Stratified GNN Explanations through Sufficient Expansion

Yuwen Ji[1], **Lei Shi** [1*], **Zhimeng Liu**[2], **Ge Wang**[2]

[1]Beihang University, Beijing, China
[2]University of Science and Technology Beijing, Beijing, China
{jiyuwen, leishi}@buaa.edu.cn, d202210287@xs.ustb.edu.cn, gewang@ustb.edu.cn

Abstract

Explaining the decisions made by Graph Neural Networks (GNNs) is vital for establishing trust and ensuring fairness in critical applications such as medicine and science. The prevalence of hierarchical structure in real-world graphs/networks raises an important question on GNN interpretability: "On each level of the graph structure, which specific fraction imposes the highest influence over the prediction?" Currently, the prevailing two categories of methods are incapable of achieving multi-level GNN explanation due to their flat or motif-centric nature. In this work, we formulate the problem of learning multi-level explanations out of GNN models and introduce a stratified explainer module, namely STF-Explainer, that utilizes the concept of sufficient expansion to generate explanations on each stratum. Specifically, we learn a higher-level subgraph generator by leveraging both hierarchical structure and GNN-encoded input features. Experiment results on both synthetic and real-world datasets demonstrate the superiority of our stratified explainer on standard interpretability tasks and metrics such as fidelity and explanation recall, with an average improvement of 11% and 8% over the best alternative on each data type. The case study on material domains also confirms the value of our approach through detected multi-level graph patterns accurately reconstructing the knowledge-based ground truth.

Introduction

Interpreting Graph Neural Networks (GNNs) (Dwivedi et al. 2020; Wu et al. 2020) is considered a key agenda for Explainable AI as GNNs grow flourishing in our machine learning toolbox, yet still suffer greatly from the well-known black box problem. Fortunately, the field has been witnessing a burst of literature in the recent few years (Yuan et al. 2022; Kakkad et al. 2023). These successful proposals build a solid theoretical foundation for GNN explanation. Both instance-level post-hoc explanation methods (notably the seminal work of GNNExplainer (Ying et al. 2019)) and mechanism for model-level explanation (e.g., XGNN (Yuan et al. 2020)) are introduced. They help to establish trust and ensure fairness in critical applications such as chemistry (Reiser et al. 2022), material science (Choudhary et al. 2022), and finance (Wu, Chao, and Li 2023).

*Corresponding author

Despite the surge of this topic, there is currently little work considering the hierarchically clustered structure within the graph whilst explaining the GNN on top of these graphs. In fact, such hierarchical structure arises naturally in scientific or social contexts and can be central to their graph characteristics and application-level properties. For example, a popular class of compound in material science, namely Metal-Organic Frameworks (MOFs), is composed of complex molecular graphs built recursively from a variety of Secondary Building Units (SBUs). As shown in Figure 1(a), the catalytic performance of MOF in a chemical reaction is influenced by both the local atomic structure of the underlying SBU and the connectivity among high-level SBUs.

In this work, we consider the problem of explaining GNN models at multiple levels aligning with the inherent hierarchical structure of the input graph. Currently, the only hierarchy-aware GNN explanation method, i.e. MotifExplainer (Yu and Gao 2022), focuses on extracting influential high-level clusters out of the graph data, but misses the opportunity to credit inter-cluster connectivity and high-level subgraphs, of the pattern crucial to most classical GNN explainers at the input graph level. Meanwhile, the subgraph explanation by classical methods (Ying et al. 2019; Luo et al. 2020; Wang et al. 2021) can also be hierarchically clustered according to the prior knowledge on the graph and achieves multi-level explanations. Yet, this mash-up approach does not capture the importance of group-based features at multiple graph levels, imposing limitations on the essential processes of candidate subgraph generation and optimization.

This paper studies explanation methods that natively support multi-level GNN interpretation. Though existing approaches have established a general pipeline for GNN explanation and adapted effective optimization algorithms, there are still several challenges in solving our recent problem. First, the classical objective function based on mutual information is only defined at the input graph level. The hierarchical graph structure vital to our work is yet to be incorporated. Second, the optimization framework adopted by existing methods depends on the context that the individual features optimized have already been embedded by the GNN explained, which can be trained altogether to optimize the objective. In contrast, high-level graph features are not necessarily embedded in the original GNN model. Finally, evaluating multi-level GNN explanations is a new task that

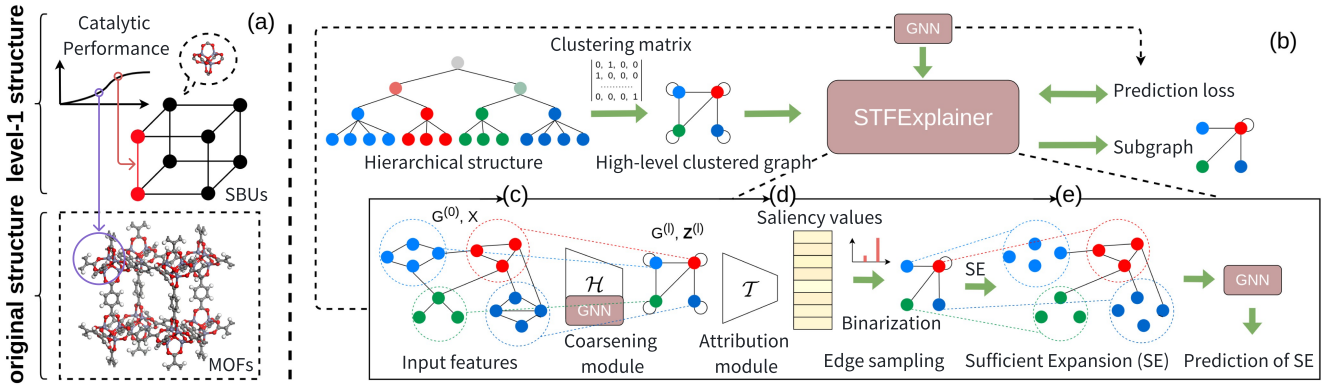

Figure 1: The framework of STFExplainer. (a) a case of the multi-level structure of MOF molecular graph and its influence on catalytic performance; (b) an overview of our GNN explanation pipeline; (c) the first stage by learning high-level graph feature embeddings through coarsening; (d) the second stage by computing important scores on high-level clustered edges for explanatory subgraph sampling. (e) the final stage of explainer optimization through sufficient expansion.

requires examining the utility of existing benchmark data and compiling performance metrics in a reasonable way.

We make several contributions to tackle these challenges.

- We formally define the objective function for multi-level GNN explanations, compatible with the widely accepted measure of mutual information. To link between high-level and input-stage graph features, a new concept of sufficient expansion is introduced;
- We propose an improved optimization framework to solve the objective function. By augmenting the original learning pipeline with additional graph coarsening and attributing modules, high-level graph features can be appropriately aggregated, represented, and optimized;
- We develop an evaluation methodology using existing benchmarks, real-world material graphs, and synthetic instances with hierarchical structures. We carefully select relevant and available metrics for the evaluation.

Experimental results on both classification and regression tasks demonstrate the superiority of our method. The fidelity/recall (i.e., key metrics for GNN explanation) is constantly ranked in top-2, with an average improvement of 11% and 8% over the best alternative in real-world and synthetic datasets respectively. A chemical case study also confirms the usefulness of extracted multi-level explanations.

Related Work

We consider two classes of most relevant work, more literature on GNN explanation methods can be found in the latest survey (Yuan et al. 2022; Kakkad et al. 2023).

Hierarchy-Aware GNN Explanations

GNN explanation methods considering hierarchical structure are limited. MotifExplainer (Yu and Gao 2022) uses clustered graph structures as motifs and ranks their representations through an attention-based method, providing more intuitive and understandable explanations. However, it cannot explain the relationship between clustered structures.

On the other hand, GLGExplainer (Azzolin et al. 2022) explains high-level relationships between local graph patterns by projecting them onto learned prototypes forming concept vectors. These vectors are used to train an entropy-based logic explainable network (Barbiero et al. 2022; Ciravegna et al. 2023) for class prediction alignment. However, it does not explain the structural relationships between clusters as the explanation is a logical formula of the prototypes. Previous work in (Ying et al. 2018; Tang et al. 2021) customizes a layered model in a priori manner for hierarchy-aware GNN explanation. (Kengkanna and Ohue 2023) explores the impact of multi-level graph representation on model learning, but achieving model-agnostic multi-level explanations is infeasible as priori methods require embedding each form of knowledge into the model structure.

Post-hoc GNN Explanations

Traditional GNN explanation methods combined with hierarchical structure can generate multi-level explanations. Parametric explanation methods (Ying et al. 2019; Yuan et al. 2020; Luo et al. 2020; Vu and Thai 2020; Wang et al. 2021) additionally train parametrized models. For instance, GNNExplainer learns soft masks for every graph and applies them to recover predictions. XGNN trains a graph generator that outputs class-wise patterns for the explained GNN. PGExplainer employs a probabilistic generative model to collectively explain GNN on multiple graphs. PGMExplainer introduces a Bayesian network to model pairs of perturbed graphs and prediction changes. Refine provides a global explanation by pre-training a class-aware attributor and achieves local explanation by fine-tuning. Parametric explainers focus on the fidelity of input groups but become suboptimal when the high-level structure is considered.

In the second class, non-parametric GNN explanation methods (Baldassarre and Azizpour 2019; Schnake et al. 2021) compute feature contribution without the need for end-to-end training. They use heuristics such as gradient-like scores that backpropagate model prediction loss to input features (Pope et al. 2019). However, non-parametric meth-

Notations	Descriptions
$G, \mathcal{V}, \mathcal{E}$	the input graph, with its node and edge set
$\mathbf{A}, \mathbf{X}, \mathbf{Z}$	adjacent matrix, node features & embedding of G
f, f_{emb}, f_{cls}	GNN, its node embedding & classification layers
$Y, \hat{Y}$	output variable and distribution of GNN models
$G^{(l)}, \mathcal{V}^{(l)}, \mathcal{E}^{(l)}$	the level-l graph, with its node and edge set
$S^{(l)}, \mathbf{A}^{(l)}$	cluster assignment & adjacency matrix in level-l
$G_s, G_s^{(l)}$	explanatory subgraph of input and level-l graph

Table 1: Notations used throughout this work.

ods are generally less preferred due to their inability to incorporate fidelity constraints in deriving GNN explanations, especially when high-level graph fidelity is considered.

Problem Definition

Table 1 lists notations related to GNN, its explanations, and graph's hierarchical structure. We then formally define multi-level GNN explanation problem focused in this work.

Notations for GNN Explanation

Denote input graph as $G = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$, and its adjacency matrix as $\mathbf{A} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$, where $\mathbf{A}_{ij} = 1$ means an edge from node v_i to v_j, and $\mathbf{A}_{ij} = 0$ otherwise. The node feature matrix is denoted by $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times d}$.

Without loss of generality, we consider the classification task where GNN works as a graph classifier f. The f learns the output distribution $\hat{Y}$ for each input graph by the conditional distribution $\hat{Y} \sim P_f(Y|G, \mathbf{X})$, where Y denotes the output variable with a class set of $1, ..., C$. Inside the GNN model, the key is to learn the final node embeddings through convolution layers, holistically denoted by $\mathbf{Z} = f_{emb}(G, \mathbf{X})$ where $\mathbf{Z}_i$ is the embedding for each node v_i. For the downstream classification task, these node embeddings are readout and then classified, normally via a Multi-Layer Perceptron (MLP). The readout and MLP layers are defined together as $\hat{Y} \sim f_{cls}(\mathbf{Z})$. Notice that $f(\cdot) = f_{cls}(f_{emb}(\cdot))$.

Explaining GNNs. Mainstream GNN explanation methods extract a subgraph $G_s \in G$ out of each input graph, as well as a set of contributing node features $\mathbf{X}_s \in \mathbf{X}$. To make sure the subgraph encompasses salient features of the GNN model, a representative objective function is introduced in the work of (Ying et al. 2019) and has been widely adopted:

$$\min_{G_s} -MI(\hat{Y}, G_s) + \mathcal{L}(G_s) \tag{1}$$

where $MI(\hat{Y}, G_s)$ denotes the mutual information between the explanatory subgraph and the outcome variable. $\mathcal{L}(\cdot)$ denotes the regularizer imposing sparsity constraints on the subgraph explanation. The mutual information can be seen as a relevance score of the extracted subgraph w.r.t the outcome, indicating the importance of the subgraph feature.

Hierarchical structure of graph data. In many science/social scenarios, graphs are formed with clustered structures, where some groups of nodes become closer/similar to each other than to the nodes outside the group. More often than not, this clustered structure happens recursively at all levels of graph and collectively determines the graph's nature, as well as the performance of various downstream tasks. As shown in Figure 1(b), we represent the graph's hierarchical structure by a clustering tree. The tree is defined by a list of cluster assignment matrices $S^{(l)} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}^{(l)}|}$ ($l = 1, \cdots$), where $\mathcal{V}^{(l)}$ denotes the node set of the level-l clustered graph $G^{(l)}$ and $S_{ij}^{(l)} = 1$ indicates that the original node v_i belongs to the level-l cluster node $v_j^{(l)}$ and 0 otherwise. We omit that some ambiguous nodes can have mixed upper-level membership, which requires minor changes.

Multi-Level GNN Explanation

As mentioned, the graph structure at multiple levels collectively determines downstream performance. For example, in the material design process of MOFs (Figure 1(a)), scientists decompose their crystal structure into SBUs and SBUs-links. Crucial SBUs and links are identified based on their co-occurrence with desired MOF properties, e.g., strong catalytic performance. Scientists can drill down to the internal structure of each SBU and study the functionality of low-level composition of metal and organic molecules to design new SBUs for high performance. Understanding GNN explanations at multiple graph levels becomes vital.

It is straightforward to define the objective for multi-level GNN explanations according to the heuristics of Eq. (1):

$$\min_{G_s^{(l)}} -MI(\hat{Y}, G_s^{(l)}) + \mathcal{L}(G_s^{(l)}), \ \ \forall l \geq 1 \tag{2}$$

where $G_s^{(l)} \subseteq G^{(l)}$ denotes level-l explanatory subgraph. The mutual information between $G_s^{(l)}$ and GNN's output $\hat{Y}$ is still used as the relevance score of high-level subgraph.

The mutual information is normally computed by decomposing into the conditional entropy. For example, in Eq. (1), the conditional entropy $H(\hat{Y}|G_s)$ can be empirically estimated by feeding the extracted subgraph G_s into the trained GNN and obtaining the output distribution. However, the new objective in Eq. (2) cannot be treated in the same manner. $H(\hat{Y}|G_s^{(l)})$ is intractable as the GNN does not directly take a high-level clustered graph as input. To make the objective traceable, we need to introduce an expansion function $E^{(l)}(\cdot)$ that translates the clustered graph at level l into a particular low-level subgraph of the original full graph G.

$$\min_{G_s^{(l)},\, G_s^{(0)}=E^{(l)}(G_s^{(l)})} -MI(\hat{Y}, G_s^{(0)}) + \mathcal{L}(G_s^{(l)}) \tag{3}$$

where $G_s^{(0)}$ denotes the expanded subgraph used for mutual information computation. As a result, to solve Eq. (3), we need to not only optimize the objective similar to Eq. (1), but also determine a reasonable expansion function.

Stratified GNN Explanation

We introduce a stratified GNN explainer, or STFExplainer for short, to fulfill the objective of the above problem and achieve multi-level GNN explanations.

STFExplainer Using Sufficient Expansion

The keys to solving Eq. (3) lie in first determining the expansion function $E^{(l)}(\cdot)$ and then minimizing the objective in the form of Eq. (1), where the optimization method proposed in the literature can be applied. To appropriately select the expansion function, we need to gain a deeper understanding of the objective. Take a closer look at both Eq. (1) and Eq. (2), though in the same form, the two objectives have subtle differences. The default objective of Eq. (1) elects the subgraph having the largest mutual information with the outcome variable while enforcing the sparsity constraint. The subgraph is defined on the original graph such that node/edge features with negative relevance to the outcome are all removed. On the other hand, the new objective of Eq. (2) selects a subgraph from the level-l graph. It is implied that, at each graph level, the features work in a group manner according to the cluster boundary. Though the final objective of Eq. (3) can be minimized by expanding each cluster to all the underlying nodes/edges with positive relevance, the optimization results diverge from our goal of uncovering high-level features that collectively exert influence.

Our analysis is supported by the representation theory of (Wang et al. 2022). They define the sufficient representation of a random variable $\mathcal{X}$ as containing all relevant information needed for inferences about the underlying distribution.

Definition 4.1 (Sufficient Representation). *The representation z_1^{suff} of random variable v_1 is sufficient for random variable v_2 if and only if:*

$$MI(z_1^{suff}, v_2) = MI(v_1, v_2) \tag{4}$$

According to the theory of sufficient representation, when mapped to the original graph, the extracted level-l subgraph $G_s^{(l)}$ should have the same mutual information with the resulting subgraph at the lowest/original graph level, w.r.t the outcome variable. The only candidate of such mapping turns out to be the full expansion of $G_s^{(l)}$ to every lowest-level node/edge belonging to clusters/edges in $G_s^{(l)}$, which is defined as the *sufficient expansion* function $SE(\cdot)$. The approach of sufficient expansion is validated through experiments in Section 5. It is reported that, whenever a sampling function is used instead of the sufficient expansion regardless of the sampling rate, the resulting mutual information will quickly deviate from the original value by the sufficient expansion, i.e., from the mutual information of the extracted high-level subgraph. Formally, the sufficient expansion of a level-l subgraph $G_s^{(l)}$ to the input graph can be defined by:

$$SE(\mathbf{A}_s^{(l)}) = \mathbf{A} \odot (S^{(l)} \cdot \mathbf{A}_s^{(l)} \cdot S^{(l)^\top}) \tag{5}$$

where $SE(\mathbf{A}_s^{(l)})$ and $\mathbf{A}_s^{(l)}$ are adjacent matrix of $SE(G_s^{(l)})$ and $G_s^{(l)}$, respectively. The multiplication of $\mathbf{A}$ ensures that the explanatory subgraphs exclude non-existent edges. In this way, we replace expansion $E(\cdot)$ with sufficient expansion $SE(\cdot)$ and rewrite the objective function of Eq. (3) as:

$$\min_{G_s^{(l)}} -MI(\hat{Y}, SE(G_s^{(l)})) + \mathcal{L}(G_s^{(l)}) \tag{6}$$

Algorithm 1: Training for explaining GNN at level-l

Input: $\{(G_o^{(0)}, G_o^{(l)}, S^{(l)}, \mathbf{X}, \mathbf{Z}, \hat{Y}_o)\}$, $f_{cls}(f_{emb}(\cdot))$
for each epoch **do**
  for each graph $G_o^{(0)}$ **do**
    $\mathbf{Z}^{(l)} \leftarrow$ get level-l embedding with Eq. (7)
    $\omega^{(l)} \leftarrow$ saliency value calculated with Eq. (8)
    $\hat{G}_s^{(l)} \leftarrow$ subgraph sampling from Eq. (9)
    $SE(\hat{G}_s^{(l)}) \leftarrow$ sufficient expansion with Eq. (5)
    $\hat{Y}_s \leftarrow f_{cls}(f_{emb}(SE(\hat{G}_s^{(l)}), \mathbf{X}))$
  end for
  Compute loss with Eq. (6) & Update ψ_1 and ψ_2
end for

Optimization Framework

We first consider the representative optimization method proposed by (Luo et al. 2020), which solves Eq. (1) in three steps. First, they apply feature attribution algorithms to compute a relevance score for each embedding of the input graph G. Second, feature selection is performed to pick important edges based on these scores and form the candidate subgraph for explanation. Finally, the subgraph is substituted into the objective function and optimized iteratively.

Directly applying the above method to Eq. (6) is prohibitive as the original GNN does not generate embeddings for high-level clusters inherently. Then relevance scores cannot be computed for cluster-level edges as those for original graph edges. Our STFExplainer introduces an improved framework: 1) learns high-level group-based embedding via coarsening module; 2) assigns relevant scores to the cluster-level edges using updated attribution module; 3) extracts explanatory subgraph using $SE(\cdot)$, evaluated and optimized in the context of Eq. (6). For clarity, refer to Algorithm 1.

Hierarchy-aware coarsening module. On top of the node embeddings $\mathbf{Z} = f_{emb}(G, \mathbf{X})$ by the GNN model, we first learn a coarsening module $\mathcal{H}$ that aggregates the representation of each high-level cluster and obtain its feature embedding $\mathbf{Z}^{(l)} \in \mathbb{R}^{|\mathcal{V}^{(l)}| \times h^*}$, where $|\mathcal{V}^{(l)}|$ is the number of clusters in the l-th graph level, and h^* denotes the output embedding size of the coarsening module. Formally, we have:

$$\mathbf{Z}_j^{(l)} = \mathcal{H}(S^{(l)}, \mathbf{Z}_j) = AGG(\{\mathbf{DNN}_{\psi_1}(\mathbf{Z}_i) | S_i^{(l)} = j\}) \tag{7}$$

where $\mathbf{DNN}_{\psi_1}$ is a deep neural network parameterized with ψ_1, $AGG(\cdot)$ is the pooling operation.

Multi-level attribution module. We then design an attribution module $\mathcal{T}$, which computes saliency value $\omega_{ij}^{(l)}$, aka the relevance score, for each level-l edge among clusters.

$$\omega_{ij}^{(l)} = \mathbf{MLP}_{\psi_2}([\mathbf{Z}_i^{(l)}; \mathbf{Z}_j^{(l)}]) \tag{8}$$

where $\mathbf{MLP}_{\psi_2}$ is a MLP parameterized with ψ_2, and $[\cdot;\cdot]$ is the concatenation operation.

Subgraph generation and optimization. Our third step largely follows (Luo et al. 2020). Reparameterization trick is

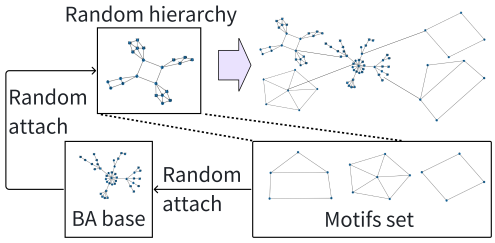

Figure 2: Generation of a synthetic dataset over BA model.

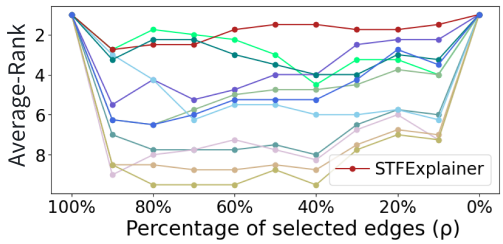

Figure 3: Average ranks of fidelity metric (varying ρ).

applied to enable a continuous relaxation of edge sampling:

$$\hat{e}_{ij}^{(l)} = \sigma((log\frac{\epsilon}{1-\epsilon} + \omega_{ij}^{(l)})/\beta) \tag{9}$$

where $\epsilon \sim Uniform(0,1)$ is the random number for sampling, and the sigmoid function σ with a temperature hyperparameter β is used to translate the relevance score $\omega_{ij}^{(l)}$ to the edge weight $\hat{e}_{ij}^{(l)}$, which is binarized with $\beta \to 0$. The edge weights are used to generate a candidate subgraph $\hat{G}_s^{(l)}$ in level-l. Finally, the mutual information between the sufficient expansion of the candidate graph $\hat{G}_s^{(l)}$ and the prediction outcome is maximized. The objective becomes:

$$\min_{\psi_1,\psi_2} -\mathbb{E}_\epsilon \mathbb{E}_c[P_f(Y=c|G)\log P_f(Y=c|SE(\hat{G}_s^{(l)}))] \tag{10}$$

Theoretical Analysis on STFExplainer

Computational complexity. Close to the work of (Luo et al. 2020) with a similar optimizer design, our method only extracts subgraph from a coarse-grained graph with a size of $|\mathcal{V}^{(l)}| \times |\mathcal{V}^{(l)}|$, smaller than the input graph of size $|\mathcal{V}| \times |\mathcal{V}|$. Additionally, the coarsening module $\mathcal{H}$ can be shared across all high-level graphs, which further reduces the complexity.

Comparison of explanatory subgraphs in different levels. Domain experts often need to compare the explanation patterns at multiple graph levels, asking: "Which semantic level on the knowledge hierarchy is more important for graph inference?" STFExplainer introduces a saliency vector $F_S \in \mathbb{R}^{L\times 2}$ to rank the utility of all graph levels up to L, by comparing the intra- and inter-cluster graph structure:

$$\mathcal{P}_{intra}^{(l)} = MI(\hat{Y}, G^{(l)} = \mathbf{A}^{(l)} \odot I^{(l)}) \tag{11}$$

$$\mathcal{P}_{inter}^{(l)} = MI(\hat{Y}, G^{(l)} = \mathbf{A}^{(l)} - \mathbf{A}^{(l)} \odot I^{(l)}) \tag{12}$$

$$F_S[l]_{l\le L} = softmax(\mathcal{P}_{intra}^{(l)}, \mathcal{P}_{inter}^{(l)}) \tag{13}$$

where $I^{(l)}$ is an identity matrix.

Experiment

We evaluate STFExplainer with a variety of data and tasks relevant to the multi-level explanation problem, including a few benchmark data for generic GNN explanation and more settings tailored to our special requirement.

Data and Task

Real-world molecular graphs from chemistry/material science and a synthetic graph dataset with injected hierarchical explanations are considered. Both classification and regression tasks are investigated, with the top-level hierarchy used as the explanation level. The MUTAG and BA-motif datasets have 2 levels of hierarchy, while the QMOFs dataset has 3. The hierarchical structure is extracted based on domain knowledge or ground truth, without any supernode cutoff.

MUTAG classification. We select the widely used Mutagenicity dataset (MUTAG) (Kazius, McGuire, and Bursi 2005; Riesen and Bunke 2008; Ying et al. 2019), which contains 4,337 molecular graphs labeled with two classes based on mutagenic effect. The mutagenicity of a molecular graph is correlated with its structure, such as the presence of certain local compounds like rings. We use the clustering algorithm from MotifExplainer (Yu and Gao 2022) to extract hierarchical structure from MUTAG graphs and apply a standard GCN model (Kipf and Welling 2016) for classification.

Catalytic performance regression of MOFs graphs. We consider the molecular graph of MOFs from the material domain due to its structure-performance relationship which is key to MOF design. MOFs graphs exhibit an unambiguous hierarchical structure with SBUs and Links, corresponding to clusters. In a typical setting, the QMOFs dataset (Rosen et al. 2021, 2022) is used to infer the bandgap value of each MOF, critical for catalytic performance. The Schnet model (Schütt et al. 2017), customized for MOFs inference, achieves a mean absolute error (MAE) of 0.298 in bandgap regression and is explained in our experiment.

Motif graph classification. Similar to previous approaches in (Luo et al. 2020), we generate a synthetic dataset with 10,000 candidate graphs, namely BA-Hierarchy-motif, by incorporating a 2-level structure into the Barabasi-Albert (BA) graphs. As shown in Figure 2, each graph has a BA component as a base, which is appended with a square structure. On the square, each of the four nodes is randomly substituted with either a $house$ or $wheel$ motif. In addition, $house$ and $wheel$ motifs are also randomly appended to the BA base as noise. The classification task is to infer whether $house$ or $wheel$ motifs are more frequent in the square of each graph. Again, the standard GCN model is applied.

Alternative Methods

STFExplainer is compared with three types of alternatives.

Hierarchical-aware GNN explanations. We modified MotifExplainer to detect high-level explanatory subgraphs by

	MUTAG			QMOFs		
	Fidelity (ACC)	Fidelity (CE)	$Spar@0.1\%$	Fidelity (R^2)	Fidelity (MAE)	$Spar@5\%$
Motif-mul (-36%)	0.814 ± 0.002	0.662 ± 0.023	0.896 ± 0.004	0.259 ± 0.006	0.887 ± 0.001	0.259 ± 0.001
Motif-add (-36%)	0.817 ± 0.004	$\mathbf{0.654 \pm 0.020}$	$\mathbf{0.905 \pm 0.006}$	0.247 ± 0.001	0.894 ± 0.003	0.255 ± 0.001
Motif-max (-38%)	0.812 ± 0.008	0.686 ± 0.029	$\mathbf{0.898 \pm 0.003}$	0.246 ± 0.002	0.894 ± 0.002	0.256 ± 0.004
SA (-11%)	$\mathbf{0.943 \pm 0.000}$	0.779 ± 0.000	0.866 ± 0.000	0.513 ± 0.002	0.588 ± 0.000	0.459 ± 0.000
DeepLift (-11%)	0.943 ± 0.000	0.779 ± 0.000	0.866 ± 0.000	0.517 ± 0.002	0.591 ± 0.001	0.204 ± 0.001
CXPlain (-50%)	0.845 ± 0.011	1.183 ± 0.001	0.782 ± 0.009	0.317 ± 0.025	0.840 ± 0.021	0.235 ± 0.023
GNNExplainer (-23%)	0.923 ± 0.001	1.054 ± 0.001	0.859 ± 0.000	0.481 ± 0.003	0.580 ± 0.001	0.446 ± 0.001
PGExplainer (-44%)	0.905 ± 0.000	0.820 ± 0.001	0.852 ± 0.000	0.506 ± 0.001	0.537 ± 0.002	$\mathbf{0.489 \pm 0.001}$
Refine-CT (-28%)	0.850 ± 0.124	0.964 ± 0.640	0.794 ± 0.134	0.449 ± 0.011	0.688 ± 0.008	$\mathbf{0.518 \pm 0.006}$
PGMExplainer (-51%)	0.572 ± 0.004	1.802 ± 0.004	0.031 ± 0.001	$\mathbf{0.581 \pm 0.005}$	$\mathbf{0.480 \pm 0.003}$	0.449 ± 0.003
STFExplainer	$\mathbf{0.951 \pm 0.016}$	$\mathbf{0.657 \pm 0.207}$	0.820 ± 0.021	$\mathbf{0.556 \pm 0.013}$	$\mathbf{0.511 \pm 0.015}$	0.407 ± 0.008

Table 2: The comparison of GNN explanation metrics on two real-world datasets. Percentage (%) beside compared methods shows avg. deviation from STFExplainer on Fidelity/Recall, which is raised to compensate for large variations among datasets.

	BA-Hierarchy-motif		
	Recall	Fidelity (ACC)	$Spar$
Motif-mul (-8%)	0.557 ± 0.001	1.000	**0.597**
Motif-add (-8%)	$\mathbf{0.560 \pm 0.002}$	1.000	0.595
Motif-max (-8%)	0.557 ± 0.001	1.000	0.596
SA (-18%)	0.428 ± 0.000	1.000	0.568
DeepLift (-18%)	0.428 ± 0.000	1.000	0.568
CXPlain (-14%)	0.481 ± 0.001	1.000	0.556
GNNExplainer (-15%)	0.467 ± 0.005	1.000	0.580
PGExplainer (-9%)	0.553 ± 0.003	1.000	0.464
Refine-CT (-20%)	0.401 ± 0.202	1.000	**0.656**
PGMExplainer (-11%)	0.529 ± 0.000	1.000	0.493
STFExplainer	$\mathbf{0.666 \pm 0.002}$	1.000	0.561

Table 3: The comparison on a synthetic dataset. Motif-mul's recall is 84% of STFExplainer (-16%), while Fidelity remains 100% (-0%), resulting in an average deviation of -8%.

aggregating the saliency values of underlying motifs. Three extensions, *-add*, *-mul*, and *-max*, were introduced to compare with STFExplainer, using the addition, product, and maximum operation in generating the importance score, respectively. Though GLGExplainer can also detect motifs, it does not quantify the saliency of each motif, and cannot be directly compared wrt. multi-level explanations.

Classical GNN explanations. Typical methods in this class, i.e., *GNNExplainer*, *PGExplanier*, *PGMExplanier*, and Refine are evaluated. Their explanations at the input graph level are further hierarchically clustered by the assignment matrices to obtain the multi-level explanations for comparison. Note that Refine exploits class-aware knowledge in an additional pre-training stage using contrastive learning (CL). CL can not be universally integrated into other GNN explanation methods, also it does not work for regression tasks. Thus, we use a version of Refine without CL instead, namely *Refine-CT*, which still retains the pre-training module.

Genetic explanation of machine learning models. The explanation methods not specialized for GNNs are also included. *SA* (Pope et al. 2019) uses model gradients w.r.t. the input graph as explanatory edge importance. *DeepLIFT* (Shrikumar, Greenside, and Kundaje 2017) decomposes the prediction of a neural network onto each specific input by a backpropagating-like operation. *CXPlain* (Schwab and Karlen 2019) applies a causal objective to explain models.

Evaluation Metrics

We repeat our experiment 10 times, use grid search to find the best hyperparameters for all methods and evaluate various DNN designs compatible with our coarsening module.

Fidelity. The fidelity (Chen et al. 2018; Liang et al. 2020; Covert, Lundberg, and Lee 2020) is a widely accepted measure for GNN explanation, quantifying how well an explanation recovers original model predictions. We report the average fidelity@ρ following the practice of Refine(Wang et al. 2021). The extra parameter ρ specifies the ratio of selected edges in the explanation, ranging from $\{0.1, 0.2, \ldots, 1.0\}$ for data without ground truth. Full results can be found in Figure 3. For classification tasks, overall accuracy ACC@ρ (larger is better) and cross-entropy (CE) loss (smaller is better) are commonly used. For regression tasks, MAE (smaller is better) and R^2 (larger is better) are normally applied.

Recall. Per the successful practice previously (Ying et al. 2019), on datasets with ground truth for the explanation, e.g. the synthetic data, the recall metric can be more accurate for evaluation (larger is better). The metric is calculated by $Recall = \mathbb{E}_{\mathcal{G}}(|\mathcal{G}_s \cap \mathcal{G}_s^*|/|\mathcal{G}_s^*|)$, where $\mathcal{G}_s^*$ is the ground-truth explanatory subgraph, $\mathcal{G}_s$ is the subgraph extracted by explanation methods, $|\cdot|$ denotes the number of edges in a graph, $\mathbb{E}$ denotes the expectation across all graph data. Normally, $\mathcal{G}_s$ is composed of top-N edges recommended by the method where N is set to the edge size of $\mathcal{G}_s^*$ for calibration.

Sparsity. As an auxiliary metric, sparsity reports the required subgraph size for GNN explanations, defined as $Spar = \mathbb{E}_{\mathcal{G}}(1 - \frac{|\mathcal{G}_s|}{|\mathcal{G}|})$. When no ground truth is available, $|\mathcal{G}_s|$ is required to have the smallest number of edges for an error rate below α. The error rate measures the deviation from the original prediction when the explanatory subgraph is used as input. The metric then becomes $Spar@\alpha$, with lower α indicating higher fidelity. For classification and regression tasks, we set α to be 0.1% and 5%, respectively, under high fidelity. While high sparsity is desirable, it is not as important as fidelity or recall metrics for explanation tasks.

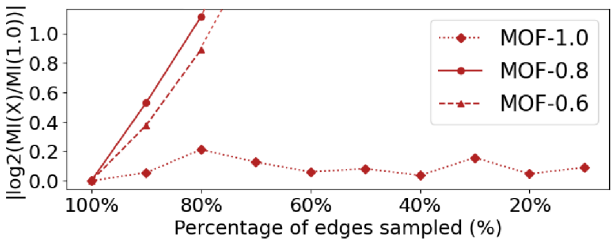

Figure 4: The variation of MI upon incomplete expansions.

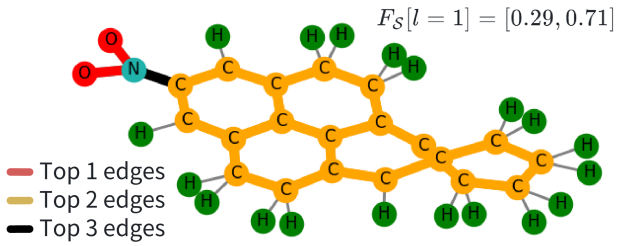

Figure 5: A case on MUTAG molecular graph explanation.

Result

STFExplainer outperforms alternatives in nearly all fidelity and recall (key metrics) in both real-world/synthetic data and classification/regression tasks (Tables 2 and 3). It ranks top-2 in 6 fidelity/recall metrics (highlighted in bold) among 11 methods, with an average deviation of 11% and 8% (percentage beside compared methods) from the best alternative on real-world and synthetic datasets, respectively. Figure 3 details the average rank of 4 fidelities of real-world data. STFExplainer becomes the top after $\rho \leq 60\%$, a common setting for the explanation. In synthetic data, all methods achieve ACC=1 but vary in explanation recall due to the inherent difference in balancing explanation fidelity and classification accuracy. Our approach prioritizes high-level interpretability, excelling in the reconstruction of motif ground truth in high-level graphs. On QMOFs data, STFExplainer slightly lags behind PGMExplainer in fidelity but is much better than other alternatives, due to the PGMExplainer's comprehensive observations through discrete perturbations.

The two-sided Wilcoxon rank sum test is also applied to compare our proposal with the leading baseline. On the MUTAG dataset, the best baseline (SA) of the real-world dataset shows no difference from ours (Fidelity (ACC): p = 0.4, Fidelity (CE): p = 1.0). However, the QMOFs task shows significant differences (p<8e-6 for Fidelity (MAE/R^2) and Fidelity (R^2)). Regarding the synthetic dataset, our method shows a noteworthy difference in recall compared to the best-performing baseline (Motif-add), with (p = 0.00794).

Experiments also validate the rationale of sufficient expansion. Figure 4, shows mutual information (confounding estimation) of low-level subgraphs with and w/o sufficient expansion. In detail, over the high-level explanatory subgraph extracted, we first obtain their full expansion in the input graph level. Then these full subgraphs are sampled using the PageRank. MI of the sampled subgraphs are calculated w.r.t the outcome variable and then compared with that of the full expansion subgraph. The absolute log is applied to the normalized MI as the sampling can either increase or decrease MI. Figure 4 shows that the MI often quickly deviates from that of the sufficient expansion result on the real-world MOFs dataset and three settings of high-level explanatory subgraph (60%∼100% of the full graph). In 2 of 3 settings, the MI changes above 2 times from the original MI when a sampling rate of 80% is applied. Therefore, to accurately represent the explanation power of high-level subgraphs, it is highly recommended to use sufficient expansion.

Case Study

The case study on the MUTAG shows the effectiveness of STFExplainer in locating explanatory patterns. Figure 5 visualizes the high-level representative explanation extracted by our method. Thin edges indicate the structure excluded while bold edges in distinct colors indicate various clusters detected. The bold red edges are rank-1 structures corresponding to the NO_2 cluster. Bold gold edges are rank-2 structures pertaining to the carbon ring cluster. Bold black edges are rank-3 structures linking NO_2 and ring together. This discovery aligns with the structure-activity relationship highlighted in MUTAG. We also calculate the saliency vector F_S for level-1: $[0.294, 0.706]$. It implies that the structure between knowledge-based clusters provides a shortcut to the mutagenic prediction than those within individual clusters.

Discussion

Model optimization. Our proposal outperforms the second-best method in only 2 out of 5 metrics, as it is built upon PGExplainer's assumption of linear independence of explained features. However, STFExplainer significantly surpasses PGE./GNNE./Refine (a class of methods) in all 3×5 metrics, which cover a wide range of applications. While it is possible to build STFExplainer over other baselines to show its universality, we have not done that due to high cost.
Synergy among multi-level explanations. As the first algorithm of its kind, we extract GNN explanations layer by layer, enabling post-processing to detect hotspots/anomalies by exploiting the commonality and discrepancy of explanations. Cross-layer optimization may complicate the solution.
Non-hierarchical adaptation. Eq. (6) will be Eq. (1) when only one layer is present, which is the objective function used by most methods. As our optimization framework is based on that of PGExplainer, the performance will also be close to PGExplainer over non-hierarchical graphs.

Conclusion

The challenge of explaining GNN decisions in real-world graphs requires multi-level explanations. Existing methods fall short due to their flat or motif-centric design. In this work, we introduce STFExplainer, which leverages sufficient expansion to generate multi-level explanations. Experimental results for both classification and regression tasks demonstrate the superiority of our approach. A case study in the context of chemical compound scenarios further confirms the utility of the multi-level explanations we extracted.

Acknowledgments

This work was supported by National Key R&D Program of China (2021YFB3500700), NSFC Grant 62172026, National Social Science Fund of China 22&ZD153, the Fundamental Research Funds for the Central Universities and SKLSDE. Lei Shi is corresponding author.

References

Azzolin, S.; Longa, A.; Barbiero, P.; Liò, P.; and Passerini, A. 2022. Global explainability of gnns via logic combination of learned concepts. *arXiv preprint arXiv:2210.07147*.

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

Barbiero, P.; Ciravegna, G.; Giannini, F.; Lió, P.; Gori, M.; and Melacci, S. 2022. Entropy-based logic explanations of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6046–6054.

Chen, J.; Song, L.; Wainwright, M.; and Jordan, M. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *International conference on machine learning*, 883–892. PMLR.

Choudhary, K.; DeCost, B.; Chen, C.; Jain, A.; Tavazza, F.; Cohn, R.; Park, C. W.; Choudhary, A.; Agrawal, A.; Billinge, S. J.; et al. 2022. Recent advances and applications of deep learning methods in materials science. *npj Computational Materials*, 8(1): 59.

Ciravegna, G.; Barbiero, P.; Giannini, F.; Gori, M.; Lió, P.; Maggini, M.; and Melacci, S. 2023. Logic explained networks. *Artificial Intelligence*, 314: 103822.

Covert, I.; Lundberg, S.; and Lee, S.-I. 2020. Feature removal is a unifying principle for model explanation methods. *arXiv preprint arXiv:2011.03623*.

Dwivedi, V. P.; Joshi, C. K.; Luu, A. T.; Laurent, T.; Bengio, Y.; and Bresson, X. 2020. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*.

Kakkad, J.; Jannu, J.; Sharma, K.; Aggarwal, C.; and Medya, S. 2023. A Survey on Explainability of Graph Neural Networks. *arXiv preprint arXiv:2306.01958*.

Kazius, J.; McGuire, R.; and Bursi, R. 2005. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of medicinal chemistry*, 48(1): 312–320.

Kengkanna, A.; and Ohue, M. 2023. Enhancing Model Learning and Interpretation Using Multiple Molecular Graph Representations for Compound Property and Activity Prediction. *arXiv preprint arXiv:2304.06253*.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Liang, J.; Bai, B.; Cao, Y.; Bai, K.; and Wang, F. 2020. Adversarial infidelity learning for model interpretation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 286–296.

Luo, D.; Cheng, W.; Xu, D.; Yu, W.; Zong, B.; Chen, H.; and Zhang, X. 2020. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33: 19620–19631.

Pope, P. E.; Kolouri, S.; Rostami, M.; Martin, C. E.; and Hoffmann, H. 2019. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10772–10781.

Reiser, P.; Neubert, M.; Eberhard, A.; Torresi, L.; Zhou, C.; Shao, C.; Metni, H.; van Hoesel, C.; Schopmans, H.; Sommer, T.; et al. 2022. Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1): 93.

Riesen, K.; and Bunke, H. 2008. IAM graph database repository for graph based pattern recognition and machine learning. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, SSPR & SPR 2008, Orlando, USA, December 4-6, 2008. Proceedings*, 287–297. Springer.

Rosen, A. S.; Fung, V.; Huck, P.; O'Donnell, C. T.; Horton, M. K.; Truhlar, D. G.; Persson, K. A.; Notestein, J. M.; and Snurr, R. Q. 2022. High-throughput predictions of metal–organic framework electronic properties: theoretical challenges, graph neural networks, and data exploration. *npj Computational Materials*, 8(1): 112.

Rosen, A. S.; Iyer, S. M.; Ray, D.; Yao, Z.; Aspuru-Guzik, A.; Gagliardi, L.; Notestein, J. M.; and Snurr, R. Q. 2021. Machine learning the quantum-chemical properties of metal–organic frameworks for accelerated materials discovery. *Matter*, 4(5): 1578–1597.

Schnake, T.; Eberle, O.; Lederer, J.; Nakajima, S.; Schütt, K. T.; Müller, K.-R.; and Montavon, G. 2021. Higher-order explanations of graph neural networks via relevant walks. *IEEE transactions on pattern analysis and machine intelligence*, 44(11): 7581–7596.

Schütt, K.; Kindermans, P.-J.; Sauceda Felix, H. E.; Chmiela, S.; Tkatchenko, A.; and Müller, K.-R. 2017. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30.

Schwab, P.; and Karlen, W. 2019. Cxplain: Causal explanations for model interpretation under uncertainty. *Advances in neural information processing systems*, 32.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*, 3145–3153. PMLR.

Tang, H.; Ma, G.; He, L.; Huang, H.; and Zhan, L. 2021. Commpool: An interpretable graph pooling framework for hierarchical graph representation learning. *Neural Networks*, 143: 669–677.

Vu, M.; and Thai, M. T. 2020. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *Advances in neural information processing systems*, 33: 12225–12235.

Wang, H.; Guo, X.; Deng, Z.-H.; and Lu, Y. 2022. Rethinking minimal sufficient representation in contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16041–16050.

Wang, X.; Wu, Y.; Zhang, A.; He, X.; and Chua, T.-S. 2021. Towards multi-grained explainability for graph neural networks. *Advances in Neural Information Processing Systems*, 34: 18446–18458.

Wu, B.; Chao, K.-M.; and Li, Y. 2023. DualFraud: Dual-Target Fraud Detection and Explanation in Supply Chain Finance Across Heterogeneous Graphs. In *International Conference on Database Systems for Advanced Applications*, 370–379. Springer.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.

Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; and Leskovec, J. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32.

Ying, Z.; You, J.; Morris, C.; Ren, X.; Hamilton, W.; and Leskovec, J. 2018. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31.

Yu, Z.; and Gao, H. 2022. Motifexplainer: a motif-based graph neural network explainer. *arXiv preprint arXiv:2202.00519*.

Yuan, H.; Tang, J.; Hu, X.; and Ji, S. 2020. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 430–438.

Yuan, H.; Yu, H.; Gui, S.; and Ji, S. 2022. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5): 5782–5799.