HARD MASKING FOR EXPLAINING GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Neural Networks (GNNs) are a flexible and powerful family of models that build nodes' representations on irregular graph-structured data. This paper focuses on explaining or interpreting the rationale underlying a given prediction of already trained graph neural networks for the node classification task. Existing approaches for interpreting GNNs try to find subsets of important features and nodes by learning a continuous mask. We alternately formulate interpretability inspired from data compression, where we consider an explanation as a compressed form of the original input representation. Our objective is to find discrete masks that are arguably more interpretable while minimizing the expected deviation from the underlying model's prediction. We empirically show that our explanations are both more predictive and sparse. Additionally, we find that multiple diverse explanations are possible, which sufficiently explain a prediction. Finally, we analyze the explanations to find the effect of network homophily on the decision-making process of GNNs.

1 INTRODUCTION

Graph Neural Networks (GNNs) are a flexible and powerful family of models that build representations of nodes or edges on irregular graph-structured data and have experienced significant attention in recent years. These methods are based on the so-called "neighborhood aggregation" scheme in which a node representation is learned by aggregation of features from their neighbors and have achieved state-of-the-art performance on node and graph classification tasks. Despite their popularity, approaches investigating their interpretability have received limited attention. This paper focuses on explaining or interpreting the rationale underlying a given prediction of already trained graph neural networks.

There have been numerous approaches proposed in the literature for the general interpretability of machine learning models. The most popular approaches are feature attribution methods that intend to attribute importance to input features given an input prediction either agnostic to the model parameter (Ribeiro et al., 2018; 2016) or using model-specific attribution approaches (Xu et al., 2015; Binder et al., 2016; Sundararajan et al., 2017). However, models learned over graph-structured data have some unique challenges. Specifically, predictions on graphs are induced by a complex combination of nodes and paths of edges between them in addition to the node features. Thus explanations for a prediction should ideally be a small subgraph of the input graph and a small subset of node features that are most influential for the prediction (Ying et al., 2019).

The only existing approach for GNN explainability proposes to learn a real-valued graph mask that selects the important subgraph of the GNNs computation graph to maximize the mutual information with the GNNs prediction (Ying et al., 2019). We identify two crucial limitations of such an approach. Firstly, although mathematically tractable, a continuous mask does not ensure sparsity compared to a discrete mask – a desirable property for interpretability. Secondly, suitable notions of what constitutes an explanation in a GNN model and its evaluation are missing.

This paper proposes an alternate notion of interpretability for GNNs grounded in ideas from data compression in information theory. Specifically, we consider an explanation as a compressed form of the original feature matrix. The goodness of the explanation is measured by the expected deviation from the prediction of the underlying model. We formalize this idea of interpreting GNN decisions as an explicit optimization problem in a *rate-distortion framework*. A subgraph of the

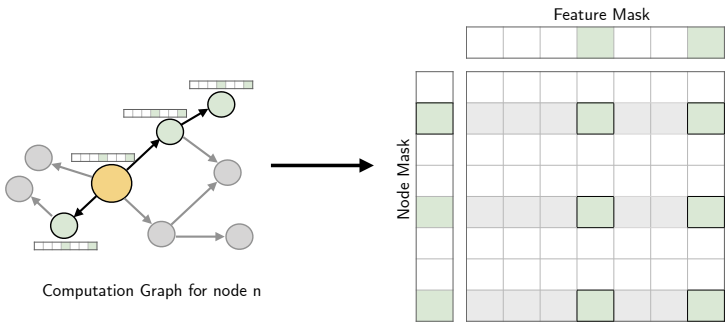

Figure 1: Computing Hard Masks for Explaining the prediction of GNN.

node's computational graph and its set of features are relevant for a classification decision if the expected classifier score remains nearly the same when randomizing the remaining features. This formulation is arguably both a crisp, robust, and understandable notion of interpretability that is easy to evaluate. We propose a simple combinatorial procedure ZORRO that aims to find a sparse subset of features and nodes in the computational graph while adhering to a user-specified level of fidelity. Our method aims to find multiple disjoint explanations (whenever possible) that guarantee an acceptable lower bound on *fidelity* or *faithfulness to the model's decision*.

Another key problem in post-hoc interpretability of GNNs is that of evaluating explanation methods. Current evaluation methods, such as those used by GNNEXPLAINER, are largely anecdotal and lack principled metrics. Secondly, there exists no ground truth for the explanation which makes comparison difficult. Other works in the general area of interpretability deem an explanation to be relevant if it agrees with human understanding. We on the other hand are not interested if an explanation is congruent to human understanding but rather faithful to the actual model decision.To this extent our optimization metric, fidelity, encodes an information-theoretic interpretation of explanation – if the explanation is highly predictive in expectation then it is a faithful explanation.

We conducted extensive experimentation on three datasets and four diverse GNN approaches – Graph Convolution Networks (Kipf & Welling, 2017), Graph Attention Networks (Veličković et al., 2018), GIN (Xu et al., 2019), and APPNP (Klicpera et al., 2019). Our main key findings are as follows.

1. We show that not one but multiple diverse explanations are possible that sufficiently explain a prediction. This multiplicity of explanations indicates the possible configurations that could be utilized by the model to arrive at a decision.
2. Unlike earlier mutual-information preserving interpretability approaches, i.e. GNNEXPLAINER (Ying et al., 2019), we show that our explanations are both more predictive and sparse. We show that even with sparser explanations, our approach contains far more predictive capacity than GNNEXPLAINER.
3. We then analyze the explanations across multiple GNN models to showcase differences between their learning behavior. We specifically show that GNN models rely heavily on homophily and that prediction errors are due inability to capture homophilic signals from their neighborhoods.

2 RELATED WORK

Representation learning approaches on graphs encode graph structure with or without node features into low-dimensional vector representations, using deep learning and nonlinear dimensionality reduction techniques. These representations are trained in an unsupervised (Perozzi et al., 2014; Khosla et al., 2019; Funke et al., 2020) or semi-supervised manner by using neighborhood aggregation strategies and task-based objectives (Kipf & Welling, 2017; Veličković et al., 2018).

This work focuses on the post-hoc interpretability of decisions made by semi-supervised models based on graph convolution networks for node classification tasks. Inspired by the success of con-

volutional neural networks, graph convolution network (GCN)(Kipf & Welling, 2017) generalizes the convolution operation for irregular graph data. GCN and several of its variants follow a neighborhood aggregation strategy where they compute a node's representation by recursive aggregation and transformation of feature representations of its neighbors. For the node classification task, the final node representations are then used to predict unlabelled nodes' classes.

Interpretability in Machine Learning. Post-hoc approaches to model interpretability are popularized by *feature attribution* methods that aim to assign importance to input features given a prediction either agnostic to the model parameters (Ribeiro et al., 2018; 2016) or using model specific attribution approaches (Xu et al., 2015; Binder et al., 2016; Sundararajan et al., 2017). *Instance-wise feature selection* (IFS) approaches (Chen et al., 2018; Carter et al., 2018; Yoon et al., 2018), on the other hand, focuses on finding a *sufficient* feature subset or explanation that leads to little or no degradation of the prediction accuracy when other features are masked. The advantage of this formulation is that the output explanation has a precise meaning in terms of the predictive power of the chosen subset. Applying these works directly for graph models is infeasible due to the complex form of explanation, which should consider the complex association among nodes in addition to the input features.

Interpretability in GNNs. Model agnostic approaches like ours to interpretability in GNNs include GNNEXPLAINER (Ying et al., 2019) and XGNN(Yuan et al., 2020). GNNEXPLAINER learns a real-valued graph mask and feature mask such that the mutual information with GNN's predictions is maximized. XGNN proposed a reinforcement learning-based graph generation approach to generate explanations for the predicted class for a graph. We instead focus on explaining node level decisions. As a model introspective approach, Pope et al. (2019) extended the gradient-based saliency map methods to GCNs, which rely on propagating gradients/relevance from the output to the original model's input features. Other works (Kang et al., 2019; Idahl et al., 2019) focus on explaining unsupervised network representations, which is out of scope for the current work.

3 PROBLEM DEFINITION AND APPROACH

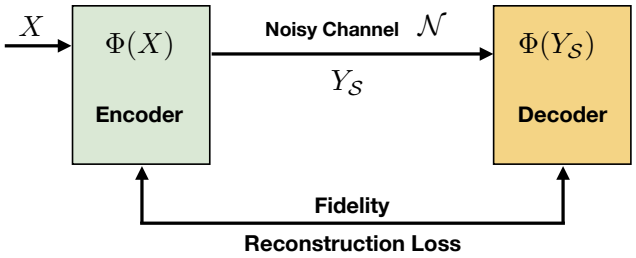

Figure 2: Our approach is based on the rate-distortion framework, which assumes two parties, the encoder and decoder. The encoder has the original (restricted) feature matrix X and sends it through a noisy channel to the decoder, which receives $Y_{\mathcal{S}}$. The goal is to determine, which parts of $Y_{\mathcal{S}}$ have to have the original values from X, such that the fidelity is high, i.e., the decoder predicts the same label as the encoder with a high probability.

Notations. Let $G = (V, E)$ be a graph, $\mathcal{X}$ the corresponding matrix of node features, and Φ a graph neural network, i.e. $\Phi(\mathcal{X})$ are the predicted labels for all nodes in V. In order to predict the label of a single node, the graph neural network Φ usual is only affected by a subset of nodes and entries of the feature matrix $\mathcal{X}$. Let $G(n) \subseteq G$ be the subgraph of these nodes, which we call the *computational graph* of the node $n \in V$ for the graph neural network Φ. The size of the computational graph depends on the number of layers of the GNN. For example, for a 2-layer network, all nodes and edges in the node's two-hop neighborhood would constitute its computational graph. $X(n)$ or briefly X denotes the feature matrix restricted to the nodes of $G(n)$, where each row corresponds to a feature vector of a node in the computational graph.

We formulate the task of explaining the model prediction for a node n, as finding a partition of the components of its computational graph into a subset, $\mathcal{S}$ of relevant nodes and features, and its

complement $\mathcal{S}^c$ of non-relevant components. In particular, the subset $\mathcal{S}$ should be such that fixing its value to the true values already determines the model output for almost all possible assignments to the non-relevant subset $\mathcal{S}^c$. The subset $\mathcal{S}$ is then returned as an explanation. To quantify *relevance*, we compute the expected value of *fidelity* in model's prediction for the noisy assignment to the non-relevant components.

Let us denote with $Y_\mathcal{S}$ the new matrix with fixing components of the $\mathcal{S}$ to their actual values and otherwise noisy entries. The values of components in $\mathcal{S}^c$ are then drawn from some noisy distribution, $\mathcal{N}$. Let $\mathcal{S} = \{V_s, F_s\}$ be the explanation with selected nodes V_s and selected features F_s. V_s denotes the binary column vector of selected nodes and F_s denote the binary row vector of selected nodes. Let S be the mask matrix such that each element $S_{i,j} = 1$ if and only if ith node (in $G(n)$) and jth feature are included in sets V_s and F_s respectively and 0 otherwise.

$$Y_\mathcal{S} = X \odot S + Z \odot (\mathbb{1} - S), Z \sim \mathcal{N}, \tag{1}$$

where $\odot$ denotes an element-wise multiplication, and $\mathbb{1}$ a matrix of ones with the corresponding size. Figure 1 shows how the fixed elements are selected by F_s and V_s.

Definition. *The fidelity of explanation $\mathcal{S}$ with respect to the graph neural network Φ and the noise distribution $\mathcal{N}$ is given by*

$$\mathcal{F}(\mathcal{S}) = \mathbb{E}_{Y_\mathcal{S}|Z\sim\mathcal{N}}\left[\mathbb{1}_{\Phi(X)=\Phi(Y_\mathcal{S})}\right]. \tag{2}$$

By fixing the fidelity to a certain user-defined threshold, say τ, we are then interested in all possible disjoint sets of explanations that would have the fidelity of at least τ. More precisely, our resulting set of explanations R is given as

$$R = \left\{\mathcal{S}_1, \mathcal{S}_2, \dots \;\middle|\; \forall_i \mathcal{F}(\mathcal{S}_i) \geq \tau \text{ and } \bigcap_i \mathcal{S}_i = \emptyset\right\} \tag{3}$$

Connection to the Rate-Distortion Theory

Our problem formulation is inspired by *rate-distortion theory* (Sims, 2016) which addresses the problem of determining the minimal information of a source signal, that should be communicated over a leaky channel, so that the source (input signal) can be approximately reconstructed at the receiver (output signal) without exceeding an expected distortion $\mathcal{D}$. In our problem, we are interested in finding a small subset $\mathcal{S}$ such that having knowledge only about the signal on $\mathcal{S}$ and filling in the rest of the information randomly will almost surely preserve the class prediction if our chosen subset contains the information that is relevant for the model's decision. Rather than measuring distortion or disagreement in the model's decisions, we instead measure fidelity or agreement among the model's decisions with the original and the distorted signal, respectively. Distortion can be computed using fidelity as $\mathcal{D} = 1 - \mathcal{F}$. A schematic representation of our problem in terms of rate-distortion framework is shown in Figure 2.

3.1 Our Approach: Zorro

We propose a simple but effective greedy combinatorial approach, which we call Zorro, to find the set of disjoint explanations with a desired level of fidelity. The pseudocode is provided in Algorithm 1. Let for any node n, $G(n)$ denote its computational graph with a set of vertices denoted by V_n. Let F denote the complete set of features. We start with zero-sized explanations and select as first element

$$\underset{f\in F_p}{\operatorname{argmax}}\ \mathcal{F}(V_p, f) \quad \text{or} \quad \underset{v\in V_p}{\operatorname{argmax}}\ \mathcal{F}(v, F_p),$$

whichever yields the highest fidelity value. In addition, we save for each possible node $v \in V_p$ and feature $f \in F_p$ the ordering R_V and R_F given by the fidelity values $\mathcal{F}(v, F_p)$ and $\mathcal{F}(V_p, f)$ respectively. We iteratively add new features or nodes to the explanation such that the fidelity is maximized over all evaluated choices. To reduce the computational cost, we only evaluate in each iteration the top K remaining nodes and features determined by R_{V_p} and R_{F_p}.

Once we found an explanation with the desired fidelity, we discard the chosen elements from the feature matrix X, i.e., we never consider them again as possible choices in computing the next explanation. We repeat the process by finding relevant selections completely disjoint from all already

found. To ensure that disjoint elements of the feature matrix X are selected, we recursively call Algorithm 3 with either remaining (not yet selected in any explanation) set of nodes or features. Finally, we return the set of explanations such that the fidelity of τ cannot be reached by using all the remaining components which are not in any of the explanations.

Algorithm 1 ZORRO(n, τ, K)

$V_n \leftarrow$ set of vertices in $G(n)$
$F \leftarrow$ set of node features
return GetExplanations(τ, K, V_n, F)

Algorithm 2 $\mathcal{F}(V_s, F_s)$

for $i = 0, \ldots,$ samples **do**
  Set $Y_{\{V_s, F_s\}}$, i.e. fix the selected values and otherwise retrieve random values from the respective columns of $\mathcal{X}$
  if $\Phi(Y_{\{V_s, F_s\}})$ matches the original prediction of the model **then**
    correct$+ = 1$
return $\frac{\text{correct}}{\text{samples}}$

Algorithm 3 GetExplanations(τ, K, V_p, F_p)

$\mathcal{S} = \emptyset$, $V_r = V_p$, $F_r = F_p$
Initialization of V_s, F_s, R_{V_p}, and R_{F_p}
while $\mathcal{F}(V_s, F_s) \geq \tau$ **do**
  $\tilde{V}_s = V_s \cup \underset{v \in \text{top}_K(V_r, R_{V_p})}{\text{argmax}} \mathcal{F}(v \cup V_s, F_s)$
  $\tilde{F}_s = F_s \cup \underset{f \in \text{top}_K(F_r, R_{F_p})}{\text{argmax}} \mathcal{F}(V_s, f \cup F_s)$
  if $\mathcal{F}(\tilde{V}_s, F_s) \leq \mathcal{F}(V_s, \tilde{F}_s)$ **then**
    $F_r = F_r \setminus f$, $F_s = \tilde{F}_s$
  else
    $V_r = V_r \setminus v$, $V_s = \tilde{V}_s$
$\mathcal{S} = \mathcal{S} \cup \{V_s, F_s\}$
$\mathcal{S} = \mathcal{S} \cup$ GetExplanations(τ, K, V_p, F_r)
$\mathcal{S} = \mathcal{S} \cup$ GetExplanations(τ, K, V_r, F_p)
return $\mathcal{S}$

The pseudocode to compute fidelity is provided in Algorithm 2. Specifically we generate the obfuscated instance for a given explanation $\mathcal{S} = \{V_s, F_s\}$, $Y_{\mathcal{S}}$ by setting the feature values for selected node-set V_s corresponding to selected features in F_s to their true values. Figure 1 visualizes how the fixed elements of the feature matrix are determined by choice of node mask V_s and feature mask F_s. To set the irrelevant values, we randomly choose a value from the set of all possible values for that particular feature in the dataset $\mathcal{X}$. To approximate the expected value in Equation 2, we generate a finite number of samples of $Y_{\mathcal{S}}$. We then compute fidelity as the fraction of samples for which the model's decision matches its decision for the original instance. Our implementation of the proposed algorithm will be made public after publication.

CHOICE OF NOISY DISTRIBUTION $\mathcal{N}$

One might argue that the irrelevant components can be set to 0 rather than any specific noisy value. However, this might lead to several side effects: in case of datasets for which a feature value of 0 is not allowed or have some specific semantics or for models with some specific pooling strategy, for example, minpool. More specifically, the idea of an irrelevant component is not that it is missing, but its value does not matter. Therefore to account for the irrelevancy of certain components given our explanation, we need to check for multiple noisy instantiations for the unselected components. Our choice of using the global distribution of features as the noisy distribution ensures that only plausible feature values are used. Next, our choice does not increase the bias towards specific values, which we would have by taking fixed values such as 0 or averages.

4 EXPERIMENTS

In our experiments, we tried to answer three primary research questions:

RQ 1. *How often do multiple explanations exist for a given prediction? What are the sparsity-fidelity trade-offs for our approach?*

RQ 2. *How effective is* ZORRO *as compared to existing approaches in terms of fidelity and sparsity?*

RQ 3. *Can we discover differences in model behavior by post-hoc analysis of explanations?*

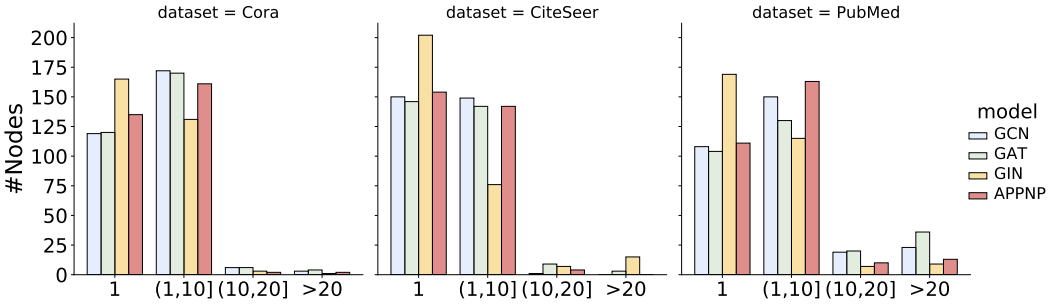

Figure 3: The number of explanations found with ZORRO at $\tau = 0.85$.

Evaluation Strategy. To answer these research question, we use the datasets **Cora**, **CiteSeer** and **PubMed** from Yang et al. (2016). We evaluate our approach on four different two-layer graph neural networks: graph convolutional network (**GCN**) (Kipf & Welling, 2017), graph attention network (**GAT**) (Veličković et al., 2018), the approximation of personalized propagation of neural predictions (**APPNP**) (Klicpera et al., 2019), and graph isomorphism network (**GIN**) (Xu et al., 2018). Table 1 shows the statistics of the datasets and models. For more details of our experimental setup, we refer to Appendix A.1.

4.1 MULTIPLICITY AND SIZE OF EXPLANATIONS

Our first result is that multiple (disjoint) explanations are indeed possible and are frequent. Figure 3 shows the number of nodes having multiple explanations. We observe that, without exception, all GNN models yield multiple disjoint explanations with $\approx 50\%$ of the 300 nodes under study have 2 to 10 explanations. The disjoint explanations produced by our algorithm can be understood as a disjoint piece of evidence that would lead the model to the same decision. We expect a much larger number of overlapping explanations if the restrictive condition on disjointness is relaxed. However, the objective here is to show that a decision can be reached in multiple ways, and each explanation is a practical realization of a possible combination of nodes and features that constitutes a decision. We are the first to establish the multiplicity of explanations for model predictions, unlike Ying et al. (2019) that outputs only one explanation as a soft mask over features and edges.

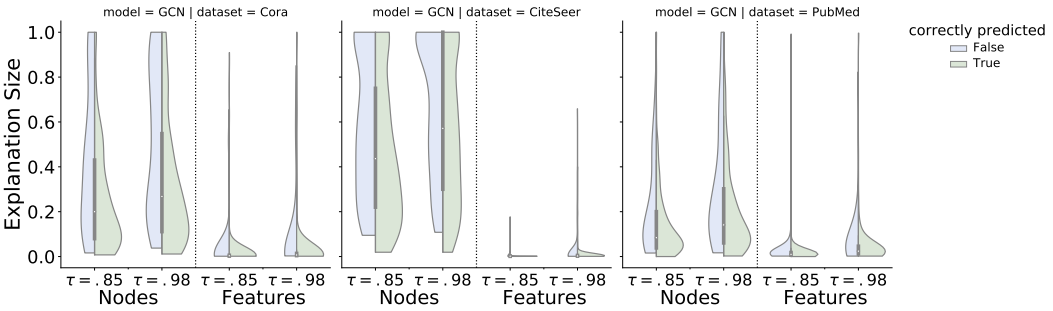

Figure 4: Comparison of explanation size of explanations measured by the proportion of selected elements, i.e., the number of selected nodes is divided by the number of nodes in the computational graph, and the number of selected features is divided by the number of features. For the results of all combinations, see Figure 8.

In general, shorter or sparser explanations are more human interpretable and hence more desirable. We conducted experiments using two fidelity thresholds ($\tau = 0.85$ and $\tau = 0.98$) and compared the sizes of the first found explanation (see Figure 4). A key observation in this result is that ZORRO is still able to find sparse explanations for high fidelity requirements, i.e., at $\tau = 0.98$, suggesting that only a small number of nodes and features are required for most of the predictions. Although there are differences between datasets in terms of the size of the explanation, the choice of τ has limited influence. Finally, we interestingly find that mispredictions tend to require a larger explanation size

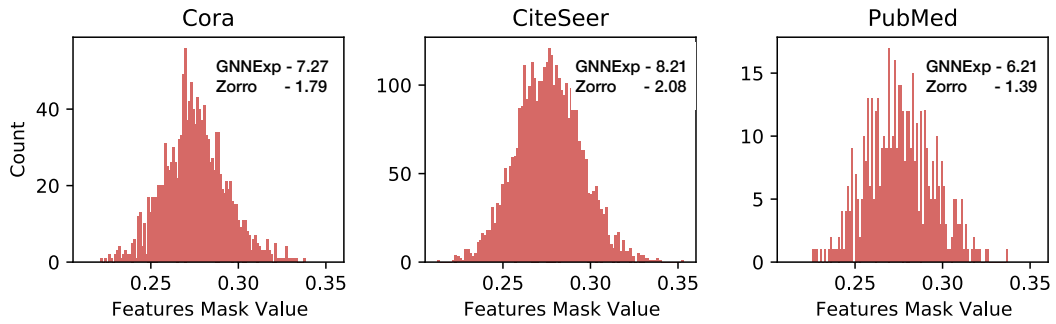

Figure 5: Distribution of the mask values over the node features corresponding to explanations of a single node of GCN, where GNNEXPLAINER achieves a fidelity of 1.0. The values in the top right corner correspond to the entropy value of the retrieved feature mask for GNNEXPLAINER and ZORRO with $\tau = .98$. For the distribution of all nodes, models, and datasets, we refer to Figure 11.

where explanation size is the fraction of nodes/features selected by ZORRO in comparison to the entire computation graph for the node.

4.2 COMPARISON WITH GNNEXPLAINER

We first note that unlike ZORRO an explanation from GNNEXPLAINER is a soft mask of importance scores $\in [0, 1]$ for node features and edges of the computational graph. In principle, one can choose the top-k features of GNNEXPLAINER with the highest importance values for comparing both the fidelity of both approaches. However, we choose the best settings of GNNEXPLAINER, i.e., the soft mask, to compare the fidelity distributions of both approaches (see Figure 6). As expected, the fidelity of explanations by ZORRO for all nodes is above is high –$[0.85, 0.95]$ – or very high –$[0.85, 0.95]$. However, GNNEXPLAINER exhibits low fidelity, i.e.,≤ 0.70 for a fairly large $\approx 40\%$ of the nodes. The results show that our explanations are much more faithful to the model than the soft explanations of GNNEXPLAINER.

But are the explanations from ZORRO high fidelity because they are less sparse than GNNEXPLAINER? To systematically measure this, we computed the entropy of explanations output by both approaches as a measure of sparsity (see Figure 5). We see that ZORRO has a very low entropy, suggesting that the hard mask has very few selected features.

We also visualize the output of the soft explanation of a node that is completely faithful to the model i.e., the explanation achieves the fidelity value of 1. We observe that the mask values are distributed around a given value explaining the low entropy. In such cases, when all masks for all features take low values, a small sized explanation cannot be obtained as all components have similar importance. Effectively the complete input is presented as an input, and no wonder that the highest value of fidelity is achieved. From the above two experiments, we conclude that ZORRO produces both sparse and high-fidelity explanations in comparison to GNNEXPLAINER.

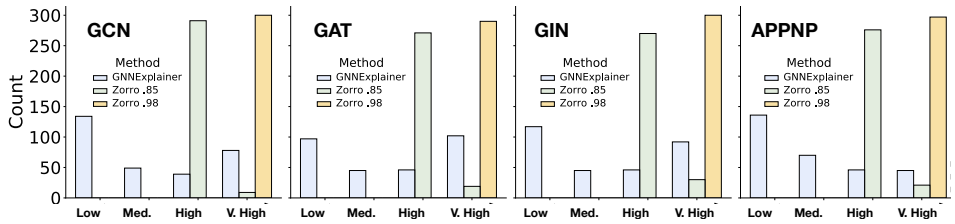

Figure 6: Comparison of fidelity, GNNEXPLAINER vs. ZORRO . **Low**: < 0.7, **Med.**: $[0.7, 0.85)$, **High**: $[0.85, 0.95)$ and **V. High**: ≥ 0.95

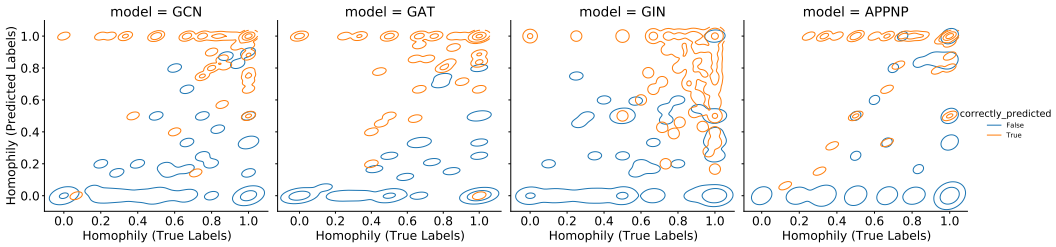

Figure 7: Dataset - PubMed. The joint distribution of the homophily with respect to the selected subgraph (in the explanation of ZORRO with $\tau = 0.85$) with true and predicted labels. The orange contour lines correspond to the distributions for correctly predicted nodes, and the blue one corresponds to incorrectly predicted nodes.

4.3 HOMOPHILY OF THE EXPLANATIONS

One of the motivations of post-hoc interpretability is to use explanations to derive insights into a model's inner workings. Towards this, we investigate the homophily of the selected nodes from our explanations since GNNs are known to exploit the homophily in the neighborhood to learn powerful function approximators. We define homophily of the node as the fraction of the number of its neighbors, which share the same label as the node itself. Intuitively, it should be easier to label a node with the correct label if a larger fraction of nodes in its computational graph shares its label.

In what follows, we use homophily to refer to the homophily of a node with respect to the selected nodes in its first found explanation. True/Predicted homophily refers to the case when true/predicted node labels are used. We investigate the joint distribution of true and predicted homophily exhibited by the studied node sample. The results are shown in Figure 7. We make the following observations.

Observation 1 - Nodes depicting the orange regions on the extreme left side of the plots are nodes that exhibited low true homophily but high predicted homophily. The class labels for such nodes are correctly predicted. However, the corresponding nodes in the explanation were assigned the wrong labels (if they were assigned the same labels as that of the particular node in question, its predicted homophily would have been increased).

Observation 2 - Several vertices corresponding to blue regions spread over the bottom of the plots have low predicted homophily. These nodes are incorrectly predicted, and their label differs from those predicted for the nodes in their explanation set. The surprising fact is that even though some of them have high true homophily close to 1, their predicted homophily is low. This also points to the usefulness of our found explanation in which we conclude that nodes influencing the current node do not share its label.

Observation 3 - We also note that for GIN and APPNP, we have some nodes with true homophily and predicted homophily close to 1 but are incorrectly predicted. This implies the node itself, as well as the most influential nodes from its computational graph, have been assigned the same label. We can conclude that the model based its decision on the right set of nodes but assigned a wrong class to the whole group.

5 CONCLUSION

We propose ZORRO as a post-hoc explanation method for the decisions made by GNN models. Inspired by rate-distortion theory, we frame the problem of explaining GNN models as a feature and node selection problem so as to minimize the expected deviation from the original decision. We proposed a simple combinatorial procedure ZORRO, which retrieves disjoint explanations consisting of *binary masks* for the features and relevant nodes while trying to optimize for fidelity. With our extensive experiments, we show multiple explanations are possible for a given decision, unlike earlier approaches that provide a soft mask. Furthermore, our explanations are sparser and achieve higher fidelity than existing approaches. Finally, our analysis of the homophily in the explanations highlighted differences in the behavior of the models between correctly and wrongly predicted nodes.

REFERENCES

Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In International Conference on Artificial Neural Networks. Springer, 2016.

Brandon Carter, Jonas Mueller, Siddhartha Jain, and David Gifford. What made you do this? understanding black-box decisions with sufficient input subsets. arXiv preprint arXiv:1810.03805, 2018.

Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. Learning to explain: An information-theoretic perspective on model interpretation. arXiv preprint arXiv:1802.07814, 2018.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

Thorben Funke, Tian Guo, Alen Lancic, and Nino Antulov-Fantulin. Low-dimensional statistical manifold embedding of directed graphs. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=SkxQp1StDH.

Maximilian Idahl, Megha Khosla, and Avishek Anand. Finding interpretable concept spaces in node embeddings using knowledge bases. In Machine Learning and Knowledge Discovery in Databases - International Workshops of ECML PKDD 2019, pp. 229–240. Springer, 2019. doi: 10.1007/978-3-030-43823-4_20.

Bo Kang, Jefrey Lijffijt, and Tijl De Bie. Explaine: An approach for explaining network embedding-based link predictions. arXiv preprint arXiv:1904.12694, 2019.

M. Khosla, V. Setty, and A. Anand. A comparative study for unsupervised network representation learning. TKDE, 2019.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (ICLR), 2017.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. International Conference on Learning Representations (ICLR), 2019.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710, 2014.

Phillip E Pope, Soheil Kolouri, Mohammad Rostami, et al. Explainability methods for graph convolutional neural networks. In Proc. of the Conference on Computer Vision and Pattern Recognition, pp. 10772–10781, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135–1144, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

Chris R Sims. Rate–distortion theory and human perception. Cognition, 152:181–198, 2016.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 3319–3328. JMLR. org, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=rJXMpikCZ. accepted as poster.

					Test Accuracy			
Name	**Classes**	**Features**	$\|V\|$	$\|E\|$	**GCN**	**GAT**	**GIN**	**APPNP**
Cora	7	1433	2708	10556	0.794	0.791	0.679	0.799
CiteSeer	6	3703	3327	9104	0.675	0.673	0.480	0.663
PubMed	3	500	19717	88648	0.782	0.765	0.590	0.782

Table 1: Datasets and statistics. The test accuracy is calculated on 1000 nodes.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In ICML, 2015.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In International Conference on Learning Representations, 2019.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings, 2016.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnn explainer: A tool for post-hoc explanation of graph neural networks. arXiv preprint arXiv:1903.03894, 2019.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Invase: Instance-wise variable selection using neural networks. 2018.

Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. Xgnn: Towards model-level explanations of graph neural networks. KDD '20, pp. 430–438. Association for Computing Machinery, 2020.

A APPENDIX

A.1 EXPERIMENTAL SETUP

We focus on explaining the decisions of GNN models with respect to the task of node classification. We fix the number of layers to two for all models and keep the rest of the *model architectures* and parameters as in the original paper. We train the models 200 epochs with ADAM optimizer and a learning rate of 0.01 and a weight decay of 0.0005. We use the model and GNNExplainer implementations of PyTorch Geometric Library (Fey & Lenssen, 2019). For GNNExplainer, we use the default values of 100 epochs and a learning rate of 0.01. For each dataset, we randomly selected 300 nodes and retrieved the explanations of GNNExplainer, and our method for the fidelity values $\tau \in \{0.85, 0.98\}$. We will publish the list of selected nodes together with the implementation. We used 100 samples to calculate the fidelity with Algorithm 2 and set $K = 10$ in our experiments.

Our algorithm has computational complexity of $\mathcal{O}(|G(n)||F| * \text{samples}) * \mathcal{O}(\Phi)$, where $\mathcal{O}(\Phi)$ is the computational complexity of the applied graph neural network. Our implementation is based on PyTorch Geometric 1.6 and Python 3.7. All methods were executed on a server with 128 GB RAM and Nvidia GTX 1080Ti.

Datasets. Three well-known citation network datasets **Cora**, **CiteSeer** and **PubMed** from Yang et al. (2016) where nodes represent documents and edges represent citation links. The class label is described by a similar word vector or an index of category. Statistics for these datasets can be found in Table 1. We used the datasets, including their training and test split from the PyTorch Geometric Library, which corresponds to the data published by Yang et al. (2016).

Models. As models we selected the well-known graph convolutional network (**GCN**) (Kipf & Welling, 2017) and graph attention network (**GAT**) (Veličković et al., 2018) as well as the approximation of personalized propagation of neural predictions (**APPNP**) (Klicpera et al., 2019), and graph

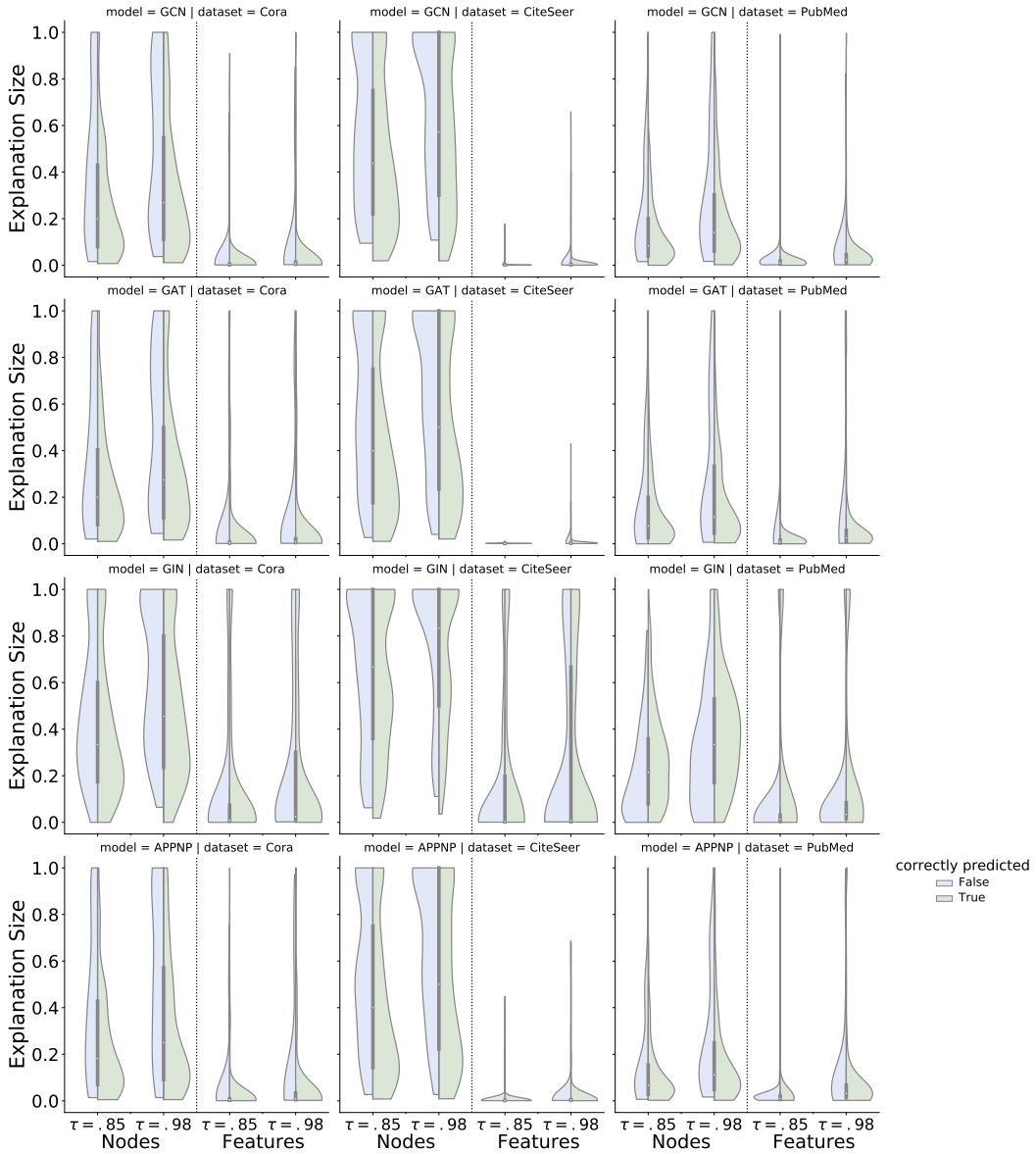

Figure 8: Comparison on explanation size of explanations measured by the proportion of selected elements, i.e. the number of selected nodes is divided by the number of nodes in the computational graph and the number of selected features is divided by the number of features.

isomorphism network (**GIN**) (Xu et al., 2018). APPNP utilizes a connection between PageRank and GCN, especially those with many layers, and extends GCNs based on the personalized PageRank. Xu et al. (2018) proposed GIN to match the representational power of the Weisfeiler-Lehman graph isomorphism test by extending the expressiveness of the feature aggregation.

A.2 Additional Experiment Results

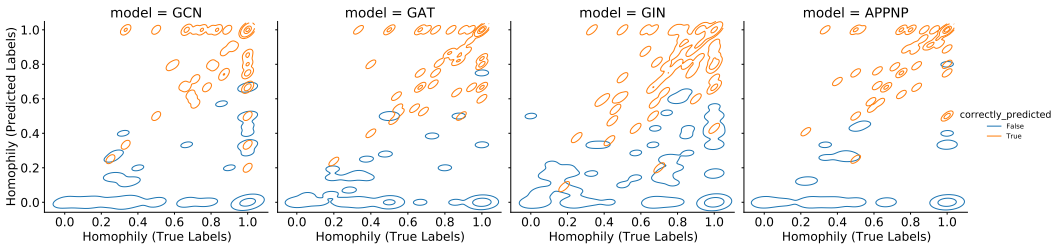

Figure 9: Dataset-Cora. Joint distribution of the homophily with respect to the selected subgraph (in the explanation) with true and predicted labels. The orange contour lines correspond to the distributions for correctly predicted nodes and the blue one correspond to incorrectly predicted nodes.

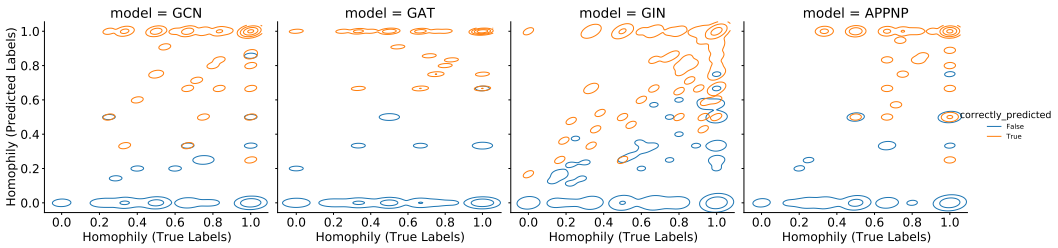

Figure 10: Dataset-CiteSeer. Joint distribution of the homophily with respect to the selected subgraph (in the explanation) with true and predicted labels. The orange contour lines correspond to the distributions for correctly predicted nodes and the blue one correspond to incorrectly predicted nodes.

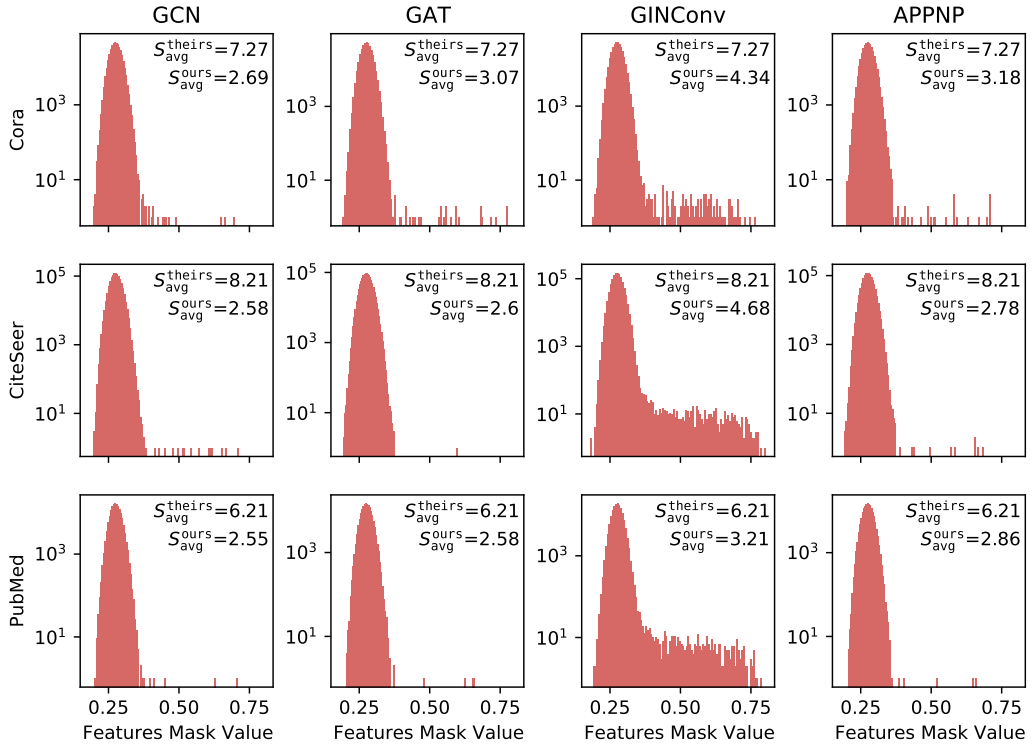

Figure 11: Distribution of the mask values over the node features corresponding to explanations of all explained nodes. $S_{\text{avg}}^{\text{theirs}}$ and $S_{\text{avg}}^{\text{ours}}$ corresponds to the average entropy S of the feature masks retrieved by GNNExplainer and Zorro with $\tau = 0.98$.