HGExplainer: explainable heterogeneous graph neural network

Grzegorz P. Mika, Amel Bouzeghoub, Katarzyna Wegrzyn-Wolska, Yessin M Neggaz

▶ **To cite this version:**

Grzegorz P. Mika, Amel Bouzeghoub, Katarzyna Wegrzyn-Wolska, Yessin M Neggaz. HGExplainer: explainable heterogeneous graph neural network. 2023. hal-04220962v1

HAL Id: hal-04220962

https://hal.science/hal-04220962v1

Preprint submitted on 28 Sep 2023 (v1), last revised 15 Nov 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HGExplainer: Explainable Heterogeneous Graph Neural Network

Grzegorz P. Mika*†, Amel Bouzeghoub*, Katarzyna Węgrzyn-Wolska† and Yessin M. Neggaz†
*SAMOVAR, Telecom SudParis, Institut Polytechnique de Paris, Palaiseau, France
†Efrei Research Lab, EFREI Paris, Pantheon Assas University, Villejuif, France
grzegorz.mika@ip-paris.fr, amel.bouzeghoub@telecom-sudparis.eu, {katarzyna.wegrzyn, yessin.neggaz}@efrei.fr

***Abstract*—Graph Neural Networks (GNNs) are an effective framework for graph representation learning in real-world applications. However, despite their increasing success, they remain notoriously challenging to interpret, and their predictions are hard to explain. Nowadays, several recent works have proposed methods to explain the decisions made by GNNs. However, they only aggregate information from the same type of neighbors or indiscriminately treat homogeneous and heterogeneous neighbors similarly. Based on these observations, we propose HGExplainer, an explainer for heterogeneous GNNs to comprehensively capture structural, semantic, and attribute information from homogeneous and heterogeneous neighbors. We first train the GNN model to represent the predictions on a heterogeneous network. To make the explainable predictions, we design the model to capture heterogeneity information in calculating the joint mutual information maximization, extracting the meta-path-based graph sampling to generate more prosperous and more accurate explanations. Finally, we evaluate our explainable method on synthetic and real-life datasets and perform concrete case studies. Extensive results show that HGExplainer can provide inherent explanations while achieving high accuracy.**

***Index Terms*—Explainable Artificial Intelligence (XAI), Graph Neural Networks (GNNs), Heterogeneous Networks, Recommender Systems, Trustworthy**

I. Introduction

Graph Neural Networks (GNNs) have become increasingly popular for learning representations of graph-structured data in real-world applications, such as Social Networks [**?**], Recommender Systems (RSs) [**?**], [**?**], [**?**], Molecules [**?**], [**?**] and Citation Networks [**?**]. GNNs broadly employ a message-passing scheme with features aggregated from its neighbors to learn node representations [**?**], [**?**]. This scheme enables the GNN to capture node features, neighborhood information, and local graph topology. GNN-based methods have achieved state-of-the-art performance in node classification [**?**], graph classification [**?**], link classification [**?**], and link prediction [**?**]. Despite the remarkable empirical effectiveness of machine learning on graphs, explaining predictions made by GNNs remains a challenging open problem. Although GNNs make valuable predictions, due to the strong non-linearity of the model, they act as black boxes, and proving that the model has made the intended use of the graph structure is complex. Furthermore, the lack of interpretability makes the GNNs untrustworthy, which prevents their application in safety-critical areas. Therefore, significant subgraphs and a set of features, also known as explanations, must be uncovered. The literature has shown that this is still a new research area that requires further investigation to understand the complexities involved in the explainability of graph-based deep learning models. Despite extensive research efforts on explainable techniques for deep models on images and texts [**?**], [**?**], [**?**], most of them cannot explain graph-based deep learning models directly.

In contrast, graphs are non-Euclidean objects, meaning no locality information exists. Each node has a different number of neighbors and contains crucial structural information, creating a sparse adjacency matrix with other nodes of the same graph. Thus, explainable methods for images and texts cannot be applied directly; still, the related problems of neural debugging have received substantial attention in deep learning. Recent methods have been proposed to explain the predictions of GNNs as reported in some surveys [**?**], [**?**], [**?**]. They provide different views to understand the graph models. Two main classes are distinguished: instance-level methods that provide input-dependent explanations by identifying important features for the prediction and model-level methods that explain general behavior and provide input-independent explanations. GNNExplainer [**?**], XGNN [**?**], PGExplainer [**?**], Grad [**?**], Grad-CAM [**?**] and SubgraphX [**?**] are examples of instance-level methods. XGNN is the only model-level method. However, despite the emergence of new methods, GNNExplainer is still the leading method to explain GNNs using a mutual-information approach.

For applications such as graph-based RSs, the input data is typically heterogeneous graphs containing various types of nodes and edges, and the recommendation problem is modeled as a link prediction task [**?**], [**?**], [**?**]. However, GNNExplainer is only designed for homogeneous graphs and classification tasks. Since GNNs emerged, RSs have become even more complex, and recommendations are more challenging to explain. *How to explain link predictions on heterogeneous graphs containing rich semantic information and easily generalize the learned explainer model* remains largely unexplored in the literature [**?**]. To shed some light on this problem, we propose HGExplainer to overcome some of GNNExplainer's limitations in this paper. Indeed, although GNNExplainer is currently the leading method to explain GNN models, there is still room for improvement.

In that respect, our proposal, HGExplainer, provides explanations of GNNs for the link prediction task in homogeneous and heterogeneous graphs. Extensive experiments on several

real-world datasets show that HGExplainer provides built-in interpretability while achieving comparable performance with the non-interpretable counterparts. In summary, the main contributions of this paper are as follows:

- We propose HGExplainer, an explainable method adapted to heterogeneous graphs that improves the transparency of the link prediction task, including RS applications. More specifically, our model first constructs a meta-path-based graph and then maximizes the joint mutual information, paying attention to the richer heterogeneity information.
- To evaluate our work, we conduct extensive experiments on two random and two real heterogeneous graph datasets. The results demonstrate promising performances of our model over baseline methods and the ability to produce semantic-aware interpretable results.
- To the best of our knowledge, this is the first model that explains heterogeneous GNNs and is being evaluated on heterogeneous graph datasets.

The rest of the paper is organized as follows. Section II discusses the literature review. Section III presents our contribution, namely, HGExplainer. In Section IV, we present our extensive experiments and discuss our results. Finally, Section V concludes our work.

II. Related Work

In this section, we first review the recent development of explainability in GNN and provide a critical analysis; then, we focus on the leading solution, GNNExplainer, and discuss its benefits and weaknesses.

A. Explanations in GNNs

As the number of GNN applications grows, understanding why GNNs make such predictions becomes increasingly critical. To explain GNN algorithms, several methods have been proposed [**?**], [**?**], [**?**]. Based on how the importance scores are obtained, we divided them into different classes: gradients/features-based methods, decomposition-based methods, surrogate methods, perturbation-based methods, and model-level methods. The gradients/features-based methods are the most straightforward approaches in a wide range of explained image and text predictions. Due to their simplicity, they can explain GNN models. These methods compute the gradients of targeted prediction concerning input features by back-propagation, and the key difference lies in how different hidden feature maps are combined. Authors in [**?**] extended explainability methods designed for CNNs to GCNNs. According to their experiments, Grad-CAM [**?**] is the most suitable among the studied methods for explanations on graphs of moderate size. The decomposition-based methods include GNN-LRP [**?**] and Excitation BP [**?**]. They build score decomposition rules to distribute the prediction scores to the input space. The surrogate methods, such as GraphLIME [**?**] and PGM-Explainer [**?**], work through the generation of a simpler surrogate model based on the relationships in the neighboring areas of the input example. Furthermore, several perturbation-based methods have been proposed, including GNNExplainer [**?**], PGExplainer [**?**], GraphMask [**?**], and SubgraphX [**?**]. Finally, the unique existing model-level method is XGNN [**?**], based on graph generation for graph classification only, providing high-level insights and a generic understanding of how GNNs work. However, all the surveyed methods explain only *Node* or *Graph Classification tasks* in the context of *homogeneous graphs*. According to recent surveys [**?**], [**?**], the perturbation-based methods differ from concurrent ones by an efficient graph perturbation and the exploitation of structural information, achieving the most promising results. On the other hand, gradient-based methods have several significant limitations, such as heuristic assumptions. Moreover, these methods have special requirements for the GNN structure, limiting their application and generalization. A significant limitation of decomposition-based methods is that they can only study the importance of different nodes, making it impossible to analyze the graph structure as a crucial heterogeneous property. In addition, they require a comprehensive understanding of the model structure, thereby limiting the scope of their usage. Finally, surrogate methods are challenging to apply to GNN models since graphs are discrete and contain topology information, which makes it challenging to define neighboring regions in graphs.

B. GNNExplainer

Although the study is still ongoing to develop an explainable GNN method, among several perturbation-based methods, we use GNNExplainer, which achieves noticeably better results in terms of efficiency and complexity. It is a model-agnostic method for learning soft masks for node features and edges to explain the predictions made by GNNs. Given a graph with node features and a trained GNN model, the method randomly initializes soft masks and treats them as trainable variables. Then, they are combined with the original graph via element-wise multiplications. Then, GNNExplainer runs a mask optimization algorithm that finds a selection of edges that maximize the model output, providing explanations for any node. Specifically, GNNExplainer extracts the subgraph and the associated subset of node features that are important for prediction. GNNExplainer explains GNN models based on the input graph perturbation and exploits the structural information from GNNs. However, the mask optimization runs for each input graph individually, which lacks a global understanding of predictions. Besides, the method only provides homogeneous explanations for node and graph classification. Even though the authors of GNNExplainer have vaguely mentioned its possible use with heterogeneous graphs in the link prediction task, it is only at a theoretical level without any experimental justification. As a result, we propose an extension of GNNExplainer adapted to the link prediction task on heterogeneous graphs, proving that the original method cannot work directly to explain more complex applications such as RSs and explaining the link prediction task on heterogeneous graphs is obviously not straightforward and not yet proven experimentally.

TABLE I
NOTATIONS USED IN THIS PAPER.

Symbols	Definitions
$\mathcal{G}$	A heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R}, \mathcal{T})$
$\mathcal{V}$	The set of nodes in a graph with different types
$\mathcal{E}$	The set of edges in a graph with different types
$\mathcal{R}$	The set of relation types
$\mathcal{T}$	The set of node types
v	A node $v \in \mathcal{V}$ in a graph
e	An edge $e \in \mathcal{E}$ in a graph
r	A edge type in a graph
t	A node type in a graph
M_{path}	The set of meta-path M_{path} in a graph
$\mathcal{G}_s$	The set of meta-path-based graph of $\mathcal{G}$
$\circ$	The composition operator on relations
$\lvert \cdot \rvert$	The cardinality of a set
Y, U, C	The set of discrete random variables
$P(Y), P(U)$	The probability distribution
$H(Y), H(Y, U)$	Entropy of a discrete random variables
$H(U\lvert Y)$	The conditional entropy of the variable Y given U
$\mathcal{I}(Y; U)$	Mutual information of two random variables Y and U
$\mathcal{X}_s$	The associated feature information of $\mathcal{G}_s$
F	The feature mask used to select features
$\mathcal{D}$	An indicator of the links probability belonging to R
$\mathcal{M}$	The result of the $max\mathcal{I}$ for each $\mathcal{T}$
q	An indicator for positive and negative triples
Φ	A heterogeneous GNN model
$\hat{d}$	A link prediction of the node pair(v_1, v_2)
$\mathcal{G}_c$	A computation graph, i.e., L-hop ego-graph of(v_1, v_2)

C. Heterogeneity Challenges

Different from homogeneous graphs, where the fundamental problem is preserving structure and property in node embedding [**?**], heterogeneous graph embedding imposes more challenges, including complex structure and attributes as well as complex applications. As nodes and edges in a homogeneous graph have the same type, each dimension of the node's or edge's attributes has the same meaning. It means that a node can directly merge the attributes of its neighbors. However, in heterogeneous graphs, the attributes of different types of nodes and edges may have different meanings [**?**]. For example, the attributes of node *author* can be the *research fields*, while the node *paper* may use *keywords* as attributes. Therefore, *how to overcome the heterogeneity of attributes and effectively fuse the attributes of neighbors* poses another challenge in heterogeneous graph embedding.

III. METHODOLOGY

In this section, we present in detail our proposed explainable method, called HGExplainer, which can be applied in various heterogeneous GNN models, including RS applications. Fig. 2 illustrates the overall framework, which falls into two parts: (1) subgraph sampling and (2) heterogeneous explainer. We first introduce the notations and problem formulation, then discuss the proposed framework's details. The workflow is that we first use heterogeneous GNN incorporating node types to learn the graph representation for the link prediction of whether the user likes the item or not. We then generate a meta-path-based graph and maximize the mutual information to calculate the most important nodes and edges for the prediction. Besides, the mathematical notations used in this paper are summarized in Tab. I.

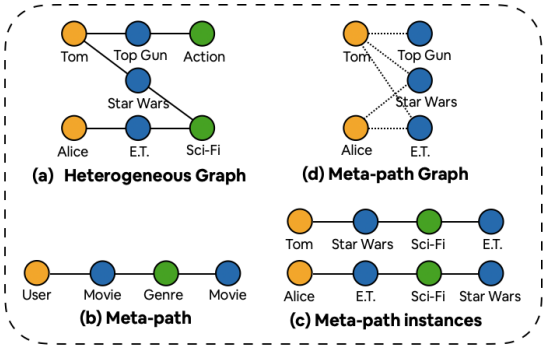

Fig. 1. An illustration of the terms defined in Section III (a) An example heterogeneous graph with three types of nodes (i.e., user, movie, and genre).(b) The User-Movie-Genre-Movie (U-M-G-M) metapath. (c) Example metapath instances of the U-M-G-M metapath. (d) The metapath-based graph for the U-M-G-M metapath.

Heterogeneous Graph definition. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R}, \mathcal{T})$ be a heterogeneous graph, where $\mathcal{V}$ denotes a set of nodes with different types, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes a set of edges with different types, $\mathcal{R} = \{r_1, r_2, ..., r_b\}$ is the set of $|\mathcal{R}| > 1$ relation types, and $\mathcal{T} = \{t_1, t_2, ..., t_i\}$ is the set of $|\mathcal{T}| > 1$ node types. Each node $v \in \mathcal{V}$ and each edge $e \in \mathcal{E}$ are associated with node type mapping function $\psi : \mathcal{V} \rightarrow \mathcal{T}$ and edge type mapping function $\phi : \mathcal{E} \rightarrow \mathcal{R}$.

Meta-path-based Graph definition. A meta-path M_{path} is defined as a path in the form of $v_1 \xrightarrow{r_1} v_2 \xrightarrow{r_2} \cdots \xrightarrow{r_b} v_{l+1}$ which describes a composite relation $r_1 \circ r_2 \circ \cdots \circ r_b$ between nodes v_1 and v_{l+1} where $\circ$ denotes the composition operator on relations. The meta-path-based neighbor is defined as the set of nodes connecting to a node v via meta-path M_{path}. For example, considering a heterogeneous graph with three types of nodes (i.e., users (U), movies (M), and genres (G)) and the meta-paths $U-M-U$, $U-M-G$ that represent two different semantics, $U - M - U$ means that two users rate the same movie and $U - M - G$ means that a user rates a movie that belongs to a genre. Considering the meta-path $U-M-G-M$ (Fig. 1), $Tom \xrightarrow{rates} StarWars \xrightarrow{hasGenre} Sci\text{-}fi \xleftarrow{hasGenre} TopGun$ represents its instance. Based on this information, $TopGun$ is a meta-path-based neighbor of Tom. Then, the meta-path-based graph $\mathcal{G}_s$ is a graph constructed by all meta-path neighbor pairs based on meta-paths M_{path} in a graph $\mathcal{G}$.

Entropy and Mutual Information definition. Let Y be a discrete random variable taking on values in a set $Y = (y_1, y_2, \cdots, y_n)$, defined by a probability distribution $P(Y)$. Then, the entropy of a discrete random variable is denoted by

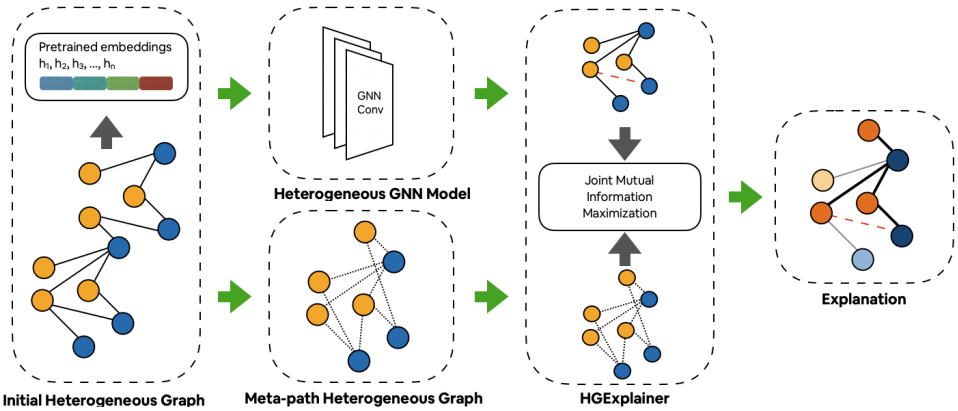

Fig. 2. Illustration of HGExplainer for explaining heterogeneous GNNs on link prediction task. The solution mainly consists of four steps: (1) train a link prediction model, (2) generate a meta-path-based graph, (3) calculate the joint mutual information maximization, (4) visualize the explanations based on a threshold.

$H(Y)$, where y_i refers to the possible values that Y can take. $H(Y)$ is defined as follows:

$$H(Y) = -\sum_{i=1}^{n} P(y_i) log(P(y_i)) \tag{1}$$

Entropy is a measure of the uncertainty of a random variable. For any two discrete random variables Y and $U = (u_1, u_2, \cdots, u_m)$, we denote $P(Y)$ and $P(U)$ as two different random variables. The joint conditional entropy of a pair of discrete random variables with a joint distribution and a conditional distribution are defined as:

$$H(Y,U) = -\sum_{j=1}^{m}\sum_{i=1}^{n} P(y_i,u_j) log(P(y_i,u_j)) \tag{2}$$

$$H(U|Y) = -\sum_{j=1}^{m}\sum_{i=1}^{n} P(y_i,u_j) log(P(u_j|y_i)) \tag{3}$$

The conditional entropy is the uncertainty left in U when a variable Y is introduced, so it is less than or equal to the entropy of both variables. The conditional entropy is equal to the entropy if, and only if, the two variables are independent. The relation between joint entropy and conditional entropy is:

$$H(Y,U) = H(Y) + H(U|Y) \tag{4}$$

$$H(Y,U) = H(U) + H(Y|U) \tag{5}$$

Mutual information is the amount of information that Y and U share. In particular, it measures how much information is communicated, on average, in one random variable about another. It is defined as:

$$\mathcal{I}(Y;U) = \sum_{j=1}^{m}\sum_{i=1}^{n} P(y_i,u_j) log\left(\frac{P(y_i,u_j)}{P(y_i)P(u_j)}\right) \tag{6}$$

$\mathcal{I}$ can be expressed as the amount of information provided by variable Y, which reduces the uncertainty of variable U. Then, the joint mutual information is defined as:

$$\mathcal{I}(Y;U|C) = H(Y|U) - H(Y|U,C) \tag{7}$$

$$\mathcal{I}(Y,C;U) = \mathcal{I}(Y;U|C) + \mathcal{I}(C;U) \tag{8}$$

where C is a discrete variable in a set $C = (c_1, c_2, \cdots, c_k)$. Interaction information can be defined as the amount of information shared by all features, but is not found within any feature subset.

A. Problem Formulation

Our key insight is the observation that the heterogeneous computation graph $\mathcal{G}_c$ of a pair of nodes v_i and v_j, which is defined by the GNN's neighborhood-based aggregation, fully determines all the information the GNN uses to generate prediction $\hat{d}$ at link $v_i \rightarrow v_j$. A heterogeneous GNN's prediction is given by $\hat{d} = \Phi(\mathcal{G}_c(v_i,v_j); \mathcal{X}_c(v_i,v_j))$ meaning that it is fully determined by the model Φ, graph structural information $\mathcal{G}_c(v_i,v_j)$, node feature information $\mathcal{X}_c(v_i,v_j)$. Formally, HGExplainer generates explanation for prediction $\hat{d}$ as $(\mathcal{G}_s, \mathcal{X}_s^F)$, where $\mathcal{G}_s$ is a meta-path-based graph of the computation graph $\mathcal{G}_c$. Furthermore, $\mathcal{X}_s$ is the associated feature of $\mathcal{G}_s$, and $\mathcal{X}_s^F$ is a subset of node features, where F is a feature mask used to select features that are important to preserve original prediction $\hat{d}$.

An overview of the proposed model architecture is shown in Fig. 2. Given the input graph $\mathcal{G}$, a pair of nodes (v_i, v_j), a trained heterogeneous GNN model Φ, and node features $\mathcal{X}_s$, our goal is to identify the most important nodes and edges for the link prediction $\hat{d}$. After training, a model with a selected edge to explain is passed to the HGExplainer. By creating a proper meta-path-based graph $\mathcal{G}_s$ with a pair of nodes (v_i, v_j), negative subgraph $\hat{\mathcal{G}}_s$, and their node features $\mathcal{X}_s$ is generated as input to the HGExplainer training. Then, entropy is calculated for each class of nodes separately in the joint mutual information maximization process. Finally, it yields common node and edge masks based on the approach presented in the next subsection.

B. Mutual Information for Heterogeneity Information

After training the GNN model, , we use the trained parameters to learn which nodes and edges on the heterogeneous

graph are important. We note that if removing the edge decreases the performance of the GNN model, this edge should be important and can be used for the explanation. Similar to [**?**], our objective is to maximize the mutual information between the GNN's prediction $\hat{d}$ and the distribution of meta-path-based graph $\mathcal{G}_s$ structure. Given a pair of nodes (v_i, v_j), the goal is to identify a meta-path-based graph $\mathcal{G}_s \subseteq \mathcal{G}_c$ and the associated features $\mathcal{X}_s = \{x_{i,j}|(v_i, v_j) \in \mathcal{G}_s(v_i, v_j)\}$ that are important for the GNN's prediction. GNNExplainer is an optimization framework using the following mutual information formula:

$$\begin{aligned} &\max_{\mathcal{G}_s} \mathcal{I}(\mathcal{D}, (\mathcal{G}_s, \mathcal{X}_s)) = \\ &H(\mathcal{D}) - H(\mathcal{D}|\mathcal{G} = \mathcal{G}_s, \mathcal{X} = \mathcal{X}_s) \end{aligned} \tag{9}$$

where D indicates the probability of links belonging to each of the relation type R. In the framework of GNNExplainer [**?**], the difficulty of using the model on heterogeneous graphs is conditioned by the lack of capturing semantic information and different meanings of different classes, which treats all graphs at a homogeneous level. Therefore, we update the above maximization of the mutual information equation (Eq. 9) taking into consideration the class label $\mathcal{T}$ as follows:

$$\begin{aligned} &\max_{\mathcal{G}_s} \mathcal{I}(\mathcal{D}, (\mathcal{G}_s, \mathcal{X}_s^{\mathcal{T}})) = \\ &H(\mathcal{D}) - H(\mathcal{D}|\mathcal{G} = \mathcal{G}_s, \mathcal{X} = \mathcal{X}_s^{\mathcal{T}}) \end{aligned} \tag{10}$$

Given a trained GNN model Φ and a prediction $\hat{d}$ or set of predictions, HGExplainer will generate an explanation by identifying a meta-path-based graph $\mathcal{G}_s$ of the computation graph $\mathcal{G}_c$ and a subset of node features $\mathcal{X}_s^F$ that are most influential for the model's prediction $\hat{d}$. Let $\mathcal{X}_s^{\mathcal{T}}$ be a subset of d-dimensional node features. It measures how many probabilities the explainable meta-path-based graph can approximate the original input heterogeneous graph $\mathcal{G}$. The total mutual information maximization is calculated as follows:

$$\max_{\mathcal{G}_s} \mathcal{I}(\mathcal{D}, (\mathcal{G}_s, \mathcal{X}_s^{\mathcal{T}})) = \sum_{z=1}^{\mathcal{T}} \mathcal{I}(M_z) \tag{11}$$

where M is the result of the mutual information maximization for each class $\mathcal{T}$. Note that once the GNN model is trained, entropy term $H(\mathcal{D})$ is constant and no longer changes because the model Φ is fixed for a trained heterogeneous GNN. As a result, maximizing mutual information between the predicted weight distribution and explanation is equivalent to minimizing the conditional entropy $H(\mathcal{D}|\mathcal{G} = \mathcal{G}_s, \mathcal{X} = \mathcal{X}_s^{\mathcal{T}})$ which can be expressed as follows:

$$\begin{aligned} &H(\mathcal{D}|\mathcal{G} = \mathcal{G}_s, \mathcal{X} = \mathcal{X}_s^{\mathcal{T}}) = \\ &-\mathbb{E}_{D|G_s, \mathcal{X}_s^{\mathcal{T}}}[logP(Y|\mathcal{G} = \mathcal{G}_s, \mathcal{X} = \mathcal{X}_s^{\mathcal{T}})] \end{aligned} \tag{12}$$

In the setting of the heterogeneous graphs, the used node embeddings from the classifier keep the heterogeneous node information. In contrast, the edge-type embeddings contain heterogeneous edge information. It promotes the interpreter to learn the edge distributions in a heterogeneous manner. In applications such as RSs, instead of finding an explanation regarding the model's confidence, the users care more about *why the trained model predicts a certain edge weight*. Inspired by [**?**], we modify the conditional entropy objective $\mathcal{I}(\mathcal{H}, (\mathcal{D}|G = G_s))$ with a cross-entropy objective between the ground truth edge weight and the model prediction. The loss function for the HGExplainer is formulated as follows:

$$\mathcal{L}_{latent} = -\sum_{m=1}^{\mathcal{T}} q^m log(\hat{d}) + (1 - q^m) log(1 - \hat{d}) \tag{13}$$

where q is an indicator set to $q = 1$ for positive triples and $q = 0$ for negative ones. Our framework is flexible with various regularization terms to preserve desired properties on the explanation. Following [**?**] to obtain a compact and succinct explanation, we impose a constraint on the explanation size by adding the sum of all elements of the mass parameters as the regularization term, and element-wise entropy is also applied to achieve discrete edge weights further. The loss consists of the loss from the latent parameters defined in Eq. (13) and the cross-entropy reconstruction loss ($\mathcal{L}_{cross}$):

$$\mathcal{L} = \lambda \mathcal{L}_{latent} + \mathcal{L}_{cross} \tag{14}$$

where hyper-parameter $\lambda \in [0, 1]$ is a trade-off weight to balance two loss functions. We iteratively train our model until stop conditions are satisfied, like the maximum number of iterations. The final explanation is an aggregation of the mutual information maximization of each class.

C. Meta-path-based Graph Explanations

The meta-path-based graph $\mathcal{G}_s$ is generated depending on the high-order topology structure, describing the multi-hop structural interactions between two nodes. Different meta-paths M_{path} represent different semantics.

In order to thoroughly learn the information of each meta-path M_{path}, we generate the corresponding meta-path-based subgraphs according to the meta-path definition and then apply aggregation methods to each meta-path-based subgraph. After aggregating the node and edge data within each meta-path, we combine the semantic information revealed by all meta-paths. Following the meta-path types, the generated meta-path-based graph $\mathcal{G}_s$ can be passed to calculate the joint mutual information (as shown in Fig. 2). Then, we map the generated meta-path-based graph with the k-hop subgraph generated for a link (v_1, v_2) to explain the prediction $\hat{d}$. Finally, we use the threshold to identify low-weight edges and remove low-weight edges, and identify the explanation subgraph $\mathcal{G}_s$. To find an optimal threshold, we propose using the Graph Embedding Similarity approach to find a trade-off between the average neighbor distances and the total number of the sampled nodes. The embedding similarity approach is also found in Reinforced Neighbor Sampler [**?**], albeit with different motivations and objectives.

Since the search space of meta-path-based graph $\mathcal{G}_s$ is enormous and a candidate explanation for prediction, direct optimization of HGExplainer's objective needs to be tractable. By doing so, it is consistent that not all edges in the original graph contribute to the model's prediction. For a specific type of node, their connections to the neighbors in different subgraphs carry different semantic information so that each subgraph can be regarded as an interaction graph with specific semantic information. Since the subgraphs are independent, learning tasks can be carried out on each subgraph in parallel, which results in more efficient learning.

D. Node and Graph Classification Tasks

So far, we have focused on the link prediction task, including the RS application. However, HGExplainer provides explanations of node and graph classification without modifying its optimization algorithm. When classifying a node, HGExplainer learns a separate mask for each class of nodes. The same principle applies to graph classification. We make the union of adjacency matrices and semantic information for all the nodes and edges of the graph whose labels we want to explain.

E. Any Heterogeneous GNN Model

Modern Heterogeneous GNNs are based on message-passing, encoder-decoder, or adversarial architectures. Unlike homogeneous graphs, we cannot predict all complex architectures because of the unlimited possibilities of heterogeneous graphs. However, HGExplainer can be applied to different architectures such as RGCN [**?**], HAN [**?**], and R-VGAE [**?**] and applications including RSs such as GraphRec [**?**], and many other. Note that our model can be easily extended to other GNN models and heterogeneous information.

F. Computational Complexity

The number of parameters in HGExplainer's optimization depends on the size of the computation graph $\mathcal{G}_c$ for node v and the subgraph's k-hop value, whose prediction we aim to explain. As a comparison, the time complexity of HGExplainer is similar to GNNExplainer. They have to retrain for the new instance, leading to the time complexity of $O(w|\mathcal{E}|)$, where w is the number of epochs for retraining.

TABLE II
STATISTICS AND PROPERTIES OF THE DATASETS USED IN EXPERIMENTS.

Dataset	#Nodes	#Edges	#Features	#Classes
Hetero SG-Base	13,150	46,472	11	2
SG-Heterophilic	13,150	46,472	11	2
IMDB	10,352	100,836	1,266	2
Epinions	279,737	715,821	1,024	2

IV. EXPERIMENTAL EVALUATION

In this section, we present an experimental evaluation of our framework. We first describe the graphs, implementation details, and experimental setup. Then, we present the evaluation results for link prediction tasks on the homogeneous and heterogeneous graph datasets. Our analysis demonstrates that HGExplainer is a promising method for identifying explanations regarding graph structure and node features. The experiments aim to address the following research questions:

- RQ1: How does HGExplainer perform in explaining link prediction?
- RQ2: What is the impact of the major components of the HGExplainer described in the previous section?
- RQ3: How does HGExplainer perform in explaining heterophilic GNNs?

A. Datasets

To fairly evaluate the performance of our model, we conducted the experiments on two real datasets, including recommendation data from the most popular movie online database IMDB [**?**] and popular social networking website Epinions [**?**]. Each collection allows users to rate items and browse/write reviews. Hence, they provide a large amount of rating information. Moreover, we chose two toy datasets Heterogeneous SG-Base and SG-Heterophilic, which are generated using GraphXAI framework [**?**]. Heterogeneous SG-Base and SG-Heterophilic are homophilic and heterophilic large datasets containing house-shaped motifs for ground-truth explanations. The node features in this graph exhibit homophily, a property commonly found in social networks. With over 10,000 nodes, this graph also provides enough examples of ground-truth explanations for rigorous statistical evaluation of explainer performance. The statistics of these datasets are sketched in Tab. II.

B. Experimental Settings

1) Parameter settings: We use R-VGAE [**?**] GNN architectures for the link prediction task in our evaluation. Finally, we follow GNNExplainer to split train/validation/test with 80/10/10% for all datasets. Each model trained deep neural networks with 300 epochs. We adopt the ADAM optimizer with a learning rate of 0.005. All experiments are conducted on a Linux machine with an NVIDIA Tesla P100-PCIE with 16GB memory. CUDA version is 11.2, and the Driver Version is 460.32.03. HGExplainer is with Python 3.7.13, PyTorch 1.12.0, and PyTorch Geometric 2.1.0.

2) Baselines: Many explainable methods cannot be directly applied to heterogeneous graphs. Nevertheless, we consider the following alternative gradient-based approaches, Grad [**?**] and Grad-CAM [**?**], that can provide insights into predictions made by heterogeneous GNNs. We compute the gradient of the GNN's loss function concerning the adjacency matrix and the associated node features, similar to a saliency map approach. Many explainability methods cannot directly explain graph-based deep learning models (as mentioned in Section II), especially for the link prediction task. To the best of our

TABLE III
THE EVALUATION OF HGEXPLAINER FOR REAL-WORLD RECOMMENDATION DATASETS BASED ON ACCURACY AND MAX-JACCARD METRIC. HGEXPLAINER OBTAINED A HIGHER SCORE ACROSS ALL TWO DATASETS. THE BEST PERFORMANCES ON EACH DATASET ARE SHOWN IN **BOLD**.

Method	**IMDB**				**Epinions**			
	Gen Precision	Gen Recall	Gen F_1	Max-Jaccard	Gen Precision	Gen Recall	Gen F_1	Max-Jaccard
Grad	0.079	0.088	0.082	0.081	0.167	0.177	0.171	0.168
GradCAM	0.083	0.097	0.086	0.084	0.171	0.181	0.174	0.172
GNNExplainer	0.103	0.111	0.108	0.102	0.183	0.191	0.186	0.184
PGExplainer	0.123	0.131	0.127	0.125	0.191	0.201	0.198	0.192
HGExplainer	**0.141**	**0.155**	**0.148**	**0.141**	**0.199**	**0.208**	**0.204**	**0.201**

knowledge, HGExplainer is the first model being evaluated on heterogeneous graph datasets. As a result, GNNExplainer [**?**] and PGExplainer [**?**] are designed for the setting of homogeneous graphs. It generates an interpretable graph by identifying the subgraph of a given graph and a subset of node features. We adapt GNNExplainer and PGExplainer in a heterogeneous setting by applying the same techniques used in our method to deal with heterogeneous graphs.

3) Evaluation metrics: We evaluate the interpretability of models by the fidelity and sparsity [**?**]. Fidelity measures the change of the shift detection result if removing the interpretable edges, and sparsity reflects the percentage of remaining edges after removing the interpretable edges. High values of both fidelity and sparsity indicate the strong interpretability the model has. Furthermore, we proposed the use of the Max-Jaccard across all possible explanations for a given triple. The Max-Jaccard score measures if the explanation method is able to accurately predict one of the possible explanations to choose from. Moreover, the generalized precision (Gen Precision) and recall (Gen Recall) measure if the predicted explanation was given a high intuitive score assigned by users, and the generalized F_1 (Gen F_1) provides an overview of performance on the generalized precision and recall. Following [**?**], we use evaluation metrics for generated datasets, measuring accuracy (GEA), faithfulness (GEF), stability (GES), Counterfactual fairness mismatch (GECF), and Group fairness mismatch (GEGF).

C. Explainability Analysis

The evaluation measures for Explainable Artificial Intelligence (XAI) systems are another essential factor in the design process of XAI systems. Explanations should correspond to different interpretability goals. Hence, to be valid for the intended purpose, the information about the quality of explanations requires different measures. Herman [**?**] notes that reliance on human evaluation of explanations may lead to persuasive explanations rather than transparent systems due to users' preference for simplified explanations.

1) Analyses (RQ1, RQ2): Assessing the quality of GNN explanations is challenging as existing evaluation strategies depend on specific datasets with no or unreliable ground-truth explanations and GNN models. An essential criterion for explanations is that they must be interpretable, i.e., provide a qualitative understanding of the relationship between the input

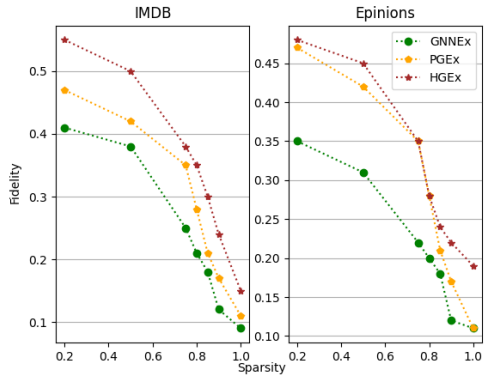

Fig. 3. Since the sparsity scores cannot be fully controlled, we compare different methods with fidelity scores under similar levels of sparsity. A larger fidelity at a given sparsity indicates a higher importance of the extracted explainable subgraph.

nodes and the prediction. Such a requirement implies that explanations should be easy to understand while remaining exhaustive. It means that HGExplainer should consider both the structure of the underlying graph and the associated features when they are available. Tab. III shows the results of our experiments in which HGExplainer jointly considers structural information and information from a small number of feature dimensions. We observed the highest accuracy of HGExplainer, which performed the best explanations regarding the generalized F_1 score on the R-VGAE link prediction task. In addition, we evaluate the performance of HGExplainer on a collection of synthetically generated graphs. The results in Tab. IV show that, while no explanation method performs well across all properties, HGExplainer outperforms other methods on average. In particular, HGExplainer generates more accurate and the least unstable explanations with the second-lowest unfaithfulness score than other GNN explanation methods. While HGExplainer highlights a compact feature representation, PGExplainer has the best counterfactual fairness score, and gradient-based approaches struggle to cope with the added noise, giving high importance scores to irrelevant feature dimensions.

Since evaluating of explanations is challenging in the absence of ground truth, we further conduct quantitative studies to compare these methods. In the absence of ground truth for

TABLE IV

THE EVALUATION OF HGEXPLAINER WITH A BASELINE ON HETEROGENEOUS SG-BASE GRAPH DATASET. ARROWS (↑/↓) INDICATE THE DIRECTION OF BETTER PERFORMANCE. OVERALL, HGEXPLAINER FAR OUTPERFORMS OTHER METHODS IN ACCURACY, STABILITY, AND GROUP FAIRNESS METRICS, WHILE PGEXPLAINER IS BEST FOR COUNTERFACTUAL FAIRNESS, AND GRADIENT METHODS PRODUCE THE MOST FAIR EXPLANATIONS. THE BEST PERFORMANCES ARE SHOWN IN **BOLD**.

Method	GEA (↑)	GEF (↓)	GES (↓)	GECF (↓)	GEGF (↓)
Grad	0.198±0.002s	**0.502**±0.005s	0.748±0.005s	0.166±0.004s	0.069±0.002s
GradCAM	0.204±0.001s	0.634±0.007s	0.311±0.004s	0.044±0.003s	0.041±0.001s
GNNExplainer	0.133±0.003s	0.622±0.006s	0.412±0.005s	0.225±0.007s	0.027±0.002s
PGExplainer	0.140±0.002s	0.632±0.007s	0.241±0.005s	**0.074**±0.003s	0.030±0.002s
HGExplainer	**0.221**±0.002s	0.617±0.006s	**0.224**±0.004s	0.144±0.003s	**0.022**±0.001s

explanations, we can still evaluate post hoc explanations using the desirable properties of the explanations we introduced. The Fidelity metric measures whether the explanations are faithfully important to the model's predictions. It removes the important structures from the input graphs and computes the difference between predictions. In addition, the sparsity metric measures the fraction of structures identified as important by explanation methods. Note that high sparsity scores mean smaller structures are identified as important, which can affect the fidelity scores since smaller structures (high sparsity) tend to be less important (low fidelity). We see in Fig. 3 that HGExplainer has higher fidelity at all sparsity levels than GNNExplainer and PGExplainer on both real datasets. It indicates that our model can better detect important edges on heterogeneous graphs.

2) Heterophilic analyses (RQ3): At first, the homophily assumption defines that nodes with similar features or same class labels are linked together. In contrast, the heterophily defines that linked nodes have dissimilar features and different class labels. We compare HGExplainer with other GNN explainers by generating explanations on GNN models trained on homophilic and heterophilic graphs generated using the SG-Heterophilic generator [**?**]. Then, we compute the graph explanation unfaithfulness scores of output explanations generated using different gradient-based methods, PGExplainer and GNNExplainer with HGExplainer. We find that HGExplainer, like the other methods, produces more faithful explanations when ground-truth explanations are homophilic than when ground-truth explanations are heterophilic (i.e., low unfaithfulness scores for light blue bars in Fig. 4). Since most local neighbor nodes are not in the same class, heterophily makes the explanations more challenging. Moreover, extracting explainable subgraphs from highly heterophilic graph data is much more complex, where proximal and distant topological structures must be discovered and exploited. These results reveal an essential limitation of existing GNN explainers and highlight an opportunity for future algorithmic innovation in GNN explainability.

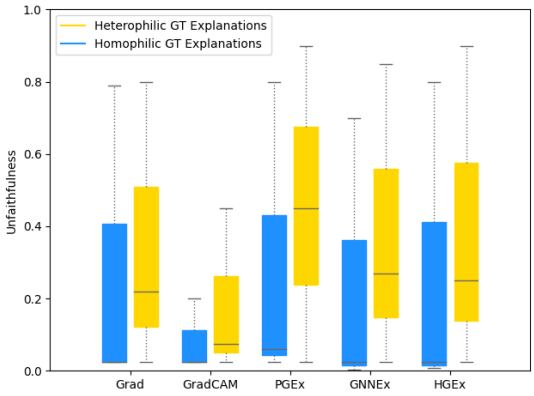

Fig. 4. Unfaithfulness scores across four GNN explainers and HGExplainer on SG-Heterophilic graph dataset consisting of either homophilic or heterophilic ground-truth (GT) explanations. GNN explainers produce more faithful explanations (lower GEF scores) on homophilic graphs than heterophilic graphs.

V. CONCLUSION AND FUTURE WORK

In this paper, we present HGExplainer, which provides an improved insight into the explanations of the link prediction task built on heterogeneous graphs. Extensive experimental results show that our method can provide a human-intelligible reasoning process with acceptable classification accuracy. However, it requires additional work on time complexity and the heterophilic model improvement. To the best of our knowledge, this work is the first one that explains the link prediction task in the context of applications such as RSs, and we hope it will serve as a beacon for works to come.

We only incorporate the heterogeneous graphs into the link prediction task and RS. At the same time, many real-world industries are associated with rich other-side information on users and items. For example, social RSs and their users are associated with social interactions and rich attributes. Therefore, exploring explanations of GNN for social recommendation with attributes would be an interesting future direction. Moreover, the explanations in RSs are crucial for end users, so one of the future steps would be to make them more interpretable using a neurosymbolic approach, including knowledge graphs and first-order logic. Finally, a user study should conducted to achieve better trustworthy feedback.

ACKNOWLEDGMENT

The contribution has been funded by the Telecom SudParis, Institut Polytechnique de Paris and EFREI Paris, Pantheon Assas University. We thank these institutions for their support.

References

[1] Agarwal, C., Queen, O., Lakkaraju, H., Zitnik, M.: Evaluating Explainability for Graph Neural Networks. Scientific Data **10**(144) (2023)

[2] Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C.L., Batra, D., Parikh, D.: VQA: Visual Question Answering. In: Proceedings of the IEEE International Conference on Computer Vision. vol. 123 (2016)

[3] Berg, R.v.d., Kipf, T.N., Welling, M.: Graph Convolutional Matrix Completion. arXiv:1706.02263 (2017)

[4] Chicaiza, J., Valdiviezo-Diaz, P.: A Comprehensive Survey of Knowledge Graph-Based Recommender Systems: Technologies, Development, and Contributions. Information **12**(6), 232 (2021)

[5] Cui, P., Wang, X., Pei, J., Zhu, W.: A Survey on Network Embedding. Proceedings of the IEEE Transactions on Knowledge and Data Engineering **31**(5), 833–852 (2018)

[6] Dai, E., Zhao, T., Zhu, H., Xu, J., Guo, Z., Liu, H., Tang, J., Wang, S.: A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. arXiv:2204.08570 (2022)

[7] Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. vol. 29, pp. 3844–3852 (2016)

[8] Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: Graph Neural Networks for Social Recommendation. In: The Proceedings of the World Wide Web Conference. pp. 417–426 (2019)

[9] Fang, Y., Zhang, Q., Yang, H., Zhuang, X., Deng, S., Zhang, W., Qin, M., Chen, Z., Fan, X., Chen, H.: Molecular Contrastive Learning with Chemical Element Knowledge Graph. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 3968–3976 (2022)

[10] Gasteiger, J., Groß, J., Günnemann, S.: Directional Message Passing for Molecular Graphs. arXiv:2003.03123 (2020)

[11] Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., He, Q.: A Survey on Knowledge Graph-based Recommender Systems. IEEE Transactions on Knowledge and Data Engineering **34**(8), 3549–3568 (2020)

[12] Hamilton, W., Ying, Z., Leskovec, J.: Inductive Representation Learning on Large Graphs. vol. 30, pp. 1025–1035 (2017)

[13] Harper, F.M., Konstan, J.A.: The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems **5**(4), 1–19 (2015)

[14] Herman, B.: The Promise and Peril of Human Evaluation for Model Interpretability. arXiv:1711.07414 (2017)

[15] Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., Chang, Y.: Graphlime: Local Interpretable Model Explanations for Graph Neural Networks. IEEE Transactions on Knowledge and Data Engineering. pp. 1–6 (2020)

[16] Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., Sayres, R.: Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In: International Conference on Machine Learning. pp. 2668–2677 (2019)

[17] Kim, J., Kim, T., Kim, S., Yoo, C.D.: Edge-labeling Graph Neural Network for Few-Shot Learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11–20 (2019)

[18] Li, I., Fabbri, A., Hingmire, S., Radev, D.: R-VGAE: Relational-variational Graph Autoencoder for Unsupervised Prerequisite Chain Learning. arXiv:2004.10610 (2020)

[19] Li, P., Yang, Y., Pagnucco, M., Song, Y.: Explainability in Graph Neural Networks: An Experimental Survey. arXiv:2203.09258 (2022)

[20] Li, X., Liu, Z., Guo, S., Liu, Z., Peng, H., Yu, P.S., Achan, K.: Pre-training Recommender Systems via Reinforced Attentive Multi-relational Graph Neural Network. pp. 457–468 (2021)

[21] Liu, J., Wang, Y., Xiang, S., Pan, C.: HAN: An Efficient Hierarchical Self-Attention Network for Skeleton-Based Gesture Recognition. arXiv:2106.13391 (2021)

[22] Luo, D., Cheng, W., Xu, D., Yu, Wenchao Zong, B., Chen, H., Zhang, X.: Parameterized explainer for graph neural network. Advances in Neural Information Processing Systems **33**, 19620–19631 (2020)

[23] Massa, P., Avesani, P.: Trust-aware Recommender Systems. In: Proceedings of the 2007 ACM Conference on Recommender Systems. pp. 17–24 (2007)

[24] Mika, G.P.: Toward a Transparent Recommender System. In: Proceedings of the 14th International Rule Challenge, 4th Doctoral Consortium, and 6th Industry Track @ RuleML+RR. vol. 2644, pp. 111–119 (2020)

[25] Pope, P.E., Kolouri, S., Rostami, M., Martin, C.E., Hoffmann, H.: Explainability Methods for Graph Convolutional Neural Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10772–10781 (2019)

[26] Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling Relational Data with Graph Convolutional Networks. pp. 593–607 (2018)

[27] Schlichtkrull, M.S., De Cao, N., Titov, I.: Interpreting Graph Neural Networks for NLP with Differentiable Edge Masking. arXiv:2010.00577 (2020)

[28] Schnake, T., Eberle, O., Lederer, J., Nakajima, S., Schütt, K.T., Müller, K.R., Montavon, G.: Higher-Order Explanations of Graph Neural Networks via Relevant Walks. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 7581–7596 (2020)

[29] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 618–626 (2019)

[30] Sun, M., Zhang, X., Zheng, J., Ma, G.: DDGCN: Dual Dynamic Graph Convolutional Networks for Rumor Detection on Social Media. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 4611–4619 (2022)

[31] Vu, M., Thai, M.T.: Probabilistic Graphical Model Explanations for Graph Neural Networks. Advances in Neural Information Processing Systems pp. 12225–12235 (2020)

[32] Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous Graph Attention Network. In: The World Wide Web Conference. pp. 2022–2032 (2019)

[33] Wang, Y., Song, Y., Li, S., Cheng, C., Ju, W., Zhang, M., Wang, S.: DisenCite: Graph-Based Disentangled Representation Learning for Context-Specific Citation Generation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 11449–11458 (2022)

[34] Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B.: Graph Neural Networks in Recommender Systems: A Survey. ACM Computing Surveys **55**(5), 1–37 (2022)

[35] Xu, D., Cheng, W., Luo, D., Liu, X., Zhang, X.: Spatio-temporal Attentive RNN for Node Classification in Temporal Attributed Graphs. In: IJCAI. pp. 3947–3953 (2019)

[36] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How Powerful are Graph Neural Networks? arXiv:1810.00826 (2018)

[37] Ying, R., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: GNNExplainer: Generating Explanations for Graph Neural Networks. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. pp. 9244–9255 (2019)

[38] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In: 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining. pp. 974–983 (2018)

[39] Yuan, H., Tang, J., Hu, X., Ji, S.: XGNN: Towards Model-Level Explanations of Graph Neural Networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining. pp. 430–438 (2020)

[40] Yuan, H., Yu, H., Gui, S., Ji, S.: Explainability in Graph Neural Networks: A Taxonomic Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(5), 5782–5799 (2022)

[41] Yuan, H., Yu, H., Wang, J., Li, K., Ji, S.: On Explainability of Graph Neural Networks via Subgraph Explorations. In: International Conference on Machine Learning. pp. 12241–12252 (2021)

[42] Yun, S., Kim, S., Lee, J., Kang, J., Kim, H.J.: Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. vol. 34, pp. 12683–13694 (2021)

[43] Zhang, M., Cui, Z., Neumann, M., Chen, Y.: An End-to-End Deep Learning Architecture for Graph Classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32, pp. 4438–4445 (2018)