Explaining Graph Neural Networks Using Interpretable Local Surrogates

Farzaneh Heidari [1 2 *] **Perouz Taslakian** [3 *] **Guillaume Rabusseau** [1 2 4]

Abstract

We propose an interpretable local surrogate (ILS) method for understanding the predictions of black-box graph models. Explainability methods are commonly employed to gain insights into black-box models and, given the widespread adoption of GNNs in diverse applications, understanding the underlying reasoning behind their decision-making processes becomes crucial. Our ILS method approximates the behavior of a black-box graph model by fitting a simple surrogate model in the local neighborhood of a given input example. Leveraging the interpretability of the surrogate, ILS is able to identify the most relevant nodes contributing to a specific prediction. To efficiently identify these nodes, we utilize group sparse linear models as local surrogates. Through empirical evaluations on explainability benchmarks, our method consistently outperforms state-of-the-art graph explainability methods. This demonstrates the effectiveness of our approach in providing enhanced interpretability for GNN predictions.

1. Introduction

Graph Neural networks (GNNs) (Scarselli et al., 2008) are neural architectures that are widely used for analyzing data with complex relational structures. They are shown to be effective for a range of applications such as traffic prediction (Cui et al., 2019) and drug discovery (Zhu et al., 2022a). However, understanding how GNNs make predictions remains a challenge due to the multi-layer computations in the GNN architecture and the graph-based nature of the input: unlike images or text, graphs are not easily reasoned about by humans. Explaining graph neural networks has thus emerged as a field of study that is concerned with explaining why a given black-box GNN – whose inner architecture is not readily available – predicts a specific output.

Graph Neural Network Explainability focuses on developing methods to interpret GNNs, allowing users to trust and comprehend their outputs. Various techniques are employed, including visualizing learned representations (Liu et al., 2022), designing attribution methods (Sanchez-Lengeling et al., 2020), and exploring adversarial attacks (Lucic et al., 2022). These approaches enhance transparency, accountability, and fairness of the models.

Graph explainability methods fall into two general categories (Yuan et al., 2022): *model-level* (global), and *instance-level*. Global explanation methods try to learn a model of the data and output the global relationship between the variables (Fan et al., 2022; Finzel et al., 2022). Instance-level explanations methods (Luo et al., 2020; Ying et al., 2019) focus on explaining the prediction of the GNN for a specific instance, which could be a node or a graph depending on the task, and output a set of most relevant nodes/edges or features. Both approaches improve transparency of the black-box graph neural network models and have various applications (Zhu et al., 2022b; Henderson et al., 2021).

In this work, we propose an instance-level graph explainability method named *Interpretable Local Surrogates* (ILS). ILS approximates the graph black-box model in the local neighbourhood of an instance with an interpretable surrogate that can be probed to explain the prediction. Our method is flexible in choice of surrogates, allowing us to select different interpretable models tailored to the specific prediction explanation requirements. Furthermore, our method can explain any black-box graph model on both node-level, edge-level and graph-level tasks. In this paper, we focus on the class of sparse linear functions, and leverage group sparsity constraints to efficiently identify a set of the most relevant nodes or features. To the best of our knowledge, our approach is the first one that can at the same time handle node and graph level models and do not require any knowledge of the black-box model (such as intermediate node embeddings or back-propagating through the original model).

We conduct comprehensive evaluations of our method on both graph-level and node-level experiments, comparing its performance against state-of-the-art explainability methods.

[*]Work done while at Samsung AI Center, Montreal, Canada. [1]DIRO, Université de Montréal. [2]Mila. [3]ServiceNow Research, Montréal, Canada. [4]CIFAR AI Chair. Correspondence to: Farzaneh Heidari <farzaneh.heidari@mila.quebec>.

Proceedings of the 2^{nd} Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023.

Across all datasets, our method consistently outperforms existing approaches or achieves comparable performance. For graph regression, we introduce a novel synthetic dataset that incorporates dependencies on both graph structure and node labels. This unique dataset adds an additional layer of complexity to the existing synthetic datasets commonly used for graph explainability evaluations. By evaluating our method in these diverse settings, we demonstrate its effectiveness in addressing explainability challenges in graph-based tasks.

Summary of Contributions We propose a novel instance-based graph explainability method that efficiently identifies the set of nodes explaining the prediction of a black-box model applied to a given input. Our method relies on the simple idea of approximating the black-box function *locally*, in the neighborhood of the input node or graph, with an interpretable surrogate model. By choosing the class of group-sparse linear models for the surrogate, our method can identify the nodes that most impact the output of the GNN locally. Our approach, which we call *Explainability with Interpretable Local Surrogate* (ILS), has the following key features:

- ILS can explain any black-box graph model, without the need for further knowledge about the inner workings of the model.
- ILS can provide instance-based explanations for both node-level, edge-level and graph-level models.
- ILS relies on a *local* approximation, without the need to globally distill the black-box model in a surrogate. This implies, in particular, that the complexity of explaining the prediction of a node-level model on one node is independent of the total number of nodes in the graph (in contrast to the DnX (Pereira et al., 2023) method, which requires distilling a linear GNN on the entire set of nodes).
- ILS provides a general explainability framework combining the fundamental ideas of local approximation and interpretable surrogates. While in this paper we solely use group sparse linear models as surrogates (allowing ILS to provide a set of nodes as explanation), any other interpretable surrogate can be used, depending on the nature of the desired explanation. This makes ILS a very versatile and flexible explanation method for black-box graph models.

1.1. Related Work

Gradient-based explainability methods The most straight-forward way to approximate the importance of features is by looking at the magnitude of the gradients or hidden feature map values. This method has been widely used for images, texts and graphs (Yuan et al., 2022). For graph models, SA (Baldassarre & Azizpour, 2019) uses the squared-norm of the gradients of the differentiable function f to produce a saliency map of the input. Grad-CAM (Selvaraju et al., 2017) computes the gradients of the target prediction with respect to the intermediate node embeddings and then averages the scores of all the layers to find the final scores. Note that Grad-CAM requires access to the output of the intermediate layers of the deep neural network.

Perturbation-based explainability methods Perturbation-based methods are widely used to explain deep image and graph models (Yuan et al., 2022). The motivation is similar to gradient methods: the variations in the output with respect to different input perturbations is used to approximate feature importance. Perturbation-based methods like GNNExplainer (Ying et al., 2019), PGExplainer (Vu & Thai, 2020), CF-GNNExplainer (Lucic et al., 2022), learn a node/edge or feature mask on the data by maximizing various objective functions.

Surrogate methods Most relevant to our work are methods relying on *surrogates*. These methods approximate the black-box model with a simple interpretable function, either locally (around the instance to be explained) or globally (akin to distillation methods). This family of methods include *PGM-Explainer* (Vu & Thai, 2020), which locally fits a Bayesian network to identify relevant nodes, GraphLIME (Huang et al., 2022), which uses sparse models as local surrogates to identify the most important node features, and DnX (Pereira et al., 2023), which globally distills the black-box model into a linear GNN, named SGC.

Limitations of existing methods GNNExplainer (Ying et al., 2019) and PGexplainer (Luo et al., 2020) assume access to the Graph Neural Network (GNN) and rely on intermediate node embeddings or back-propagation through the GNN (Pereira et al., 2023). However, this assumption limits their applicability as it requires modifying or accessing specific parts of the model architecture, which may not be feasible in practice. In an attempt to address this limitation, DnX (Pereira et al., 2023) proposes a global interpretable surrogate for GNNs. While DnX is fast and demonstrates good performance, it encounters challenges with large datasets as it needs to find a global surrogate fitting all nodes in the graph, even when explanations are only needed for a subset of nodes. Additionally, due to the limited expressiveness of the global surrogate model fitted on the entire dataset, DnX does not perform well on graph-level tasks.

2. Background

2.1. Notation

A graph $G = (V, E)$ is given by a finite set of vertices V and an edge set $E \subset V \times V$. For attributed graphs, we use $X \in \mathbb{R}^{n \times d}$ to denote the feature matrix whose rows contain the feature vectors $x_v \in \mathbb{R}^d$ for each vertex $v \in V$. Given a node $v \in V$, we use the notation $N(v)$ for the set of its neighbors in G. We use lower-case letters for vectors, upper-case letters for matrices and calligraphic letters for higher-order tensors. We denote the number of non-zero rows of a matrix W by $\|W\|_0 = |\{i \mid W_{:,i} \neq 0\}|$, and use $X_{i,:}$ (resp. $X_{:,j}$) to denote the ith row (resp. jth column) of $X \in \mathbb{R}^{n \times d}$. Let $[k]$ denote the set of integers from 1 to k. Given a subset of row indices $S \subset [n]$, we use $X_{S,:} \in \mathbb{R}^{|S| \times d}$ to denote the matrix X restricted to the rows in S. These notations are straightforwardly extended to higher-order tensors. We use $\mathrm{vec}(X) \in \mathbb{R}^{nd}$ to denote the vectorization of a matrix $X \in \mathbb{R}^{n \times d}$. The inner product of two matrices of the same size A and B is defined by $\langle A, B \rangle = \mathrm{vec}(A)^\top \mathrm{vec}(B)$. Given a tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, we use $\mathcal{T}_{(1)} \in \mathbb{R}^{d_1 \times d_2 d_3}$ to denote its mode-1 matricisation (Kolda & Bader, 2009), which is the matrix with rows $\mathcal{T}_{:,i,j}$ for $i \in [d_2], j \in [d_3]$.

2.2. Graph Neural Networks

Let $G = (V, E)$ be an undirected graph with nodes $v_i \in V$ and edges $(v_i, v_j) \in E$, such that each node has a feature vector x_i. A GNN applied to G performs iterative message passing and aggregation steps to learn representations of the nodes (and/or edges) based on a given task. At each layer t, the representation $h_i^{(t-1)}$ of node v_i is updated by aggregating the representations of its neighboring nodes, $h_j^{(t-1)}$, using an aggregation function

$$m_{N(v_i)}^t = \text{AGGREGATE}^{(t)}(\{h_j^{(t-1)}, \forall j \in N(v_i) \cup \{v_i\}\}) \tag{1}$$

where $h_i^{(0)} = x_i$. The updated representation $h_i^{(t)}$ is obtained by combining the received messages from neighboring nodes using an update function.

$$h_i^{(t)} = \text{UPDATE}^{(t)}(h_i^{(t-1)}, m_{N(v_i)}^t). \tag{2}$$

3. Method

In this section we present our approach for extracting instance-based explanations from black-box graph models.

Local approximation of black-box models. From a bird's eye view, to explain the prediction of an input black-box model f on a given input $\hat{x}$ our method solves the following optimization:

$$\min_{g \in \mathcal{M}} d_{\hat{x}}(f, g) \tag{3}$$

where $\mathcal{M}$ is a class of interpretable models and $d_{\hat{x}}$ is some notion of distance between functions that depends on $\hat{x}$. Intuitively, $d_{\hat{x}}(f, g)$ measures how similar f and g are *in the neighborhood* of $\hat{x}$. In this work, we use the ℓ_2 distance w.r.t. a Gaussian measure centered at $\hat{x}$:

$$d_{\hat{x}}(f, g) = \int_x (f(x) - g(x))^2 \, \mathrm{d}\mu_{\hat{x}}(x)$$

where $\mu_{\hat{x}} = \mathcal{N}(\hat{x}, \sigma^2)$ is the normal distribution with variance parameter σ^2.

When the function to be explained is a graph model f, we want to explain the prediction $f(\hat{G}, \hat{X})$ on a given input instance consisting of a graph $\hat{G}$ and its feature matrix $\hat{X} \in \mathbb{R}^{n \times d}$. We first assume that f is a real-valued function, i.e., $f(\hat{G}, \hat{X}) \in \mathbb{R}$ (we will later discuss how our method can be used for classification tasks). We propose to instantiate Problem (3) w.r.t. the neighborhood of the node features $\hat{X} \in \mathbb{R}^{n \times d}$, leading to the minimization problem

$$\min_{g \in \mathcal{M}} \int_X (f(\hat{G}, X) - g(\hat{G}, X))^2 \, \mathrm{d}\mu_{\hat{X}}(X) \tag{4}$$

where $\mu_{\hat{X}}$ is the product measure defined by

$$\mu_{\hat{X}}(X) = \prod_{i=1}^{n} \prod_{j=1}^{d} \mu_{i,j}(X_{i,j}) \tag{5}$$

with $\mu_{i,j} = \mathcal{N}(\hat{X}_{i,j}, \sigma^2)$. This equality can be conveniently rewritten as a risk minimization problem

$$\min_{g \in \mathcal{M}} \mathop{\mathbb{E}}_{\substack{\xi_{i,j} \sim \mathcal{N}(0, \sigma^2) \\ i \in [n], j \in [d]}} (f(\hat{G}, \hat{X} + \xi) - g(\hat{G}, \hat{X} + \xi))^2 \tag{6}$$

whose solution can be approximated using the empirical risk minimization principle. Concretely, we generate a dataset of input-output examples using the black-box model,

$$X^{(t)} = \hat{X} + \xi^{(t)}, \;\; Y^{(t)} = f(\hat{G}, \hat{X}^{(t)}) \quad t = 1, \cdots, N$$

where all the entries of the matrices $\xi^{(t)}$ are drawn i.i.d. from $\mathcal{N}(0, \sigma^2)$, and we solve the minimization problem

$$\min_{g \in \mathcal{M}} \frac{1}{N} \sum_{t=1}^{N} (Y^{(t)} - g(\hat{G}, X^{(t)}))^2. \tag{7}$$

The overall ILS explanation method is summarized in Algorithm 1.

Group sparse linear models as surrogates. Now that we have a tractable way to tackle Problem (3) for graph black-box models, we discuss the choice of the class of interpretable models $\mathcal{M}$. Several choices for $\mathcal{M}$ are possible depending on the nature of the desired explanations. In this work, we focus on explanations consisting of a set of nodes

that are most relevant for the prediction $f(\hat{G}, \hat{X})$, for which group sparse linear models are well suited. In this context, we define the class of group sparse linear models as linear functions of the node features that rely on only a small subset of the nodes. Formally, given a target number K of nodes to be returned as an explanation, Problem (6) becomes

$$\min_{W \in \mathbb{R}^{n \times d}} \frac{1}{N} \sum_{t=1}^{N} (Y^{(t)} - \langle W, X^{(t)} \rangle)^2 \quad \text{subject to } \|W\|_0 \leq K \tag{8}$$

where we recall that $\|W\|_0$ denotes the number of non-zero rows of the matrix W. Let $\hat{W}$ be a solution of this problem and let $i_1, i_2, \cdots, i_K \in [n]$ be the indices of the non-zero rows of $\hat{W}$. One can easily see that the function g computed by the surrogate is given by

$$g(G, X) = \sum_{j=1}^{K} \langle W_{i_j,:}, X_{i_j,:} \rangle,$$

for any node features $X \in \mathbb{R}^{n \times d}$. That is, g is a linear function of the features of the nodes in the explanation set $S = \{v_{i_1}, v_{i_2}, \cdots, v_{i_K}\} \subset V$ (and these nodes only).

Problem (8) is a generalization of the well-known sparse linear regression problem with structured sparsity (Natarajan, 1995; Yuan & Lin, 2006). While this problem is NP-hard, the Group Orthogonal Matching Pursuit (OMP) method (Swirszcz et al., 2009) can be used to efficiently find an approximate solution. The Group OMP algorithm used to fit the group sparse linear surrogate is described in Algorithm 2.

Explaining node-level, edge-level and graph-level models. As described previously, ILS explains the prediction of a black-box model taking graphs as input. While it is straightforward to use ILS to explain predictions of models trained for making predictions about entire graphs (graph-level), one may wonder how to use it for node-level and edge-level models. Let f be a black-box model trained for a node level task, i.e., $f : V \to \mathbb{R}$. Even though f takes nodes as input, the prediction $f(v)$ for a given node $v \in V$ will (almost always) depend on the graph structure and the nodes' features. For example, in the case of a GNN, $f(v)$ depends on the graph structure and the features of the nodes in the receptive field. It is thus natural to think of the prediction $f(v)$ as the output of the function $f_v : (G, X) \mapsto f(v)$ (a change in the node features or graph structure will likely result in a change in the output of a node-level model). To explain the prediction of a black-box node-level model $f(v)$ for a given node $v \in V$ with attribute matrix X, we give the function f_v as input to ILS. ILS then locally approximates f around the node features X by a group-sparse linear model to identify the node explanation set. The case of edge-level models can be handled in the exact same fashion.

Beyond real-valued black-box models. Until now, we have assumed that the output of the black-box model is a real value: $f(G, X) \in \mathbb{R}$. This raises the question of how to use ILS to explain classification models. While ILS can only take a real-valued black-box model as input, it is easy to extract relevant real-valued functions from models trained on classification tasks. For example, for a GNN trained on a binary classification task, one would consider the function $f : G, X \mapsto \mathbb{P}(Y = 1 \mid G, X)$ computed by the output layer of the GNN. For a multi-class task, to explain the class c predicted by the GNN on a given input instance to be explained, one would consider the function $f : G, X \mapsto \mathbb{P}(Y = c \mid G, X)$. Note that in both cases, one could consider using the log-probability as the output rather than the probability itself, which may be more aligned with the fact that ILS locally approximates f with a *linear* model. While the class probability is a natural choice for the function to feed into ILS, other choices could be considered, e.g., the entropy of the class distribution, the difference between the probabilities of the top 2 predicted classes, etc.

4. Experiments

We conduct experiments for explaining GNN models trained on both graph-level and node-level tasks. For the graph-level experiments, we propose a new explainability dataset; for the node-level explainability task we rely on six datasets that are widely used in other explainability literature. Code can be found here [1].

4.1. Datasets

Graph-level dataset We consider graph level tasks where the predictions of the model depend on the features of a specific motif in the graph. Our main goal in this setting is to evaluate our explainability method on a task where the GNN finds both the important structure and a ground truth function that maps the node features to the graph label. We thus propose a new synthetic graph-level dataset called *BA-GraphR* in which every sample is a Barabasi-Albert random graph (BA) (Barabási & Albert, 1999) with a motif attached to one of the nodes. Each motif is a subgraph of 5 nodes, arranged in the shape of either a house or a wheel. For regression, BA-GraphR consists of 1000 graphs of 40 nodes each. The node features are sampled independently at random from a standard normal distribution. The regression label of each graph is equal to the sum of the node features of the nodes in the motif, $y = \sum_{v_i \in M} \sum_{j=1}^{d} [x_{v_i}]_j$. The label of the graphs in this dataset depends on the node features in the motif. We use the same dataset structure for the graph classification task, but with 500 graphs with a house motif and 500 graphs with a wheel motif. The task is a binary

[1] https://github.com/farzana0/GNNEXP_ILS

Algorithm 1 ILS: Explainability with Interpretable Local Surrogates

Input: f (black-box model), $G = (V, E)$, $X \in \mathbb{R}^{n \times d}$ (instance to be explained), N_s (number of samples), $\mathcal{M}$ (class of interpretable models), σ^2 (variance parameter)
Output: $g \in \mathcal{M}$ (an interpretable model explaining $f(G, X)$)

for $t = 1, \cdots N_s$ **do**
  $\xi_{i,j}^{(t)} \sim \mathcal{N}(0, \sigma^2)$ for $i = 1, \cdots, n, j = 1, \cdots, d$
  $X^{(t)} = \hat{X} + \xi^{(t)}$
  $Y^{(t)} = f(\hat{G}, \hat{X}^{(t)})$
end for
Solve $\arg\min_{g \in \mathcal{M}} \frac{1}{N} \sum_{t=1}^{N} (Y^{(t)} - g(\hat{G}, X^{(t)}))^2$ *// E.g. using GOMP (Algorithm 2)*
Return g

Algorithm 2 GOMP: Group Sparse Linear Surrogate with Orthogonal Matching Pursuit (Swirszcz et al., 2009)

Input: $\{(X^{(t)}, Y^{(t)}\}_{t=1}^{N} \subset \mathbb{R}^{n \times d} \times \mathbb{R}$ (training data), K (number of nodes in the explanation set / non-zero groups)
Output: $W \in \mathbb{R}^{n \times d}$ an approx. solution to $\arg\min_{W \in \mathbb{R}^{n \times d}} \frac{1}{N} \sum_{t=1}^{N} (Y^{(t)} - \langle W, X^{(t)} \rangle)^2$ subject to $\|W\|_0 \leq K$, $S = \{i_1, \cdots, i_K\}$ the indices of the non-zero groups (corresponding to the nodes in the explanation set)

Let the 3rd order tensor $\mathcal{X} \in \mathbb{R}^{N \times n \times d}$ be defined by $\mathcal{X}_{t,:,:} = X^{(t)}$ for $t = 1, \cdots N$
Let $y \in \mathbb{R}^N$ be defined by $y_t = Y^{(y)}$ for $t = 1, \cdots, N$
// Data whitening
for $u = 1, \cdots, n$ **do**
  Perform SVD: $\mathcal{X}_{:,u,:} = UDV^\top$
  $\mathcal{X}_{:,u,:} \leftarrow \mathcal{X}_{:,u,:} V^\top D^{-1}$
end for
// Group Orthogonal Matching Pursuit
$S \leftarrow \emptyset$, $W \leftarrow 0 \in \mathbb{R}^{n \times d}$
for $k = 1, \cdots, K$ **do**
  $i_k \leftarrow \arg\max_{j=1,\cdots,n} \|\mathcal{X}_{:,j,:}^\top (\mathcal{X}_{(1)} \text{vec}(W) - y)\|_2$
  $S \leftarrow S \cup \{i_k\}$
  $W_{S,:} \leftarrow \arg\min_{M \in \mathbb{R}^{k \times d}} \|(\mathcal{X}_{:,S,:})_{(1)} \text{vec}(M) - y\|_2$
end for
Return W, S

classification of the house and wheel graphs.

Node-level dataset We use 6 synthetic dataset, widely used in other explainability works (Pereira et al., 2023; Ying et al., 2019). All these datasets are set up for node classification, which we briefly describe below. More details about the datasets can be found in Luo et al. (2020).

- BA-House: a random BA graph with 700 nodes where 80 house-shaped motifs of size 5 have been attached with one edge to a randomly selected nodes.
- BA-Community: two BA-house graphs connected by an edge between two randomly selected nodes.
- BA-Grids: a random BA graph with 1020 nodes where 80 grid-shaped motifs of size 9 have been attached to randomly selected nodes.
- Tree-Cycles: a random tree graph with 991 nodes where 80 cycle-shaped motifs each of size 6 have been attached to randomly selected nodes.
- Tree-Grids: a random tree graph with 1231 nodes where 80 grid-shaped motifs of size 9 have been attached to randomly selected nodes.
- BA-Bottle: a random BA graph with 700 nodes where 80 bottles-shaped motifs of size 5 have been attached to randomly selected nodes.

In all above datasets, the task is a multi-class node classification, where the nodes in the random BA graph have the same label and the nodes in the motif have a label that depends on their position in the motif. The node features used in all of theses datasets is a vector of all-ones.

4.2. Baselines and Metrics

Naive Gradient Baseline We consider a simple baseline identifying the most relevant nodes by sorting them according to the magnitude of the corresponding directional deriva-

Table 1. Graph-level Explainer Results

Method	**Classification**	**Regression**
Gradient	49.08	95.30
Gradient H1	30.69	87.11
Gradient H2	**64.20**	83.73
ILS	44.29	**96.61**
ILS H1	54.28	89.47
ILS H2	58.78	80.95
PGExplainer	30.01	85.67

Table 2. Node-level Explainer Results

Model	BA-House	BA-Community	BA-Grids	Tree-Cycles	Tree-Grids	BA-Bottle
ILS (ours)	98.7± NA	**95.5± NA**	**98.3± NA**	91.5± NA	**90.0± NA**	99.7± NA
Gradient	99.5± NA	92.1 ± NA	97.9± NA	92.2± NA	89.8± NA	**99.8± NA**
GNNExplainer	77.5 ± 1.2	64.7 ± 1.0	89.2 ± 2.0	77.2 ± 9.0	71.1 ± 1.0	73.3 ± 3.0
PGExplainer	95.0 ± 1.1	70.6 ± 2.0	86.2 ± 9.0	92.4 ± 5.2	76.7 ± 1.2	98.2 ± 3.0
PGM-Explainer	97.9 ± 0.9	92.2 ± 0.2	88.6 ± 0.9	**94.1 ± 0.8**	86.8 ± 2.0	97.5 ± 1.5
DnX	97.7 ± 0.2	94.6 ± 0.1	89.8 ± 0.1	83.3 ± 0.4	80.2 ± 0.1	99.6 ± 0.1
FastDnX	**99.6 ± NA**	95.4 ± NA	93.9 ± NA	87.3 ± NA	85.0 ± NA	**99.8 ± NA**

tives of the GNN function $f : \mathbb{R}^{n\times d} \to \mathbb{R}$ (which we treat as a function of the node feature matrix). More precisely, we approximate the directional gradient in the direction of each node feature vectors using finite differences:

$$s_u = \frac{|f(X + \varepsilon M^{(u)}) - f(X)|}{\varepsilon} \quad \text{for } u = 1, \cdots, n$$

with a small ε and where $M^{(u)} \in \mathbb{R}^{n\times d}$ is the matrix full of 0 except for its uth row which is filled with ones. This baseline is similar to the method implemented in (Faber et al., 2021) as node gradient, except that we estimate the directional gradient with the simple approximation above without requiring to open the black-box model.

Other baselines We compare our method against DnX and FastDnX (Pereira et al., 2023), GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), PGM-Explainer (Vu & Thai, 2020) and the naive gradient method. All baseline results in Table 2 (except for the gradient method) are taken directly from (Pereira et al., 2023). In the graph-level experiments, we used built-in function of pytorch geometric explain (Fey & Lenssen, 2019) for PG-Explainer with parameters as $\#epochs = 60$, $lr = 0.03$ for regression and $lr = 0.00001$ for the classification task and the algorithm returns top 20 edges.

Evaluation metrics We evaluate our method on explanation accuracy: the number of nodes predicted by the explanation method that are also in the ground-truth explanation, divided by the total number of nodes in the ground-truth explanation. In our synthetic datasets, the ground-truth number of nodes is equal to the number of nodes in the motif.

4.3. Experiment Results

Graph-level experiments We train a GIN model (Xu et al., 2019) on a BA-GraphR dataset consisting of 1000 graphs with 40 nodes each for the graph classification and regression tasks. We run ILS on each of the 1000 graphs to explain the prediction of the GIN model and report the average explanation accuracy in Table 1. In this table ILS is compared with the gradient and PGExplainer baselines. Since PGExplainer outputs edge explanation sets rather than node explanation sets, we convert the edge-level explanations to node level ones similarly to what was done in (Pereira et al., 2023). We also use ILS on the function computed by the GNN considering the hidden layers as inputs, which we denote with H1 and H2 for the gradient baseline and ILS (for the 1st and 2nd hidden layers, respectively).

On the classification task, we observe that both the gradient baseline and ILS outperform PGExplainer by a large margin. Both methods perform better when they have access to the internal hidden layer embedding. This latter point is expected, as the hidden layers of the GNN likely encode the structural information needed to perform the classification task, which consists of identifying a motif in the graph. On the regression task, both methods significantly outperform

PGExplainer again. In this case, both methods perform better when the function to explain is the one taking the node features as input, rather than the hidden layers' embeddings. This is expected since, in this task, the output is a linear function of the features of the nodes in the motif.

Node-level experiments In these experiments, we use the trained GCN models used in previous work (Pereira et al., 2023; Luo et al., 2020) as the black-box model. For each graph, we run ILS to explain the prediction of the trained GCN on each of the motif nodes and report the average explanation accuracy in Table 2. Despite its simplicity, our method is the best-performing method on all the datasets except for Tree-Cycles and BA-House. Similar to (Faber et al., 2021), we observe that the node gradient methods have a reasonable performance for most of the datasets, while being much faster than other approaches. To speed up ILS, we limit the input to the nodes in the receptive field of the GCN.

5. Discussion

Comparison with DnX (Pereira et al., 2023) The DnX method consists of two steps: 1) distilling the black-box model in a linear surrogate, a *simple graph convolution network* (SGC) (Wu et al., 2019), on the whole dataset, and 2) find a sparse approximation of the surrogate to identify nodes in the explanation set. The main difference between DnX and ILS stems from the notions of local versus global approximation. DnX requires the model to be well approximated *globally*, making it not well suited for tasks where a simple GNN like SGC performs poorly, such as graph level tasks (Wu et al., 2019; Huang et al., 2020). In contrast, ILS only assumes that a sparse linear model can approximate the black-box model *locally* in the neighborhood of an input instance, making the linear nature of the surrogate less restrictive. The global nature of the distillation step of DnX also implies that to explain the prediction of a GNN on one node, one first needs to distill the GNN into an SGC on the whole set of nodes of the graph, making the time complexity of the explanation process dependent on the number of nodes in the graph. In contrast, since ILS only performs a local approximation, its time complexity only depends on the size of the GNN's receptive field.

Comparison with GraphLIME (Huang et al., 2022) GraphLime is based on the same assumption as ILS, that in the local neighbourhood of an instance, a sparse model can approximate the black-box model. However, GraphLIME provides explanations at the feature level (the explanation consists of a set of features rather than nodes) and the model used for the approximation is non-linear. Moreover, GraphLIME can only handle graph-level tasks while ILS can handle both node-level and graph-level tasks.

Graph structure and surrogates While the group sparse linear surrogate does not take the graph structure into account (its input is the feature matrix only), it can still identify the most important nodes even in datasets where the task does not directly depend on the node features. This is because the GNN function, which was used to generate the datasets to fit the group sparse model, depends on the graph structure. Moreover, ILS can explain models with both attributed and non attributed graphs. In the experiments where the label does not depend on the features and the features are all ones, because of the sensitivity of the GNN function to the features of some of the nodes (important nodes), adding noise to these arbitrary features affects the GNN's output. In graph classification tasks where the output does not depend on the features and features are uniformly drawn from normal distribution, we can instead use node embeddings (at different layers of the GNN) as the input to the surrogate model (assuming the model being explained is not black-box, but a given GNN).

Unknown node explanation set size Lastly, we point out that in the case where the number of relevant nodes is not known in advance, one could use the Group Lasso algorithm (Natarajan, 1995) to fit the surrogate. The Group Lasso algorithm solves a convex relaxation of the group sparse linear problem, where the amount of sparsity is not enforced as a hard constraint (as in the Group-OMP algorithm), but encouraged smoothly through a regularization term whose weight is controlled by a hyper-parmater λ. One could use a validation set (generated in the same fashion as for the surrogate training dataset, using the black-box model) to select the best value of λ. This would result in ILS being able to adaptively choose the size of the explanation set.

6. Conclusion

We have introduced an instance-level graph explainability method, referred to as ILS, which involves fitting interpretable local surrogates in the neighborhood of a specific example. By leveraging the interpretability of these simple surrogates, ILS provides valuable insights into the behavior of graph models within the local neighbourhood of an instance. Notably, ILS is capable of accommodating black-box graph models and efficiently identifies the most relevant nodes for a given prediction through a group sparse optimization approach. Through empirical evaluations, we have demonstrated the effectiveness of our method and compared it to state-of-the-art explainability methods.

Our future work involves examining the feature selection capability of ILS and exploring different interpretable surrogate functions. In particular, we believe using a similar group sparse surrogate, but enforcing group sparsity at the

feature level rather than at the node level, is a promising direction to apply ILS when the interest is in feature level explanations. Additionally, we plan to evaluate our method on a broader range of datasets to further validate its performance and generalizability. Lastly, we plan on studying the theoretical properties of ILS. More specifically, we believe that ILS offers strong faithfulness and stability guarantees (Agarwal et al., 2022), which we plan to demonstrate formally.

Acknowledgement

This research is supported by the NSERC CREATE InterMath-AI (IMA) and CIFAR AI chair funding.

References

Agarwal, C., Zitnik, M., and Lakkaraju, H. Probing gnn explainers: A rigorous theoretical and empirical analysis of gnn explanation methods. In *International Conference on Artificial Intelligence and Statistics*, pp. 8969–8996. PMLR, 2022.

Baldassarre, F. and Azizpour, H. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686*, 2019.

Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Cui, Z., Li, X., Gao, S., and Wei, X.-Y. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2019.

Faber, L., Moghaddam, A. K., and Wattenhofer, R. When comparing to ground truth is wrong: On evaluating gnn explanation methods. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021.

Fan, S., Wang, X., Mo, Y., Shi, C., and Tang, J. Debiasing graph neural networks via learning disentangled causal substructure. In *NeurIPS*, 2022.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric. https://github.com/pyg-team/pytorch_geometric, 2019. Version 1.2.0.

Finzel, B., Biehl, M., and Hammer, B. Generating explanations for conceptual validation of graph neural networks. *KI-Künstliche Intelligenz*, pp. 1–15, 2022.

Henderson, R., Clevert, D.-A., and Montanari, F. Improving molecular graph neural network explainability with orthonormalization and induced sparsity. In *International Conference on Machine Learning*. PMLR, 2021.

Huang, Q., He, H., Singh, A., Lim, S.-N., and Benson, A. R. Combining label propagation and simple models out-performs graph neural networks. *arXiv preprint arXiv:2010.13993*, 2020.

Huang, Q., Zhang, Q., Wu, X.-M., and Zhu, W. Graphlime: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 2022. doi: 10.1109/TKDE.2022.3149339.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Liu, Z., Wu, F., Wang, S., Wang, Z., Wu, J., Cui, W., and Qu, H. Visualizing graph neural networks with corgie: Corresponding a graph to its embedding. *IEEE Transactions on Visualization and Computer Graphics*, 28(6): 2500–2516, 2022.

Lucic, A., Doe, J., and Johnson, S. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

Luo, D., Ding, K., Xu, Y., Li, X., Zhi, Y., and Zhang, Q. Parameterized explainer for graph neural network. In *Advances in Neural Information Processing Systems 33*, pp. 19620–19631, 2020.

Natarajan, B. K. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

Pereira, T., Smith, J., and Johnson, S. Distill n'explain: Explaining graph neural networks using simple surrogates. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023.

Sanchez-Lengeling, B., Outeiral, C., Barros, R., and Aspuru-Guzik, A. Evaluating attribution for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5898–5910, 2020.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

Swirszcz, G., Abe, N., and Lozano, A. C. Grouped orthogonal matching pursuit for variable selection and prediction.

In *Advances in Neural Information Processing Systems*, volume 22, 2009.

Vu, M. and Thai, M. T. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. In *Advances in Neural Information Processing Systems 33*, pp. 12225–12235, 2020.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871. PMLR, 2019.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. GNNExplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems 32*, 2019.

Yuan, H., Xu, X., Yan, Y., Huang, Z., and Ji, S. Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (1):49–67, 2006.

Zhu, Z., Yang, X., Zeng, X., Zhang, M., and Huang, Y. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022a.

Zhu, Z., Zhang, L., Wang, H., Jin, X., Huang, H., and Yu, P. S. Neural-symbolic models for logical queries on knowledge graphs. In *International Conference on Machine Learning*. PMLR, 2022b.