph-related tasks, explainability on GNN predictions has been less explored [9, 10]. There have been several successes on transferring explainability techniques developed for explaining convolutional neural networks (CNNs) to GNNs with minor modifications [9, 11–13]. Recently, graph-oriented explainability methods have also been proposed to explain a prediction by extracting essential subgraphs from the input without changing the prediction [14–17]. Generally, these methods provide explanations in the form of contribution scores for each component in the input graph for a given decision of a GNN. Although those methods are designed to highlight important nodes or edges of the input graph for the target decision, their explanations often require additional models to be trained for generating graph masks [14, 15].

We propose a method named Sharpening Explanations for graph neural networks using Explanations from Neighborhoods, abbreviated as SEEN. Given a prediction to be explained and an explainability method, SEEN enhances the target explanation by aggregating the auxiliary explanations from the assistant nodes near the target node. Because graphs are used to represent relations between two nodes, we can assume that there is a strong correlation between the explanation of the target node and the explanations for its neighbors. Specifically, we conjecture that given a pair of nodes, if the first node has a significant influence on the prediction of the second node, the explanation of the prediction of the former would be positively correlated to the target explanation to some extent. In this regard, SEEN aggregates the auxiliary explanations by determining their importance weights based on the contribution scores of the corresponding nodes in the target explanation. Since acquiring auxiliary explanations does not require alteration on neither the input graph nor the trained model, SEEN is capable of providing sharpened but still intact explanations based on the original model and data. Moreover, it is worth noting that SEEN can be applied in a post-hoc manner with diverse explainability techniques on node classification tasks due to its independence in the way that individual explanations are generated. We evaluated SEEN on widely used synthetic datasets for measuring explanation accuracy on node classification. Our qualitative and quantitative evaluations demonstrated that applying SEEN can significantly improve explanation accuracy compared to the original evaluations.

2. RELATED WORK

Graph neural networks The foundation of GNNs has been presented by Bruna et al. [4], based on a spectral graph theory and has been expanded by numerous reports, such as Defferrard et al. [5], Kipf et al. [2], and Duvenaud et al. [18]. The message-passing scheme proposed by Gilmer et al. [7], which generalizes the GNN mechanism in terms of message, update, and readout functions, represents important progress on GNN formulations. Most modern GNNs fall into the message-passing formulation, including graph convolutional network (GCN) [2], graph attention network (GAT) [6], GraphSAGE [1], and graph isomorphism network (GIN) [8], which show outstanding performance on graph-structured data. Advances in GNN models have made GNNs a favored model for various graph-related tasks, including node classification [1, 2, 6], graph classification [7, 8, 18], and link prediction [19, 20]. In our paper, we utilize the GCN architecture as the model system to be explained when evaluating SEEN, based on its popularity in learning graph-structured data and expandability to diverse message-passing GNNs without losing generality.

Explainability methods for GNN There has been an increasing number of works that study explainability methods for deep neural networks, especially CNNs: gradient/feature-based methods [21–24], perturbation-based methods [25–27], and decomposition methods [28, 29]. Inspired by these studies, the interpretability of GNNs has also been addressed by similar approaches [10]. Gradient/feature-based methods are proposed, including Sensitivity Analysis (SA) [13], Guided Backpropagation [13], Class Activation Mapping (CAM) [9], and Gradient-weighted Class Activation Mapping (Grad-CAM) [9]. Several perturbation-based methods are proposed including GNNExplainer [14], PGExplainer [15], ZORRO [30], GraphMask [16], and Causal Screening [31]. Decomposition-based methods are also applied to explain the deep graph neural networks, including Layerwise Relevance Propagation (LRP) [13], Excitation Backpropagation [9] and GNN-LRP [32].

On the other hand, a few papers have proposed methods to enhance explanations of a given explainability method for CNNs [33, 34]. In general, explanation enhancement methods make copies of the input image with a small perturbation and incorporate the explanations for them to give a better explanation for the target prediction. For example, SmoothGrad [33] takes random samples in the neighborhood of the input and average the explanations of the samples. Enhancing Visual Explanations using Image Transformations (EVET) [34] provides a visually clear explanation that takes into account geometric transformations of the input image. SEEN is similar to them in the way that we incorporate the auxiliary explanations to sharpen the target explanation. However, SEEN do not modify the input graph, instead it gathers neighboring nodes from the graph and integrates the explanations for them, whereas Smoothgrad and EVET modify the input image to obtain the auxiliary explanations.

3. PRELIMINARIES

Graph neural networks Define a graph $\mathcal{G} = (V, E)$ by a set containing N nodes $V = \{v_1, ..., v_N\}$ and a set containing M edges $E = \{e_1, ..., e_M\}$. A GNN model Φ takes a graph $\mathcal{G}$ as an input and performs node-level, graph-level, or edge-level predictions through a series of message-passing layers and a pooling layer if required. The input graph $\mathcal{G}$ can be presented by three matrices, an adjacency matrix $A \in \{0, 1\}^{N \times N}$, a node feature matrix $X_v \in \mathbb{R}^{N \times D}$, and an optional edge feature

matrix $X_e \in \mathbb{R}^{N \times N \times K}$ if edge features are provided, where D and K denote the number of features for nodes and the number of features for edges, respectively.

Each layer of GNNs based on the message-passing scheme can be divided into three steps: message generation, neighborhood aggregation, and representation update steps [7]. In the message generation step, the message function Message takes the node representations h_i and h_j of edge (v_i, v_j) and its edge representation h_{ij} from previous layer and calculates message $m_{ij} = \text{Message}\left(h_i, h_j, h_{ij}\right)$. The neighborhood aggregation step collects messages through the aggregate function Aggregate from neighboring nodes $\mathcal{N}_i$ of v_i, generating aggregated message $m_i' = \text{Aggregate}\left(\{m_{ij} \mid v_j \in \mathcal{N}_i\}\right)$. The representation update step merges aggregated message m_i' with previous representation h_i through the Update function. The updated representation $h_i' = \text{Update}\left(h_i, m_i'\right)$ becomes the node representation for node v_i in the current layer and is propagated to further layers.

Explaining GNN predictions We focus on explaining a node classification model, which is the target scope of this paper. Given a graph $\mathcal{G}$, a node classification model $\Phi : \mathcal{G} \rightarrow y$, and an target node v_t, an explainability technique estimates an explanation $S(v_t)$ for a model prediction $\Phi\left(G, v_t\right)$. An explanation $S(v_t)$ can be either a set of node scores $S(v_t) = \{s_v \mid v \in V\}$ or a set of edge scores $S(v_t) = \{s_e \mid e \in E\}$, depending on the explainability methods.

4. SEEN: SHARPENING EXPLANATIONS USING EXPLANATIONS FROM NEIGHBORHOODS

In this section, we introduce our explanation sharpening method using neighborhood explanations for enhancing the explanation quality on node classification tasks. Our method, SEEN, accumulates auxiliary explanations from assistant nodes of a target node, without modifications on an input graph. Given the explanation target node v_t, our method collects the set of assistant nodes $V_a = \{v_a \mid v_a \neq v_t, v_a \in V\}$ from the input graph $\mathcal{G}$ and performs auxiliary explanation for each assistant node v_a. Obtained auxiliary explanations are aggregated with the original explanation from the target node, considering their importance on the target prediction. It should be noted that SEEN is a post hoc process that is attachable to the original explainability techniques and that SEEN gathers additional information and updates the original explanation. The overview for the SEEN is depicted in FIGURE 1. We begin this section with the detailed process for the auxiliary explanation acquisition (Section 4.1) and then discuss importance-based explanation aggregation for sharpening the original explanation with the auxiliary explanations (Section 4.2). We discuss the motivation on our explanation sharpening method (Section 4.3).

4.1 Auxiliary explanation acquisition

The first step of the auxiliary explanation acquisition is the selection of assistant nodes from which to collect explanations. Selecting an appropriate set of assistant nodes V_a is crucial for obtaining helpful auxiliary explanations. To collect meaningful and supportive assistant nodes, we set a distance-based boundary on the assistant node pool. Distance-based boundary methods measure the number of edges in the shortest path between a candidate node and a target node to exclude unnecessary nodes for explanation, which acts as a boundary for sampling assistant nodes from an

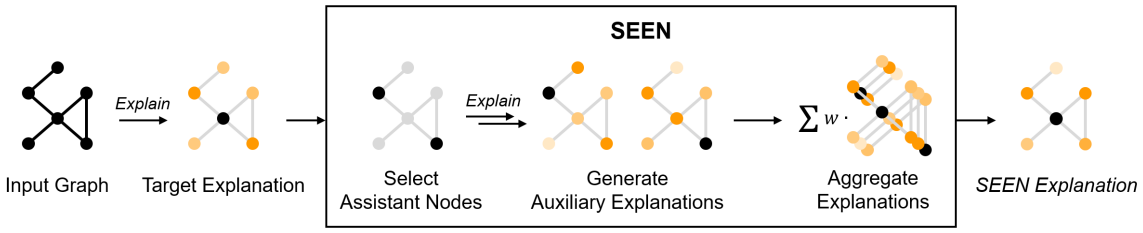


Figure 1: Schematic for sharpening the explanation with SEEN on node classification. With the input graph, trained model and explanation target, SEEN performs explanation sharpening through selecting assistant nodes, generating auxiliary explanations from the assistant nodes, and aggregating the target and auxiliary explanations.

input graph. We filter the nodes outside of the k-hop neighborhood of the target node v_t when k message-passing layers exist in a model, instead of considering all nodes within the entire graph. The nodes outside of the k-hop neighborhood have zero influences on the prediction for the target node, which is the main focus to be explained, and thus, is considered to be inappropriate for collecting auxiliary explanations for the target.

When the assistant nodes are prepared, the explainability technique is applied to generate explanations for the assistant nodes. It should be noted that explaining the assistant nodes is not perfectly identical to explaining the target node. Given the explainability technique Explain, target node v_t, and model prediction y_t, the logit y^c with the most probable class label c, is utilized to explain the target node. On the other hand, to explain the assistant node v_a, the logit y_a^c with the same class label c is employed, regardless of the predicted class for v_a:

$$S(v_t) = \text{Explain}\left(\mathcal{G},\, v_t\right) = \text{Explain}\left(\mathcal{G},\, y_t^c\right), \quad y_t^c = \Phi\left(G,\, v_t\right)^c \tag{1}$$

$$S(v_a) = \text{Explain}\left(\mathcal{G},\, v_a\right) = \text{Explain}\left(\mathcal{G},\, y_a^c\right), \quad y_a^c = \Phi\left(G,\, v_a\right)^c \tag{2}$$

where Explain is the explainability technique, c is the true class label for v_t, and $\Phi(G, v)^c$ is the logit for model prediction on v with class c. Using the logit with the same class label for explanation allows sharing explanations across the nodes.

4.2 Explanation aggregation

With the explanations $S(v_t)$ and $\{S(v_a) \mid v_a \in V_a\}$ that are obtained from the target node v_t and assistant node V_a, respectively, a summarized explanation $\bar{S}(v_t)$ is calculated as a final explanation for the target node v_t. When aggregating explanations, the choice of aggregation formula can be diverse. Here, we hypothesize that the auxiliary explanation obtained from the important assistant node would be more influential and supportive for the target explanation. In detail, we assign a high weight to the auxiliary explanation generated by the assistant node that had high importance in the original target explanation, whereas a low weight is assigned to the auxiliary explanation from the low-importance assistant node. To model the importance-based weighted summation of auxiliary explanations, we design our aggregation formula to incorporate two coefficients, α and β, for modeling the significance of auxiliary explanations and exponentially decaying weight with respect to the importance ranking of assistant nodes. It is possible to use an arbitrary decaying series to model the decaying weights. We chose a simple, exponentially decaying series for our study to

examine the efficacy of our system (Equation 3).

$$\bar{S}(v_t) = S(v_t) + \alpha \sum_{r=1}^{|V_a|} \beta^{r-1} S(v^{(r)}) \tag{3}$$

where $|V_a|$ is the number of assistant nodes within the assistant node set, $v^{(r)}$ is the rth assistant node by the decreasing importance score from the target explanation $S(v_t)$, α is the weight coefficient for auxiliary explanations in range $[0, 1]$, and β is the decay coefficient in range $[0, 1]$ for addressing auxiliary explanations with low importance scores. The coefficients α and β are shared within a graph in which identical values are applied to explain all nodes for a given dataset and explainability technique combination.

The equation can be simplified in particular combinations of α and β; for example, when α is set to 0, the equation disregards all auxiliary explanations similar to common explanation techniques. If $\alpha > 0$ and $\beta \to 1$, the equation models equivalent handling of auxiliary explanations regardless of the importance of its origin node:

$$\bar{S}(v_t) \approx S(v_t) + \alpha \sum_{v_a \in V_a} S(v_a) \tag{4}$$

4.3 Motivation

The intuition behind the collection of auxiliary explanations of SEEN is to gather information across the receptive fields of GNNs and earn a (locally) shared explanation. Each message-passing layer of a GNN transfers information from each node to their neighboring nodes, expanding the receptive field by one hop. With a k-layered GNN, the receptive field corresponds to the k-hop neighborhood of the prediction target. Because all nodes within the k-hop contribute to the prediction for a target node, a bounding assistant node pool is needed in the k-hop neighborhood.

Within the receptive field, SEEN aggregates auxiliary explanations in order of their node importance from the target explanation $S(v_t)$ to refine explanations through overlaying neighborhood explanations. In terms of a community detection problem that predicts each node to certain classes, predicting the class for a node near the community boundary would suffer from ambiguity of the boundary criteria. Neighborhoods of the node that partially share the criteria can help to sharpen the boundary to decide to which community the target belongs, due to the local proximity between the target node and neighborhoods. Accumulating and overlaying the boundary criteria within the local neighborhoods would refine the original criteria and can be extended to sharpening explanations for the predictions. Based on this idea, we conjecture that the explanation could be sharpened by accumulation of auxiliary explanations from neighboring nodes that have a nonzero contribution to the target prediction.

5. EXPERIMENTS

To assess the effectiveness of our conjecture, we conducted a series of experiments on explaining predictions from graph neural networks in node classification tasks. First, we describe datasets (Section 5.1) and the model employed for training and explaining graph neural networks (Section 5.2).

Second we present explainability techniques to be examined (Section 5.3) and evaluation criteria (Section 5.4) for measuring the performance of SEEN. Through qualitative and quantitative analyses, we demonstrate that our method can improve the explanation accuracy to a maximum of 12.71% in the best performing case, without losing accuracy in the least-performing cases. Screening the aggregation coefficients α and β shows that our conjecture, which seeks sharpening explanations through accumulation of explanations from neighborhoods, is valid. Moreover, the screening process also reveals that a certain trend exists for coefficient-performance relation and assigning appropriate values is highly important for maximizing the sharpening effect of SEEN.

5.1 Datasets

We utilize four widely used public synthetic datasets for evaluating explainability techniques on node classification tasks constructed by Rex et al. [14]: BA-Shapes, BA-Community, Tree-Cycles, and Tree-Grid. All datasets pursue the classification of each node into their role in attached motifs, including not-participating positions. Note that ground-truth explanations are clearly defined on the datasets, whereas real-world datasets do not, which allows direct measurement of explanation accuracies by comparing with their ground-truth explanation. No node features are provided for any datasets, except for the BA-Community dataset. Data splits for each dataset are taken from the code by Dongsheng et al. [15], which divides the dataset into 80%, 10%, and 10% portions for training, validation, and testing, respectively.

5.2 Model

All of experiments are conducted with a GCN model [2] for node classification with three message-passing layers. The GCN model is trained with cross-entropy loss for ten different random seed values to generate ten individually trained models. The trained models are then frozen and shared by all explanation experiments. Detailed architecture, training hyperparameters, and averaged model accuracies for the GCN model are provided in Appendix.

5.3 Explainability methods

To evaluate the explanation sharpening performance of SEEN, we adopt the following three GNN explainability techniques, which are compatible with explaining node classification tasks: sensitivity analysis (SA) [9, 13, 21], Grad*Input [29, 35], and GradCAM [9, 23, 35]. Explainability methods based on gradients and features are particularly well-suited for SEEN due to their fast and no-training-required characteristics. However, we would like to emphasize that SEEN is not associated with specific techniques for the enhancement target. When calculating gradients was required for explaining assistant nodes, we measured te gradient of the logit that belongs to the same class as the target node is predicted to be, as previously mentioned in Equation 2 in Section 4.1.

Sensitivity Analysis We utilize the basic SA method [21], which calculates the gradient of the prediction with respect to the features and generates scores by taking the absolute value of the gradient. SA measures the influence of each input value on the final prediction through the gradient and

assumes that higher absolute gradient values indicate the higher importance. Due to its simplicity, transferring SA to GNNs can be easily done by calculating the gradient with respect to the node features instead of pixels [9, 13]. It is possible to apply SA on explaining predictions with scoring both nodes and feature elements, we focus on scoring each node in this paper.

Grad*Input Similar to the SA on GNNs, Grad*Input [29, 35] on GNNs calculates the explanation through element-wise multiplication of node features and their gradient over the prediction and the consequent summation over the feature dimension to generate node scores. Compared to the SA, the gradient from Grad*Input is directly multiplied to the input, and the absolute value is obtained after the multiplication.

GradCAM GradCAM [23] generalizes the class activation map (CAM) method [22], which requires a global average pooling layer in the model architecture, with gradients from each layer. Node representations from intermediate message-passing layers are element-wisely multiplied with their gradients over prediction and summed into a single score value. A detailed formulation for applying GradCAM on GNNs varies from reported papers [9, 35], such as the selection of the target layer, averaging the gradients node-wisely or taking the absolute value; we take an approach similar to the GradCAM(all) method applied by Benjamin et al. [35], which multiplies the gradient and features element-wisely and averages the scores from each layer. Absolute values are obtained at the final stage of the explanation.

5.4 Evaluation

We perform experiments with all combinations of the aforementioned four datasets and three explainability techniques to quantify the explanation sharpening ability of SEEN. Since the ground-truth explanation (the binary labels whether each node is motif-participating) is available for the datasets due to their synthetic nature, we measure the area under the receiver operation characteristic curve (AUC-ROC) between the ground-truth and the obtained explanations, similar to Rex et al. [14] and Benjamin et al. [35]. Based on the measure, we calculate the difference in the explanation accuracy with or without SEEN on each combination.

It should be noted that a modification is made on the test set of the datasets when compared to the original protocol by Rex et al. Originally, an evaluation is conducted only for the set of selected nodes, one node per one motif. Instead, we perform an evaluation for all nodes participating in the motifs instead of using the selected node set. For example, all five nodes that construct a house-structured motif in the BA datasets are accounted for in the evaluation target, rather than utilizing only the left top node of each house-structured motif [36].

5.5 Results

Evaluation on explanation sharpening TABLE 1 shows the change in the explanation accuracy when SEEN is applied to different pairs of datasets and explainability techniques. The significance of the difference is analyzed with a one-sided paired t-test or a one-sided Wilcoxon signed-rank test based on the normality test results. Statistical analyses show that SEEN significantly increases the

accuracy on providing motif-participating nodes as an explanation for the predictions, to a maximum of 12.71% on the BA-Community dataset. In the least-performing cases, the measured explanation accuracy is equivalent to the original explanation when statistically analyzed. An extended table with the standard deviations, p-values, and aggregation coefficients is included in Appendix. These results indicate that SEEN successfully generates a sharpened explanation with higher accuracies through collecting supportive auxiliary explanations around the target node.

Table 1: Explanation accuracies (AUC-ROC) before and after applying SEEN. The best performing accuracies among the combinations of α and β are listed for each dataset and explainability technique. Differences that are not significant by statistical analyses are marked as n.s.

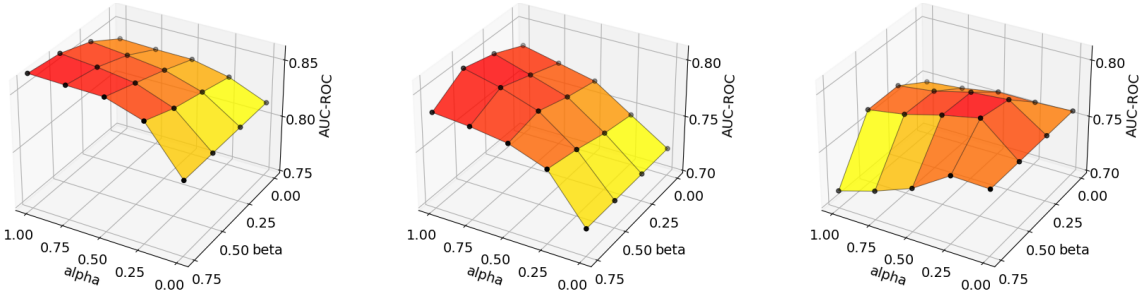
Explainability Methods	**BA-Shapes**	**BA-Community**	**Tree-Cycles**	**Tree-Grid**
SA	0.935	0.625	0.886	0.814
SA + SEEN	0.938	0.637	0.900	0.866
Improvement	*n.s.*	1.85%	*n.s.*	*n.s.*
Grad*Input	0.925	0.614	0.894	0.722
Grad*Input + SEEN	0.936	0.618	0.934	0.800
Improvement	1.18%	0.52%	4.46%	10.77%
GradCAM	0.904	0.573	0.750	0.757
GradCAM + SEEN	0.918	0.645	0.756	0.785
Improvement	1.62%	12.71%	*n.s.*	*n.s.*

Analysis on explanation aggregation coefficients Our aggregation design leverages the amount of auxiliary explanation to be incorporated with the target explanation through two coefficients, α and β. To understand the performance trend with respect to the coefficients, we perform a grid scan over the coefficients by a 0.25 increment within $[0, 1]$ for α and $[0, 1)$ for β and measure the explanation accuracy. The $\beta = 1$ configurations are additionally tested but excluded from the results for visual clarity due to the dramatic decrease in performance compared to the other configurations (data not shown). The low performance for $\beta = 1$ suggests that equivalently taking into account all auxiliary explanations is not beneficial for sharpening the explanation. Thus, assigning an appropriate series of decreasing weights when aggregating explanations is necessary for accuracy enhancement.

The accuracy trends for Tree-Grid dataset (FIGURE 2) show that maximum performance is achieved in high α and medium-to-high β, indicating the helpfulness of auxiliary explanations toward the target explanation. The performance trends for all combinations are depicted in Appendix. Moreover, a similar performance trend is observed in other datasets, proving that our hypothesis on sharpening an explanation with importance-ranked neighborhood explanations. Note that accuracies change even up to 0.1 in the AUC-ROC upon the choice of α and β. We recommend conducting a parameter scan to obtain a higher performance increase with SEEN.

An interesting observation is that GradCAM shows distinctive performance patterns and that the maximum performance is obtained in low α and medium β, compared to other methods that generally reach their maximum at high α and high β. We assume that the difference may be attributable to the difference in handling explanation-hop relations; GradCAM collects explanations from each

and all message-passing layers/hops of GNN, whereas SA and Grad*Input generate explanation within the input layer, which is basically 0-hop. The performances for GradCAM with $\alpha = 0.25$ are still shown to be higher than those with $\alpha = 0$, suggesting that auxiliary explanations are supportive in all cases.



(a) SA (b) Grad*Input (c) GradCAM

Figure 2: Effects of aggregation coefficients on explanation accuracy for Tree-Grid dataset. The results for the remaining combinations are listed in the Appendix.

6. CONCLUSION

In this paper, we propose SEEN, an explanation sharpening method using auxiliary explanations obtained from neighboring nodes. Through aggregation of auxiliary explanations with importance-based weights, SEEN generates a neighborhood-shared explanation without a modification of the data or an extra model training for explanation enhancement. Our experiments show that our simple method can significantly increase the explanation accuracy, in conjunction with the open choice for the explainability technique. Moreover, accuracy trends of aggregation coefficients confirm that our strategy to assign high weights on important nodes is suitable for sharpening explanations by overlaying auxiliary explanations. We expect that our approach would improve the reliability of GNN predictions and expand the scope of the explainability technique to various GNN applications.

7. ACKNOWLEDGMENT

We thank Taehee Lee for his helpful comments. Funding in direct support of this work was obtained from Samsung SDS.

References

[1] Hamilton W, Ying Z, Leskovec J. Inductive Representation Learning on Large Graphs. Adv Neural Inf Process Syst. 2017;30:1025-35.

[2] Kipf TN, Welling M. Semi-supervised Classification With Graph Convolutional Networks. In: International Conference on Learning Representations. 2017.

[3] Xu D, Cheng W, Luo D, Liu X, Zhang X. Spatio-Temporal Attentive RNN for Node Classification in Temporal Attributed Graphs. In: IJCAI; 2019:3947-3953.

[4] Bruna J, Zaremba W, Szlam A, LeCun Y. Spectral Networks and Deep Locally Connected Networks on Graphs. In: International Conference on Learning Representations. 2014.

[5] Defferrard M, Bresson X, Vandergheynst P. Convolutional Neural Networks on Graphs With Fast Localized Spectral Filtering. Adv Neural Inf Process Syst. 2016;29.

[6] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, et al. Graph Attention Networks. In: International Conference on Learning Representations. 2018.

[7] Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural Message Passing for Quantum Chemistry. In: International conference on machine learning. 2017:1263-1272.

[8] Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? In International Conference on Learning Representations. 2019.

[9] Pope PE, Kolouri S, Rostami M, Martin CE, Hoffmann H. Explainability Methods for Graph Convolutional Neural Networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019:10764-10773.

[10] Yuan H, Yu H, Gui S, Ji S. Explainability in Graph Neural Networks: A Taxonomic Survey. IEEE Trans Pattern Anal Mach Intell. 2023;45:5782-5799.

[11] Liu H, Xu J, Wu Y, Guo Q, Ibragimov B, et al. Learning Deconvolutional Deep Neural Network for High Resolution Medical Image Reconstruction. Inf Sci. 2018;468:142-154.

[12] Liu H, Wang H, Wu Y, Xing L. Superpixel Region Merging Based on Deep Network for Medical Image Segmentation. ACM Trans Intell Syst Technol. 2020;11:1-22.

[13] Baldassarre F, Azizpour H. Explainability Techniques for Graph Convolutional Networks. In: International Conference on Machine Learning (ICML) Workshops Workshop on Learning and Reasoning with Graph-Structured Representations. 2019;2019.

[14] Ying R, Bourgeois D, You J, Zitnik M, Leskovec J. Gnnexplainer: Generating Explanations for Graph Neural Networks. Adv Neural Inf Process Syst. 2019;32:9240-51.

[15] Luo D, Cheng W, Xu D, Yu W, Zong B, et al. Parameterized Explainer for Graph Neural Network. Adv Neural Inf Process Syst. 2020;33:19620-19631.

[16] Schlichtkrull MS, De Cao N, Titov I. Interpreting Graph Neural Networks for NLP With Differentiable Edge Masking. 2020. ArXiv preprint: https://arxiv.org/pdf/2010.00577

[17] Vu M, Thai MT. Pgm-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks. Adv Neural Inf Process Syst. 2020;33:12225-12235.

[18] Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, et al. Convolutional Networks on Graphs for Learning Molecular Fingerprints. Adv. Neural Inf. Process. Syst. 2015;28:2224-2232.

[19] Zhang M, Chen Y. Link prediction based on graph neural networks. Advances in neural information processing systems. 2018;31:5171-5181.

[20] Kazemi SM, Poole D. Simple Embedding for Link Prediction in Knowledge Graphs. Adv Neural Inf Process Syst. 2018;31:4289-4300.

[21] Simonyan K, Vedaldi A, Zisserman A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. 2013. ArXiv preprint: https://arxiv.org/pdf/1312.6034.pdf

[22] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning Deep Features for Discriminative Localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016:2921-2929.

[23] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, et al. Grad-Cam: Visual Explanations From Deep Networks via Gradient-Based Localization. In: Proceedings of the IEEE international conference on computer vision; 2017:618-626.

[24] Chattopadhay A, Sarkar A, Howlader P, Balasubramanian VN. Grad-Cam++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In: IEEE winter conference on applications of computer vision (WACV). 2018; 2018: 839-847.

[25] Zintgraf LM, Cohen TS, Adel T, Welling M. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. In International Conference on Learning Representations. 2017.

[26] Fong RC, Vedaldi A. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In: Proceedings of the IEEE international conference on computer vision. 2017:3449-3457.

[27] Fong R, Patrick M, Vedaldi A. Understanding Deep Networks via Extremal Perturbations and Smooth Masks. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019: 2950-2958.

[28] Bach S, Binder A, Montavon G, Klauschen F, Müller KR, et al. On Pixel-Wise Explanations for Non-linear Classifier Decisions by Layer-Wise Relevance Propagation. PLoS One. 2015;10:e0130140.

[29] Shrikumar A, Greenside P, Kundaje A. Learning Important Features Through Propagating Activation Differences. In: International conference on machine learning; 2017: 3145-3153.

[30] https://openreview.net/forum?id=uDN8pRAdsoC

[31] https://openreview.net/forum?id=nzKv5vxZfge

[32] Schnake T, Eberle O, Lederer J, Nakajima S, Schütt KT, et al. Higher-Order Explanations of Graph Neural Networks via Relevant Walks. IEEE Trans Pattern Anal Mach Intell. 2022;44:7581-7596.

[33] Smilkov D, Thorat N, Kim B, Viégas F, Wattenberg M. Smoothgrad: Removing Noise by Adding Noise. 2017. ArXiv preprint: https://arxiv.org/pdf/1706.03825.pdf

[34] Oh Y, Jung H, Park J, Kim MS. Evet: Enhancing Visual Explanations of Deep Neural Networks Using Image Transformations. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision; 2021:3578-3586.

[35] Sanchez-Lengeling B, Wei J, Lee B, Reif E, Wang P, et al. Evaluating Attribution for Graph Neural Networks. Adv Neural Inf Process Syst. 2020;33:5898-5910.

[36] Omeiza D, Speakman S, Cintas C, Weldermariam K. Smooth Grad-Cam++: An Enhanced Inference Level Visualization Technique for Deep Convolutional Neural Network Models. 2019. ArXiv preprint: https://arxiv.org/pdf/1908.01224.pdf

Appendix A. Experimental details

A.1 Model architecture and training

We use basic graph convolutional network (GCN) [2] model for all experiments. Three-layered GCN model with node feature concatenation and fully connected layer is used. Adam optimizer with learning rate 0.001 and L2 weight decay 0.001 is adopted for all datasets, except for Tree-Grid dataset which utilized 0.002 for the weight decay. Each model for BA-shapes, Tree-Cycles, and Tree-Grid datasets is trained for 10000 epochs, whereas models for BA-Community dataset are trained for 5000 epochs. Datasets are taken from GNNExplainer paper [14] under Apache License 2.0. Dataset split is taken from the PGExplainer code [15], which splits train/validation/test sets by 80/10/10%.

Appendix B. Supplementary results

B.1 Explanation accuracy

All explanation-generating experiments are conducted in ten-fold with the ten aforementioned trained models for each dataset. Extended accuracy table with standard deviations and p-values corresponding to the TABLE 1 is shown in TABLE 2. For statistical analysis, each result is first assessed with normality test to check their distribution. For pairs with at least one $p < 0.05$ in normality test, one-sided Wilcoxon signed-rank test is used for analysis. For the rest, one-sided Student's T-test is used.

Table 2: Explanation accuracies (AUC-ROC) before and after applying Seen with standard deviations, p-values, and their best performing α and β values.

Explainer	**BA-Shapes**	**BA-Community**	**Tree-Cycles**	**Tree-Grid**
SA	0.935±0.009	0.625±0.020	0.886±0.050	0.814±0.112
SA + Seen	0.938±0.005	0.637±0.018	0.900±0.035	0.866±0.136
p-value	6.5×10^{-2}	9.4×10^{-3}	1.4×10^{-1}	5.2×10^{-2}
(α, β)	(0.5, 0.5)	(1.0, 0.75)	(1.0, 0.5)	(1.0, 0.75)
Grad*Input	0.925±0.011	0.614±0.015	0.894±0.051	0.722±0.143
Grad*Input + Seen	0.936±0.056	0.718±0.015	0.934±0.033	0.800±0.124
p-value	9.8×10^{-4}	1.5×10^{-2}	9.8×10^{-4}	2.1×10^{-2}
(α, β)	(1.0, 0.5)	(1.0, 0.25)	(1.0, 0.5)	(1.0, 0.5)
GradCAM	0.904±0.055	0.573±0.001	0.750±0.041	0.757±0.029
GradCAM + Seen	0.918±0.000	0.645±0.024	0.756±0.071	0.785±0.060
p-value	1.6×10^{-4}	2.8×10^{-6}	1.4×10^{-1}	5.2×10^{-2}
(α, β)	(0.25, 0.25)	(1.0, 0.25)	(0.25, 0.5)	(0.25, 0.5)

B.2 Aggregation coefficients

Performance scan over aggregation coefficients is conducted within $[0, 1]$ for α and $[0, 1)$ for β with 0.25-increament. Note that performances for varying β with $\alpha = 0$ are identical, depicted as the right bottom line of each plot in FIGURE 3.

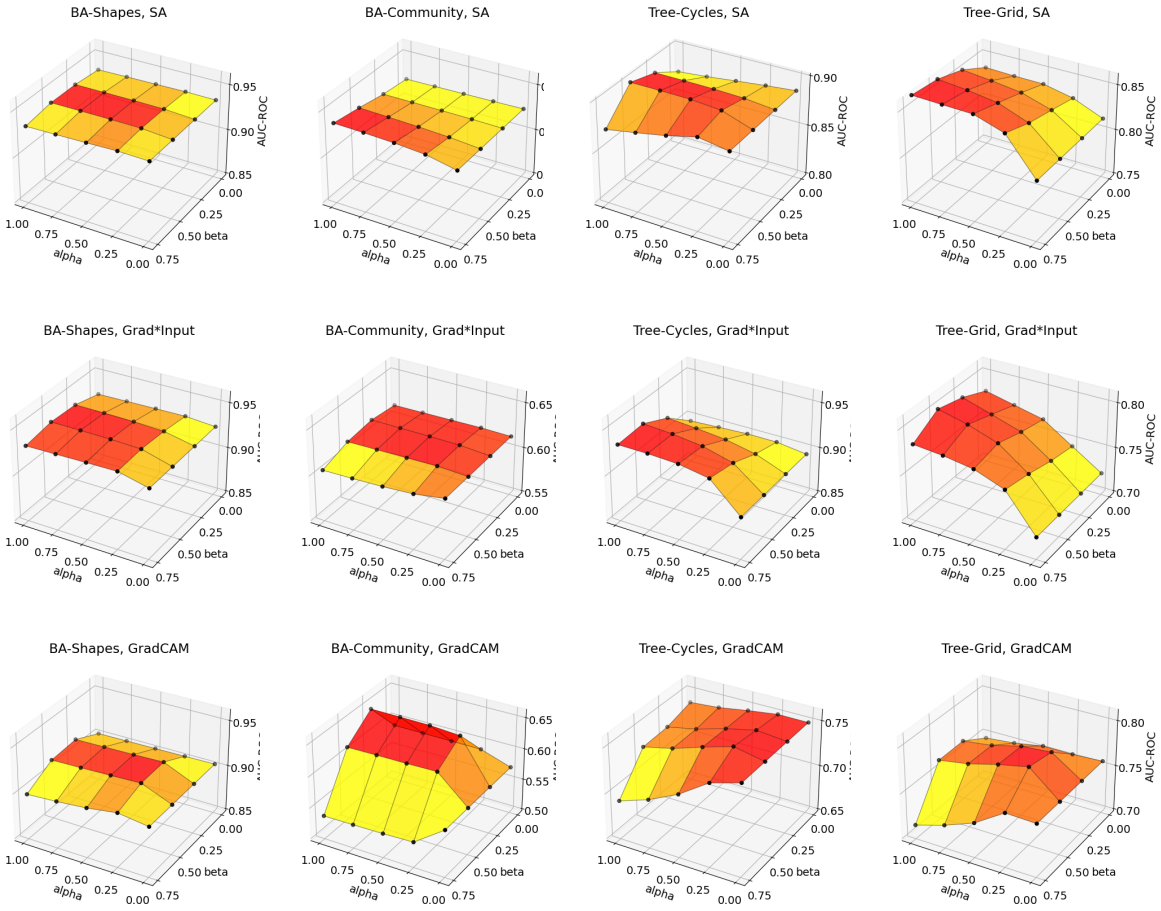


Figure 3: Effects of aggregation coefficients on explanation accuracy for BA-Shapes, BA-Community, Tree-Cycles, and Tree-Grid datasets.