

Combining Stochastic Explainers and Subgraph Neural Networks can Increase Expressivity and Interpretability

Indro Spinelli ^{1a} and Michele Guerra ² and
Filippo Maria Bianchi ² and Simone Scardapane ^{1b *}

1 - Sapienza University of Rome
a - DI Dept., Via Salaria 113, 00198 Rome - Italy
b - DIET Dept., Via Eudossiana 18, 00184 Rome - Italy

2 - UiT the Arctic University of Norway - Dept. of Mathematics and Statistics
Hansine Hansens veg 18, 9019 Tromsø - Norway

Abstract. Subgraph-enhanced graph neural networks (SGNN) can increase the expressive power of the standard message-passing framework. This model family represents each graph as a collection of subgraphs, generally extracted by random sampling or with hand-crafted heuristics. Our key observation is that by selecting “meaningful” subgraphs, besides improving the expressivity of a GNN, it is also possible to obtain interpretable results. For this purpose, we introduce a novel framework that jointly predicts the class of the graph and a set of explanatory sparse subgraphs, which can be analyzed to understand the decision process of the classifier. The subgraphs produced by our framework allow to achieve comparable performance in terms of accuracy, with the additional benefit of providing explanations.

1 Introduction

Graph neural networks (GNNs) are neural network models designed to adapt and perform inference on graph domains [1] providing state-of-the-art results in a comprehensive set of scenarios. Nevertheless, regulations require high task performance and a transparent decision process. For this reason, several researchers have investigated techniques to explain GNNs’ predictions inferring the reasons that led to a specific outcome by a trained model [2, 3]. However, recent efforts toward “explainable-by-design” GNNs rather than post-hoc explainers are opening up new, interesting approaches [4, 5]. On a separate line of research, recent studies [6] demonstrated that by providing the GNN with subgraphs that give different views of the same graph, it is possible to increase the expressive power of the standard message-passing framework. We propose to connect these two topics and build an explainable by-design subgraph-enhanced GNN. We use a data-driven approach to learn small and representative subgraphs that increase the expressive power for the downstream task and that can be used as explanations.

*This work was partly funded by Sapienza grant RM1221816BD028D6 DESMOS. I. Spinelli has been fully supported by PNR MUR project PE0000013-FAIR.

2 Related Works

2.1 Explainability in GNNs

Explainability techniques enable humans to interpret the decision process of GNNs, discover potential sources of error, find biases and limitations in the model and learn more about the data and the task at hand. Most approaches in the literature are post-hoc explainers differentiated according to the techniques used to explain a trained GNN. In particular, there are gradient-based approaches [7], perturbation-based approaches [2, 3] and decomposition methods [8]. Meanwhile fewer research efforts have been devoted to explainable-by-design GNNs [4, 5]. Of particular interest for this work is **PGExplainer** [3], which uses a small network to parametrize the probability of each edge ω_{ij} of being part of the explanatory subgraph, and samples from this distribution to obtain the final explanation subgraph characterized by edges e_{ij} . The optimization objective maximizes the mutual information between the masked prediction and the prediction obtained from the original graph. In addition, an element-wise entropy term encourages sparsity on ω , and an $l1$ -norm forces small explanation subgraphs.

2.2 Subgraph-enhanced graph neural networks

The study of the expressive power of GNNs has always been of central interest to the community. It has recently been shown that it is possible to create more expressive GNNs using standard architectures that process several subgraphs of the input graph [6]. In particular, **ESAN** [6] represents each graph as a bag of subgraphs $\{\mathcal{G}_1, \dots, \mathcal{G}_m\}$ chosen according to some predefined policy, e.g., all graphs obtainable by removing one edge (edge deleted strategy) or one node (node deleted strategy) from the original graph. The encoder implements a module L_1 consisting of several message-passing layer that process each subgraph independently, and then a second message-passing module L_2 preprocesses the aggregation of the subgraphs working as an information-sharing module for the subgraphs. For example, to compute the embedding H_i for the subgraph \mathcal{G}_i , they use the following procedure:

$$H_i = L_1(\mathcal{G}_i) + L_2\left(\sum_j \mathcal{G}_j\right). \quad (1)$$

Then, in the last layer of the encoder, a pooling operation aggregates the node embeddings into subgraph embeddings with a global pooling operation. Finally, a set learning module [9] aggregates the obtained subgraph representations into a single one used in downstream tasks. The authors of [6] name this general configuration DSS-GNN.

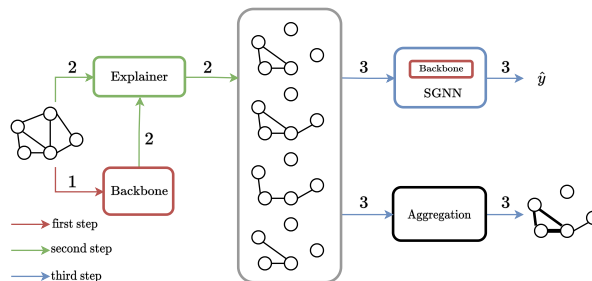


Fig. 1: In the **first step**, we train the backbone of the SGNN, a GIN classifier, using the original graphs. In the **second step**, we train the explainer with the original graph and the backbone’s predictions and use it to create the new representation consisting of bags of explanation subgraphs. Finally, in the **third step**, we train the whole SGNN framework, fine-tuning the backbone using the explanatory subgraphs. Our model outputs the predicted label and an explanation obtained by combining all the subgraphs used during training.

3 Proposed Framework

In this work, we consider an undirected and unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ is the set of node indexes, and $\mathcal{E} \subseteq \{(i, j) \mid i, j \in \mathcal{V}\}$ is the set of edges connecting pairs of nodes. The entities and relationships represented by nodes and edges depend on the data and the application. The graph topology can be represented by the adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$. Other operators matching the sparsity pattern of \mathbf{A} , such as the graph Laplacian, can be used to define the weighted connectivity of the graph. Letting \mathbf{D} be one of these operators encoding the topological information, a GNN layer is defined by (ignoring biases for simplicity, but w.l.o.g.):

$$\mathbf{H} = \phi(\mathbf{D}\mathbf{X}\mathbf{W}), \quad (2)$$

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is a matrix collecting all vertex features row-wise, $\mathbf{W} \in \mathbb{R}^{d \times q}$ is a matrix of trainable coefficients, and ϕ is an element-wise non-linear function (e.g., for ReLU $\phi(s) = \max(0, s)$). A GNN can stack multiple layers in the form of (2) to learn highly nonlinear node representations that account for larger neighbourhoods on the graph. In this work, we are interested in graph classification tasks. This task requires a global pooling operation, such as mean, max, or min pooling, to build a graph representation from the node representations produced by stacking the layers described in Equation (2).

Our goal is to develop a framework that jointly predicts the graph class and the explanation masks, highlighting the parts of the graph that contribute the most to the prediction. The key to our framework is the role played by the subgraphs which are learned end-to-end, based on the optimization of the classification loss. Firstly, they must improve the expressive power of the base GNN classifier. Secondly, they must serve as explanation masks by selecting the

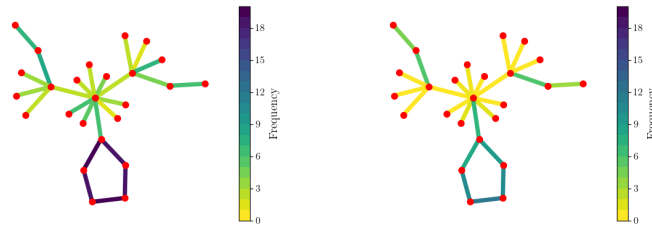


Fig. 2: The collection of subgraphs is represented as a heatmap. On the left, there are the subgraphs obtained with the noise based strategy. On the right, those obtained with the progressive top-K selection.

parts of the input that mostly contribute to determine the correct class. We expect such subgraphs to be more informative than post-hoc explanation masks since they are directly generated by the model to maximize the classification performance. We summarize our framework, which consists of three main steps highlighted in Fig. 1. In the **first step**, we train the backbone of the SGNN, which is a classifier that processes the original graph. In the **second step** we apply PGExplainer with a minor modification to circumvent the “introduced evidence” [10] issue due to the presence of soft masks. Specifically, we binarize the soft weights to create “hard” explanation masks. These modifications lead to design choices for the regularization terms that are different from those originally used in PGExplainer. In particular, we apply the regularization directly on the “hard” explanation mask e instead of ω . Therefore we remove the element-wise entropy term and the l_1 -norm, in this case, is equivalent to an l_0 -norm minimizing the number of edges appearing in the explanation. Furthermore, we devise two strategies to obtain a heterogeneous yet meaningful set of subgraphs. The first is to use the reparametrization trick also in the inference phase allowing the noise, which can be tuned as a hyperparameter, to inject diversity that leads to multiple different subgraphs. In the second approach we introduce a budget on the explanation. We adapt the binarization threshold to obtain a set of explanations with K edges, where K ranges from 5% to 75% of the initial number of edges in the original graph. An example of these two strategies is presented in Figure 2, on the synthetic dataset BA-2Motifs where the five nodes cycle is responsible for the prediction. In the **third step**, we insert the backbone GNN into the DSS-GNN model introduced in ESAN in [6]. The backbone works as the encoder that processes each explanation subgraph independently. A new message-passing module preprocesses the aggregation of the explanations working as an information-sharing module for the subgraphs. Finally, a new set learning module [9] aggregates subgraph representations obtained after a global pooling operation into a single one used in downstream tasks. Besides being used by the information-sharing module, the subgraph obtained by aggregating all the explanations is used to explain the model’s prediction.

Model	MUTAG	NCI1	PROTEINS	IMDB-BINARY	IMDB-MULTI
GIN	89.4 ± 5.6	82.7 ± 1.7	76.2 ± 2.8	75.1 ± 5.1	52.3 ± 2.8
DSS ND 1.0	89.0 ± 4.4	83.0 ± 2.0	76.5 ± 3.7	75.4 ± 3.5	52.3 ± 2.9
DSS ND 0.1	89.9 ± 2.8	83.1 ± 2.0	76.6 ± 4.7	75.7 ± 2.3	52.5 ± 2.2
DSS ED 1.0	89.9 ± 4.3	82.7 ± 1.7	74.0 ± 4.4	76.7 ± 3.6	52.6 ± 2.6
DSS ED 0.1	89.9 ± 5.5	82.1 ± 2.0	75.1 ± 4.7	76.4 ± 2.7	52.6 ± 2.8
OURS	92.1 ± 4.3	82.9 ± 2.3	76.3 ± 4.4	76.0 ± 3.1	53.5 ± 3.8

Table 1: TUDataset classification accuracy. We report the mean and the standard deviation obtained by running a 10-fold cross-validation.

4 Experimental Evaluation

We want to prove that our framework retains the same classification performance of known SGNN architectures. However, our framework has the added benefit of re-using the subgraphs involved in the computations as plausible explanations for the model’s prediction. We used the same evaluation proposed in ESAN [6]. We selected GIN [11] and DSS-GNN with the GIN backbone as baselines. We used the most compact model consisting of 4 GIN layers with two linear layers of a hidden size of 32. The batch size is also 32. We selected the top performing policies from [6]: node deleted (ND) and edge deleted (ED). Furthermore, we consider both their deterministic versions: one that uses the entire bag of subgraphs (referred to as “1.0”) and the sampled versions, which keeps only 10% of the bag (referred to as “0.1”). We report the results in Table 1. We can observe that subgraph-enhanced GNNs achieve slightly better accuracies and that the results obtained using explanatory subgraphs instead of randomly generated ones are comparable. However, the former provides an explanation helping human experts to extract new knowledge from the model.

5 Conclusion

We introduced a framework to bridge subgraph-enhanced GNNs with explainability techniques. First, we pre-train a base GNN such that we can train a post-hoc explainer modified to satisfy the need for multiple and discrete explanation masks. Next, the explainer generates a bag of explanation subgraphs. Then, we resume the training of the base model, including equivariant layers, to process the new representation in the SGNN fashion. Finally, we combine the subgraphs in a unique heatmap highlighting the most relevant parts responsible for the prediction and the prediction itself. This additional information allows human experts to interpret and extract knowledge from the model. The experimental evaluation showed that our model achieves comparable classification performance to state-of-the-art models while providing, at the same time, an explanation.

References

- [1] Davide Bacciu, Federico Errica, Alessio Micheli, and Marco Podda. A gentle introduction to deep learning for graphs. *Neural Networks*, 129:203–221, 2020.
- [2] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in Neural Information Processing Systems*, 32:9240, 2019.
- [3] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in Neural Information Processing Systems*, 33, 2020.
- [4] Indro Spinelli, Simone Scardapane, and Aurelio Uncini. A meta-learning approach for training explainable graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–9, 2022.
- [5] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. Protgnn: Towards self-explaining graph neural networks. In *Thirty-Sixth AAAI Conference on Artificial Intelligence*, pages 9127–9135. AAAI Press, 2022.
- [6] Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M. Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. In *The Tenth International Conference on Learning Representations, ICLR*. OpenReview.net, 2022.
- [7] Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10764–10773, 2019.
- [8] Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T Schutt, Klaus-Robert Mueller, and Gregoire Montavon. Higher-order explanations of graph neural networks via relevant walks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.
- [9] Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements. In *Proceedings of the 37th International Conference on Machine Learning, ICML’20*. JMLR.org, 2020.
- [10] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [11] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.