GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks

Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang

Abstract—Recently, graph neural networks (GNN) were shown to be successful in effectively representing graph structured data because of their good performance and generalization ability. However, explaining the effectiveness of GNN models is a challenging task because of the complex nonlinear transformations made over the iterations. In this paper, we propose GraphLIME, a local interpretable model explanation for graphs using the Hilbert-Schmidt Independence Criterion (HSIC) Lasso, which is a nonlinear feature selection method. GraphLIME is a generic GNN-model explanation framework that learns a nonlinear interpretable model locally in the subgraph of the node being explained. Through experiments on two real-world datasets, the explanations of GraphLIME are found to be of extraordinary degree and more descriptive in comparison to the existing explanation methods.

Index Terms—Graph neural networks, interpretability, explanation

1 Introduction

Deep Neural Network (DNN) is essentially a new machine learning algorithm based on discriminant model. DNN can model for complex and nonlinear problems and learn the underlying features of data to obtain more abstract features, which can improve the model's capability for prediction or classification. It has a large number of applications such as image recognition, automatic speech recognition, disease diagnosis, etc [1].

Although the high degree of nonlinearity gives DNNs a powerful model representation capability, DNN is a black-box model, which make the model and its predictions hard to be interpreted. Also, people usually will not use a model or a prediction if they do not trust it. For example, if there is a medical diagnosis system that has high accuracy but can not present faithful explanations for its decisions, doctors will not use it. Therefore, it is very important to explore what a DNN model learns from data and why it takes a particular decision in a way that humans can understand. Recent works aimed at interpreting general DNNs mainly focus on two research routes. One approach is to locally approximate models with a simple and interpretable model such as linear regression, which can itself be probed for explanations [2], [3]. The other one is to examine models for relevant components such as identifying the most representative features in the input data [4], [5], [6] or influential input instances [7], [8].

- *Qiang Huang, Yuan Tian, and Yi Chang are with the School of Artificial Intelligence, International Center of Future Science, Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Changchun, Jilin 130012, China, and also with the Innovation Group of Marine Engineering Materials and Corrosion Control, Southern Marine Science and Engineering Guangdong Laboratory, Zhuhai, Guangdong 519000, China. E-mail: huangqiang18@mails.jlu.edu.cn, {yuantian, yichang}@jlu.edu.cn.*
- *Makoto Yamada is with Kyoto University, Kyoto 606-8501, Japan, and also with RIKEN AIP, Kyoto 606-8501, Japan. E-mail: makoto.yamada@riken.jp.*
- *Dinesh Singh is with RIKEN AIP, Kyoto 606-8501, Japan. E-mail: dinesh.singh@riken.jp.*

Manuscript received 8 December 2020; revised 1 June 2022; accepted 18 June 2022. Date of publication 30 June 2022; date of current version 5 June 2023.
This work was supported in part by the National Natural Science Foundation of China under Grants U19A2065 and 61976102 and in part by Jilin Province Youth Science and Technology Program under Grant 20210508060RQ. The work of Makoto Yamada was supported by MEXT KAKENHI under Grant 20H04243.
(Corresponding author: Qiang Huang.)
Recommended for acceptance by Y. Zhang.
Digital Object Identifier no. 10.1109/TKDE.2022.3187455

In many real-world cases, there are considerable amounts of data without regular spatial structures, called non-Euclidean data, and they can be naturally represented as a graph, for example, graph data extracted from social networks, citation networks, electronic transactions, protein structures, molecular structures, and so on. Thus, modeling and analyzing graph data is a challenging task since it needs to combine both feature information of nodes as well as graph information together. Currently, Graph Neural Networks (GNNs) are widely used because of their powerful modeling capability with regard to graphs. GNNs use neural networks to incorporate the feature information of nodes in a graph and as well as the structure information and pass these messages through the edges of the graph in non-Euclidean domains. However, GNNs are notoriously difficult to interpret, and their predictions are hard to explain, similar to that in DNNs. It is important to develop an interpretation method for GNNs because it can improve the transparency of a GNN model and contribute to getting humans to trust the model. Although there exists a large number of interpretation methods designed for DNNs, these methods are not suited for GNN-model interpretation because they do not explicitly use the graph information but perform in Euclidean domains.

Recently, GNNexplainer [9] was proposed. It can find the subgraph and select features of the explained node as explanations, but it mainly focuses on graph structures and not on finding useful features. An alternative idea is to use LIME [3], which uses a linear explanation model to find features as explanations for GNN. However, the performance of LIME can be poor, because LIME does not take the graph structure information into account. Moreover, if the underlying model is highly nonlinear even in the locality of the prediction, such as that in the case of GNN models, a linear explanation model may not be able to produce faithful explanations.

In this paper, we propose another GNN explanation method based on LIME [3] in a nonlinear manner. GraphLIME bases on the following two points: (1) It has been shown that GNN model is usually nonlinear and deeper GNN models perform better than shallow models in GNN for extensive graph data [10], [11], thus using linear model to fit its decision process is unbefitting because of its nonlinearity as shown in Fig. 2. Thus, a nonlinear explanation model is needed to approximate the GNN model. (2) Although the key component of GraphLIME is nonlinear, its interpretability is mathematically guaranteed (See Section 3.4). More specifically, GraphLIME is a model-agnostic and nonlinear approach for providing locally faithful explanations for GNN-based models in a subgraph, whose procedure with a toy explanation sample is shown briefly in Fig. 1.

2 Related Work

There are two main families of models to provide interpretability for neural models. The first family of models focuses on finding a simple and proxy model to interpret the model being explained in a model-agnostic way. For example, LIME [3] has been proposed to explain the predictions of any classifier by learning an interpretable linear model locally around the prediction. Black Box Explanations through Transparent Approximations (BETA) [2] is another model-agnostic framework to explain the behavior of the model being explained by simultaneously optimizing for fidelity to the original model and interpretability of the explanation. DeepRED [12] was proposed to extract rules from deep neural networks and generate explanations for predictions, and ANN-DT [13] was proposed to extract binary decision trees from neural networks, it is an interpretable model to interpret the main model.

The second family of models focuses on the relevant aspects of computation in the neural model being explained. Erhan *et al.* [14]

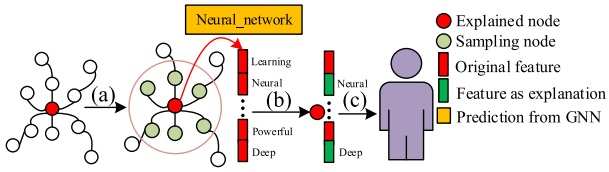


Fig. 1. A toy sample of GraphLIME for explanation of Graph Neural Neworks.

proposed to inspect feature gradients to find good qualitative interpretations of high-level features represented by neural models. DeepLIFT [15] was proposed to enable interpretability of neural networks by comparing the activation of each neuron to its 'reference activation' and assigning contribution scores according to the difference. The method proposed in [6] uses two fundamental axioms and a standard gradient operator to attribute the prediction of a deep network to its input features. Instead of creating surrogate models, some methods has been proposed to identify the patterns of input data and find influential samples for relevant information [7], [8]. However, few studies have focused on the structural information of graphs with these kinds of interpretation methods. Recently, perturbation-based approachs for GNN explanation are proposed. For example, [9] was proposed to utilize mutual information to find a subgraph with associated features for interpreting GNN models; PGExplainer [16] learns a parameterized model to predict whether an edge is important; SubgraphX [17] explains GNNs by exploring and identifying important subgraphs; GraphSVX [18] utilizes decomposition technique to explain GNNs based on the Shapley Values from game theory. Beyond that, RelEx [19] is a general relational model explanation framework to explain black-box relational models with only access to the outputs of the black-box.

3 Proposed Method: GraphLIME

3.1 Formulation of GraphLIME Explainer

Formally, let $\mathcal{X} \subset \mathbb{R}^d$ be the domain of a vector $\boldsymbol{x}$. For each node v_i in a graph, we have an associated feature vector $\boldsymbol{x}_i$, where each feature is extracted from the real world and can be understood by humans. We define an explanation as a model $g \in G$, where G is a class of *interpretable* models such as linear regressions, decision trees etc. In other words, a model $g \in G$ can provide intuitional interpretations in an interpretable manner. The explanation model class G used in this paper is Hilbert-Schmidt Independence Criterion Lasso (HSIC Lasso) [20], [21], [22], which is a kernel based nonlinear interpretable feature selection algorithm. We will discuss more details about this in a later subsection. Let the domain of the explanation model $g \in G$ also be $\mathcal{X} \subset \mathbb{R}^d$, which means that g acts over the original features that are understandable.

Let the GNN model being explained be denoted by $f : \mathbb{R}^d \to \mathbb{R}$. In a classification problem, $f(\boldsymbol{x}_i)$ denotes the probability (or binary indicator) that the instance $\boldsymbol{x}_i$ belongs to a certain class . Let v denote the node whose prediction needs to be explained, $\boldsymbol{X}_n \in \mathbb{R}^{n \times d}$ represents the sampling information matrix that can capture the locality of the explained node v, where n is the number of neighbors of v. Then, we obtain an interpretable model $g \in G$ using the local information matrix $\boldsymbol{X}_n$ of v to approximate f, and generate the locally faithful explanations based on g.

Given a GNN model f, the node v being explained, the sampling local information matrix $\boldsymbol{X}_n$ of v, and an interpretable explanation model g, the explanations for the explained v are obtained as follows:

$$\zeta(v) \leftarrow \underset{g \in G}{\operatorname{argmin}}\ g(f, \boldsymbol{X}_n), \tag{1}$$

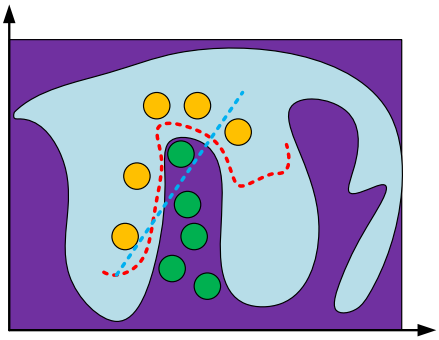


Fig. 2. A toy sample to present the advantage of nonlinear explanation model comparing to linear explanation model.

where $\zeta(v)$ is the set of features as explanations of the node being explained v, it is generated based on the optimal explanation model g.

3.2 Sampling for Local Exploration by N-Hop Network Neighbors

In a graph, a node's local information is decided by its neighbors in a subgraph; hence, the explainers for a GNN model in a graph should not only focus on the features of the given node but also the correlations between the features of its neighbouring nodes, because a given target node needs to aggregate information from its neighboring nodes. In order to learn the local behavior of a GNN model being explained, we consider N-hop network neighbors to sample the neighboring nodes of a given node. A node v's N-hop network is defined as the network formed by v and the nodes whose distance from v is within N hops and links. By performing N-hop network sampling, we obtain:

$$\boldsymbol{X}_n = [\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_m]$$

where $\boldsymbol{x}_i$ is the corresponding associated feature vector of v_i, $v_i \in \mathcal{S}_N$, $\mathcal{S}_N$ is the set of N-hop neighbors of the explaining node v, and m is the number of N-hop neighbors of v_i. Moreover, we can also obtain prediction $y_i = f(\boldsymbol{x}_i)$ for a given GNN model f, which can be used as the label in the explanation model g.

3.3 Nonlinear Explanation Model: HSIC LASSO

We consider a feature-wise kernelized nonlinear method called Hilbert-Schmidt Independence Criterion Lasso (HSIC Lasso) [20], [21], [22] as the explanation model for GNN models. The HSIC Lasso is a "supervised" nonlinear feature selection method. Given a GNN model f, a node v whose prediction needs to be explained, the neighbor set $\boldsymbol{X}_n$ of node v, and "supervised" paired N-hop neighboring nodes $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \boldsymbol{X}_n$, the probability $y_i = f(\boldsymbol{x}_i)$ that x_i belongs to a certain class is used as the *label* of the HSIC Lasso explanation model. For each prediction being explained, the HSIC Lasso optimization problem for an explanation model $g \in G$ is given as

$$\begin{aligned} &\min_{\boldsymbol{\beta} \in \mathbb{R}^d} \frac{1}{2} \|\bar{\boldsymbol{L}} - \sum_{k=1}^{d} \beta_k \bar{\boldsymbol{K}}^{(k)}\|_F^2 + \rho \|\boldsymbol{\beta}\|_1 \\ &\text{s.t.} \quad \beta_1, \ldots, \beta_d \geq 0, \end{aligned} \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\rho \geq 0$ is the regularization parameter, $\|\cdot\|_1$ is the ℓ_1 norm to enforce sparsity, $\bar{\boldsymbol{L}} = \boldsymbol{HLH}/\|\boldsymbol{HLH}\|_F$ is the normalized centered Gram matrix, $\boldsymbol{L}_{ij} = L(y_i, y_j)$ is the kernel for the output, $\boldsymbol{H} = \boldsymbol{I}_n - \frac{1}{n}\boldsymbol{1}_n\boldsymbol{1}_n^\top$ is the centering matrix, $\boldsymbol{I}_n$ is the n-dimension identity matrix, $\boldsymbol{1}_n$ is the n-dimension vector whose elements are all 1, $\bar{\boldsymbol{K}}^{(k)} = \boldsymbol{HK}^{(k)}\boldsymbol{H}/\|\boldsymbol{HK}^{(k)}\boldsymbol{H}\|_F$ is the normalized centered Gram matrix for the k-th feature, and $[\boldsymbol{K}^{(k)}]_{ij} = K(\boldsymbol{x}_i^{(k)}, \boldsymbol{x}_j^{(k)})$ is the kernel for the k-th

dimensional input. In this paper, we use the Gaussian kernel for both input and the predictions of $\boldsymbol{X}_n$ from the given GNN model:

$$K(\boldsymbol{x}_i^{(k)}, \boldsymbol{x}_j^{(k)}) = \exp\left(-\frac{(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_j^{(k)})^2}{2\sigma_x^2}\right),$$

$$L(y_i, y_j) = \exp\left(-\frac{\|y_i - y_j\|_2^2}{2\sigma_y^2}\right),$$

where σ_x and σ_y are the Gaussian kernel widths. Note that we consider the complete graph in this paper. However, we can explicitly take the local graph information into account by modifying $\boldsymbol{K} \circ \boldsymbol{A} \rightarrow \boldsymbol{K}$ and $\boldsymbol{L} \circ \boldsymbol{A} \rightarrow \boldsymbol{L}$ respectively, where $\boldsymbol{A} \in \{0,1\}^{n \times n}$ is an adjacency matrix with self-loop and $\circ$ is the elementwise product.

We employ the nonnegative least angle regression [23] to optimize the Eq. (2), and then we can obtain the coefficient vector $\boldsymbol{\beta}$ and select the top-K features as the explanations for the prediction of the node being explained based on it.

3.4 Interpretation of HSIC Lasso

Here, we present the interpretation of the nonlinear explanation model HSIC Lasso. The HSIC Lasso contains the main concept of minimum redundancy maximum relevancy (mRMR) [24], which is a widely used classical supervised feature selection algorithm that can find non-redundant features with strong dependence on the output values. We can rewrite the first term of Eq. (2) as

$$\begin{aligned}&\frac{1}{2}\|\bar{\boldsymbol{L}} - \sum_{k=1}^{d} \beta_k \bar{\boldsymbol{K}}^{(k)}\|_F^2 \\ &= \frac{1}{2}\sum_{k,m=1}^{d} \beta_k \beta_m \mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{f}_m) - \sum_{k=1}^{d} \beta_k \mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{y}) + \frac{1}{2} \qquad (3)\end{aligned}$$

where $\boldsymbol{f}_k \in \mathbb{R}^n$ is the feature vector corresponding to the k-th feature, $\mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{y}) = \mathrm{tr}(\bar{\boldsymbol{K}}^{(k)}\bar{\boldsymbol{L}})$ is the normalized variant of the empirical estimate of the Hibert-Schmidt independence criterion (HSIC) [25], $\mathrm{NHSIC}(\boldsymbol{y}, \boldsymbol{y}) = 1$ is constant, and $\mathrm{tr}(\cdot)$ is the trace operator. HSIC, which is based on a universal reproducing kernel such as the Gaussian kernel, is a non-negative function that estimates the independence between two random variables. A larger HSIC value indicates more dependency between the two variables, and it is zero if and only if the two random variables are statistically independent. The proof of Eq. (3) is left to Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2022.3187455.

Algorithm 1. Locally Nonlinear Explanation: GraphLIME

Input: GNN classifier f, Number of explanation features K
Input: the graph $\mathcal{G}$, the node v being explained
Output: K explanation features
1: $\boldsymbol{X}_n = N\_hop\_neighbor\_sample(v)$
2: $\boldsymbol{Z} \leftarrow \{\}$
3: **for all** $x_i \in \boldsymbol{X}_n$ **do**
4: $\quad y_i = f(\boldsymbol{x}_i)$
5: $\quad \boldsymbol{Z} \leftarrow \boldsymbol{Z} \cup (\boldsymbol{x}_i, y_i)$
6: **end for**
7: $\boldsymbol{\beta} \leftarrow$ HSIC Lasso$(\boldsymbol{Z})$ ▷ with x_i as features, y_i as label
8: $\zeta(v) \leftarrow$ Top-K features as explanations based on $\boldsymbol{\beta}$

In Eq. (3), we ignore the value of $1/2$ because it is constant. We consider the values of $\mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{y})$ and $\mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{f}_m)$. For $\mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{y})$, if there is strong dependency between the k-th feature vector $\boldsymbol{f}_k$ and the output vector $\boldsymbol{y}$, the value of $\mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{y})$ should be large and the corresponding coefficient β_k should also take a large value in order to minimize Eq. (2). Meanwhile, if $\boldsymbol{f}_k$ is independent of $\boldsymbol{y}$, the value of $\mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{y})$ should be small so that β_k tends to be eliminated by l_1-regularizer. This property can help select the most relevant features from the output vector $\boldsymbol{y}$.

For $\mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{f}_m)$, if $\boldsymbol{f}_k$ and $\boldsymbol{f}_m$ are strongly dependent (i.e., redundant features), the value of $\mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{f}_m)$ should be large and either of the two coefficients β_k and β_m tends to be zero in order to minimize the Eq. (2). This means that the redundant features will not be selected by HSIC Lasso.

In Algorithm 1, we summarize the model-agnostic and nonlinear explanation framework based on N-hop network sampling and HSIC Lasso for GNNs as a procedure, which we call GraphLIME. For GraphLIME, the NHSIC operator calculates the empirical HSIC value asymptotically converging to the true HSIC in $O(1/\sqrt{|S_N|})$ according to Theorem 3 in [25], and the memory complexity is $O(d|S_N|^2)$.

4 Experiments

4.1 Setting

We trained the GraphSAGE [26] and GAT [27], which are widely used GNN models, for the following explanation experiments. We performed simulated user experiments to evaluate the effectiveness of the proposed framework GraphLIME and other explanation methods. More specifically, we compared the proposed framework GraphLIME with the LIME [3] framework, which utilizes the perturbation method to sample data and train a linear explanation model lasso for selecting features as explanations according to the coefficients from the linear explanation model. In addition, we compared it with GNNexplainer [9] which utilizes mutual information to find a subgraph with associated features for interpreting GNN models, and SHAP framework [5] which calculating the SHAP values as a unified measure of feature importance . We also compared it with a method based on the greedy procedure [28], which greedily removes the most contributory features of the prediction until the prediction changes, and the random procedure, which randomly selects K features as the explanations.

In our experiments, we used two graph datasets, namely Cora and Pubmed. Cora and pubmed are two publication datasets, each feature indicates the absence/presence of the corresponding word in Cora and the TF/IDF value of the corresponding word in Pubmed. More statistical details of the Cora and Pubmed datasets can be found in Appendix, available in the online supplemental material. We sampled the 2-hop network neighbors of the node being explained and then randomly split the datasets into training sets (80%) and testing sets (20%).

4.2 Does the Explanation Framework Filter Useless Features?

In the first simulated user experiment, we investigated whether GraphLIME could filter useless features and select informative features as the explanations. For this, we compared the frequency of samples on different number of selected "noisy" features over different explanation frameworks to compare their abilities of denoising data.

Concretely, we artificially and randomly added 10 "noisy" features in each sample's feature vector and then trained a GraphSAGE model or GAT model whose test accuracy was more than 80%. Thus we obtained the useless set of features (the 10 "noisy features") for the trained model. Finally, we set the number of selected features K as 10 and produced explanations on 200 test samples for each explanation framework and compared their performance in terms of the frequency distribution of samples on different number of noisy features.

We used histograms to plot the frequency distributions of samples on different number of noisy features for the six different explanation frameworks on Cora and Pubmed. The distribution is shown in Fig. 3. It can be seen that the number of noisy features selected by the proposed framework GraphLIME and GNNexplainer are in general less than that selected by the other explanation frameworks. For

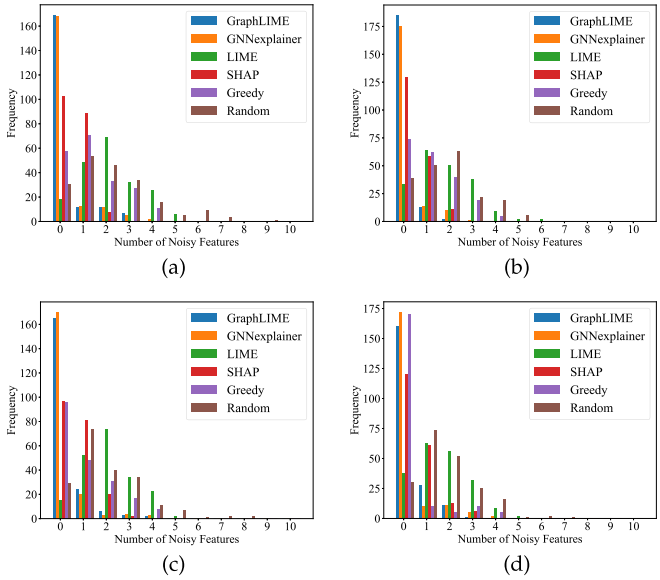


Fig. 3. Distribution of noisy features on (a) Cora for GraphSAGE; (b) Pubmed for GraphSAGE; (c) Cora for GAT; (d) Pubmed for GAT.

GraphLIME and GNNexplainer, the frequency of samples on different number of selected noisy features mainly focus on 0. This means that the GraphLIME and GNNexplainer frameworks rarely select useless features as explanations, which is very useful when the graph data has a large amount of noise. LIME is not capable of ignoring the noisy features, and the distribution of LIME is mainly around 1 to 4. For Random procedure, the frequency distribution is similar to that of LIME. The SHAP and Greedy explanation procedure are slightly better than LIME and Random, but not comparable with GraphLIME and GNNexplainer. The results demonstrate that the proposed GraphLIME can denoise graph data for the GNN model when the data contains a large amount of noise and that it is more capable of finding informative features as explanations.

4.3 Do I Trust This Prediction?

The prediction made by a classifier model might not be credible; therefore, it is important for an explanation framework that the explanations can aid users in deciding whether a prediction is trustworthy. In this experiment, we compared this kind of ability for different explanation frameworks.

Firstly, we randomly selected 30% of the features as "untrustworthy" features and denote the randomly selected feature set as $\boldsymbol{f}_{un}$. Then, we trained a GraphSAGE or GAT classifier and obtained the predictions on testing samples. We assumed that the users could identify these "untrustworthy" features and that they would not want these features as explanations. Secondly, we developed *oracle* "trustworthiness" for each prediction of test samples as follows:

$$l_i = \begin{cases} 1 & (trustworthy), \quad if \quad y_i = y_i' \\ 0 & (untrustworthy), \quad if \quad y_i \neq y_i', \end{cases}$$

where y_i is the prediction from the trained classifier for sample i, y_i' is the prediction after removing those "untrustworthy" features in $\boldsymbol{f}_{un}$. We regard *oracle* "trustworthiness" l_i as the true label to decide whether a prediction is credible. Thirdly, for GraphLIME and LIME, we assumed that the simulated users regard predictions from the classifier as "untrustworthy" if the prediction made by another approximation linear model changes when all "untrustworthy" features that appear in the explanations are removed. For GNNexplainer, Greedy, and Random explainer, the prediction was deemed untrustworthy by the users if any "untrustworthy" features appeared in their explanations. Finally, we compared the decisions made by the simulated users with the *oracle* "trustworthiness".

We set the number of selected features as $K = 10, 15, 20, 25$ and reported the averaged F1-Score on the trustworthy predictions for each explanation framework over 100 rounds. The results in Table 1 show that GraphLIME is superior to the other explanation methods. The lower F1-Score of the other explanation methods indicate that they achieved a lower precision (i.e., trusting too many predictions) or a lower recall (i.e., mistrusting predictions more than they should), while the higher F1-Score of the proposed framework GraphLIME indicates that it achieved both high precision and high recall.

4.4 Does the Explainer help to Identify the Better Model?

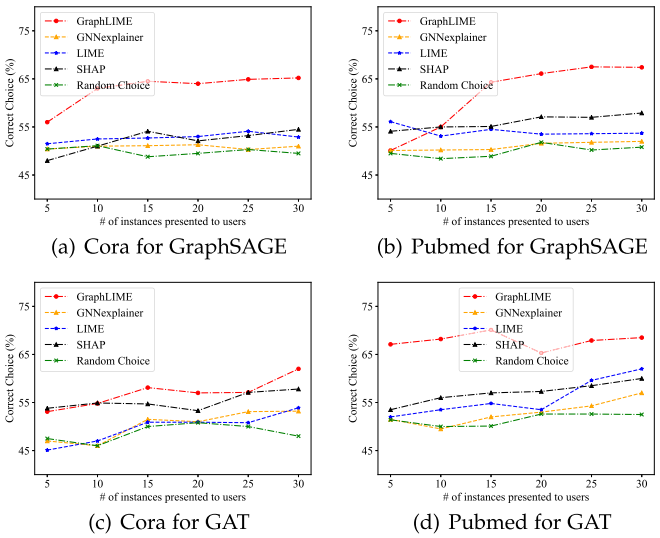
In this experiment, we explored whether the explanation framework could be used for model selection and for guiding users to choose the *better* one among two different GNN classifiers. We compared the performance of different frameworks in selecting the *better* classifier by reporting the accuracy of selecting the real better classifier.

In order to simulate a situation where the models use not only informative features in the real world but also ones that introduce spurious relations, We artificially added 10 noisy features to the data, and marked them as untrustworthy features. We created pairs of classifiers by repeatedly training pairs of GraphSAGE or GAT classifiers until their training accuracy and test accuracy were both above 70% and the difference in their test accuracy was more than 5%. Then, the number of selected features K is set as 10 and we obtained explanations of the samples being explained from the explanation framework on two competing classifiers and recorded the number of untrustworthy features appearing in the

TABLE 1
Average F1-Score (%) of Trustworthiness for Different Explainers

K	10				15				20				25			
	GraphSAGE		GAT		GraphSAGE		GAT		GraphSAGE		GAT		GraphSAGE		GAT	
	Cora	Pub	Cora	Pub	Cora	Pub	Cora	Pub	Cora	Pub	Cora	Pub	Cora	Pub	Cora	Pub
Random	40.4	21.9	37.1	36.2	37.7	16.9	23.2	27.6	36.6	15.3	24.1	31.7	36.0	14.6	17.9	23.5
Greedy	71.7	63.5	72.9	55.8	71.6	62.8	63.8	53.3	71.5	62.4	61.9	51.4	71.5	62.6	61.8	52.4
GNNexplainer	77.6	79.5	73.7	70.8	71.6	77.6	79.0	69.4	77.1	79.3	72.4	65.6	76.5	75.1	63.8	61.7
SHAP	78.08	73.6	62.2	69.9	78.29	71.2	67.3	68.7	77.7	76.7	60.8	68.1	77.1	74.1	61.7	67.3
LIME	89.3	83.6	91.3	79.1	89.7	84.0	87.6	76.4	89.8	84.5	87.9	70.3	89.8	84.6	87.6	71.3
GraphLIME	**95.3**	**92.5**	**96.1**	**92.1**	**95.6**	**92.3**	**96.2**	**91.2**	**95.4**	**91.6**	**95.1**	**92.0**	**95.4**	**91.5**	**95.3**	**92.0**

('Pub' denotes 'Pubmed' dataset).



(a) Cora for GraphSAGE (b) Pubmed for GraphSAGE

(c) Cora for GAT (d) Pubmed for GAT

Fig. 4. The performance of identifying better model.

explanations for the two classifiers. Note that the *better* classifier should have fewer untrustworthy features in its explanations; therefore, we selected the classifier with fewer untrustworthy features as the better classifier and compared this choice with the real better classifier with higher test accuracy.

The goal of this experiment was to verify whether the explanation framework could guide users to identify better models based on explanations from the explanation framework. Considering that users may not have time to examine a large number of samples, let B denote the number of samples being explained, which they are willing to look at in order to identify the better classifier. We use Submodular Pick [3] to select B samples and present these samples to the users for examining, Submodular Pick will be detailed in the Appendix, available in the online supplemental material.

We present the accuracy of selecting the real better classifier as B varies from 5 to 30, averaged over 200 rounds, in Fig. 4. We compared the performance of GraphLIME with that of SHAP, LIME and GNNexplainer, and omitted Greedy and Random because they could not produce useful explanations as shown in the previous two experiments. Moreover, we also plotted the accuracy curve for random choice. These results demonstrate that the proposed method GraphLIME can be used to perform model selection and that it outperforms SHAP, LIME and GNNexplainer, which are only slightly better than the random choice. And it is worthy to note that the performance of GraphLIME also improved With the increase in number of presented instances.

5 Conclusion

Most of the explanation methods for neural models are designed for general neural networks, while only few works exist for GNNs. In this paper, we presented a model-agnostic local interpretable explanation framework for GNN, which we call GraphLIME. It is able to leverage the feature information of the N-hop network neighbors of the node being explained and their predicted labels in a local subgraph, utilize the Hilbert-Schmidt Independence Criterion Lasso (HSIC Lasso), which is a nonlinear interpretable model for capturing the nonlinear dependency between features and predicted outputs, and produce finite features as the explanations for a particular prediction. Experiments on two real-world graph datasets for two kinds of GNN models demonstrated the effectiveness of the proposed framework. It could filter noisy features and select the real informative features, guide users in ascertaining trust in predictions, and help users to identify the better classifier.

Acknowledgments

Makoto Yamada and Yi Chang are Co-Corresponding Authors.

References

[1] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.
[2] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec, "Interpretable & explorable approximations of black box models," 2017, *arXiv:1707.01154*.
[3] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144.
[4] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan, "Learning to explain: An information-theoretic perspective on model interpretation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 883–892.
[5] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4765–4774.
[6] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.
[7] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1885–1894.
[8] C.-K. Yeh, J. Kim, I. E.-H. Yen, and P. K. Ravikumar, "Representer point selection for explaining deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9291–9301.
[9] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating explanations for graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9240–9251.
[10] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 5680–5695, 2020.
[11] W. Hu, M. Fey, H. Ren, M. Nakata, Y. Dong, and J. Leskovec, "OGB-LSC: A large-scale challenge for machine learning on graphs," 2021, *arXiv:2103.09430*.
[12] J. R. Zilke, E. L. Mencía, and F. Janssen, "Deepred–rule extraction from deep neural networks," in *Proc. Int. Conf. Discov. Sci.*, 2016, pp. 457–473.
[13] G. P. Schmitz, C. Aldrich, and F. S. Gouws, "ANN-DT: An algorithm for extraction of decision trees from artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1392–1401, Jun. 1999.
[14] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," Univ. Montreal, QC, Canada, Tech. Rep. 1341, 2009.
[15] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3145–3153.
[16] D. Luo *et al.*, "Parameterized explainer for graph neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 19 620–19 631.
[17] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, "On explainability of graph neural networks via subgraph explorations," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12 241–12 252.
[18] A. Duval and F. D. Malliaros, "GraphSVX: Shapley value explanations for graph neural networks," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2021, pp. 302–318.
[19] Y. Zhang, D. Defazio, and A. Ramesh, "Relex: A model-agnostic relational model explainer," in *Proc. AAAI/ACM Conf. AI Ethics Soc.*, 2021, pp. 1042–1049.
[20] M. Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama, "High-dimensional feature selection by feature-wise kernelized lasso," *Neural Computation*, vol. 26, no. 1, pp. 185–207, 2014.
[21] M. Yamada *et al.*, "Ultra high-dimensional nonlinear feature selection for big biological data," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1352–1365, Jul. 2018.
[22] H. Climente-González, C.-A. Azencott, S. Kaski, and M. Yamada, "Block HSIC lasso: Model-free biomarker detection for ultra-high dimensional data," *Bioinformatics*, vol. 35, no. 14, pp. i427–i435, 2019.
[23] B. Efron *et al.*, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.
[24] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[25] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Proc. Int. Conf. Algorithmic Learn. Theory*, 2005, pp. 63–77.
[26] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[27] J. B. Lee, R. Rossi, and X. Kong, "Graph classification using structural attention," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1666–1674.
[28] D. Martens and F. Provost, "Explaining data-driven document classifications," *Mis Quart.*, vol. 38, no. 1, pp. 73–100, 2014.