User-friendly, Interactive, and Configurable Explanations for Graph Neural Networks with Graph Views

Tingyang Chen
chenty@zju.edu.cn
Zhejiang University
Hangzhou, China

Dazhuo Qiu
dazhuoq@cs.aau.dk
Aalborg University
Aalborg, Denmark

Yinghui Wu
yxw1650@case.edu
Case Western Reserve University
Cleveland, Ohio, USA

Arijit Khan
arijitk@cs.aau.dk
Aalborg University
Aalborg, Denmark

Xiangyu Ke
xiangyu.ke@zju.edu.cn
Zhejiang University
Hangzhou, China

Yunjun Gao
gaoyj@zju.edu.cn
Zhejiang University
Hangzhou, China

ABSTRACT

Explaining the behavior of graph neural networks (GNNs) has become critical due to their "black-box" nature, especially in the context of analytical tasks such as graph classification. Current approaches are limited to providing explanations for individual instances or specific class labels and may return large explanation structures that are hard to access, nor directly queryable. In this paper, we present GVEX [1] (**G**raph **V**iews for GNN **EX**planation) – our system developed to offer *user-friendly*, *interactive*, and *configurable* explanations for GNNs based on *graph views*.

GVEX provides a configuration component to enable users to easily select a desired number of important nodes from different classes, thereby generating explanations tailored to multiple classes of interest. Furthermore, GVEX generates high-quality explanation subgraphs by identifying important nodes exploiting factual and counterfactual properties and by computing their aggregated influence on the remaining nodes following the GNN message passing paradigm. Lastly, GVEX performs a summarize step on top of lower-tier explanation structures to generate higher-tier graph patterns that offer direct access for users with (domain-aware) queries. Our demonstration will highlight **(1)** a novel two-tier explanation structure called *explanation views*, consisting of graph patterns and a set of explanation subgraphs, which provide high-quality explanations for GNNs; **(2)** the system's intuitive GUI facilitates user interaction to configure personalized settings, e.g., classes of interest and explanation size, and compare with other explanation algorithms; **(3)** GVEX generates queryable explanations, making it easy for human experts to access and inspect with domain knowledge. Our demonstration video is at: **https://youtu.be/q9d7ldulIuQ**.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Information systems** → **Graph-based database models**.

SIGMOD-Companion '24, June 9–15, 2024, Santiago, Chile.

ACM ISBN 979-8-4007-0421-5/24/06
https://doi.org/10.1145/3626246.3654735

KEYWORDS

Graph neural networks, Explainable AI, Graph views

ACM Reference Format:
Tingyang Chen, Dazhuo Qiu, Yinghui Wu, Arijit Khan, Xiangyu Ke, and Yunjun Gao. 2024. User-friendly, Interactive, and Configurable Explanations for Graph Neural Networks with Graph Views. In *Companion of the 2024 International Conference on Management of Data (SIGMOD-Companion '24), June 9–15, 2024, Santiago, Chile.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3626246.3654735

1 INTRODUCTION

Graph classification is essential in a variety of real-world applications including drug discovery, text classification, and recommender systems. Graph neural networks (GNNs) have demonstrated remarkable potential in graph classification in many real-world domains. However, generating high-quality explanations for GNNs is still a challenging task, which prevents users from comprehending the GNN mechanism. Given a GNN classifier $\mathcal{M}$ and a graph dataset $\mathcal{G}$, one wants to discover a critical fraction of $\mathcal{G}$ that is responsible for the occurrence of specific class labels of interest, assigned by the GNN $\mathcal{M}$ over $\mathcal{G}$. Such explanations should **(1)** capture both important features and structural information; **(2)** be *queryable*, making it easy for human experts to access and inspect with domain knowledge; **(3)** be *configurable* to enable users freely obtaining explanations tailored to classes of interest and of specific sizes. These explanations foster a more intuitive grasp of the model's functionality and its connection to human expertise.

We demonstrate the benefits of our GNN explaining system, GVEX [1], considering the classification of chemical compounds into "Mutagens" vs. "Nonmutagens". *Mutagenicity* signifies a chemical compound's capacity to induce mutations. A GNN classifies four graphs in Figure 1 into two groups with class labels, mutagens and nonmutagens, respectively. A chemical analyst aims to investigate the specific reasons "why" certain graphs are identified as mutagens, "what" crucial substructures could contribute to these findings; however, the large quantity of chemical graphs makes it difficult to directly analyze the GNN outcomes. We illustrate our two-tier explanation structures referred to as *explanation views*, consisting of subgraphs and patterns, and are analogous to *graph views* [2]. **(1)** Lower-tier subgraphs explain the GNN's classification mechanism and the crucial molecular substructures contributing to the outcomes. They serve as both factual (that preserves the result of

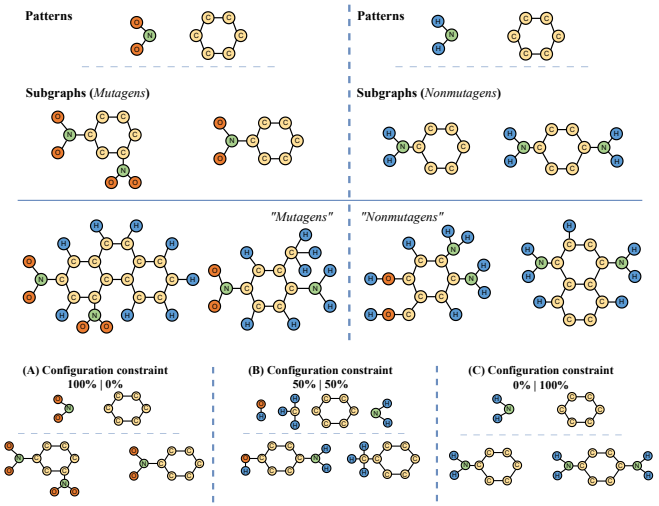

Figure 1: Our two-tier explanations for GNN-based drug classification. Subgraphs represent "lower-tier" structures. Patterns represent "higher-tier" queryable structures. The three configuration scenarios at the bottom indicate whether the user prefers only one class or is interested in the nature of both classes.

classification) and counterfactual explanations (which flips the result if removed). Additionally, we exploit the influence maximization principle following the GNN message passing paradigm to quantify the effectiveness of subgraphs. Next, the higher-tier patterns summarize the details of subgraphs through constrained pattern mining on lower-tier subgraphs for efficient search and comparison of these subgraphs. **(2)** Such patterns can be suggested to the analysts for further inspection, or be conveniently issued as graph queries for downstream analysis, e.g., "*which patterns occur in mutagens the most frequently?*", "*which patterns occur in both mutagens and nonmutagens?*" **(3)** Users have the flexibility to configure different scenarios based on their preference for a single class or interest in the characteristics of both classes, e.g., they want to understand representative substructures that are discriminative enough to distinguish mutagens and nonmutagens. For the mutagens label, NO_2 is a real toxicophore (i.e., a pattern that indicates an increased potential for mutagenicity) as verified by domain experts [3]. In contrast, users find NH_2 as a common pattern in the nomutagens label. When users attempt to understand both classes, GVEX presents the salient patterns of both classes, shedding light on important patterns that occur in this molecular dataset.

Previous GNN explaining tools [6] do not explicitly support configurable and queryable explanation structures and are not optimized to generate explanations for user-specific class labels of interest, nor of specific sizes. In this demonstration, we will showcase how we address the challenges associated with generating high-quality and user-friendly explanations for GNNs. We achieve this through our end-to-end GVEX system, which offers users convenient interaction features. By incorporating graph datasets and a user-defined configuration as inputs from the frontend's interaction components, GVEX automatically loads the data into the backend. It initially presents the visualization of the graph datasets along with their statistical information. Subsequently, it generates explanation views by invoking the core algorithm (§3). Additionally, GVEX provides

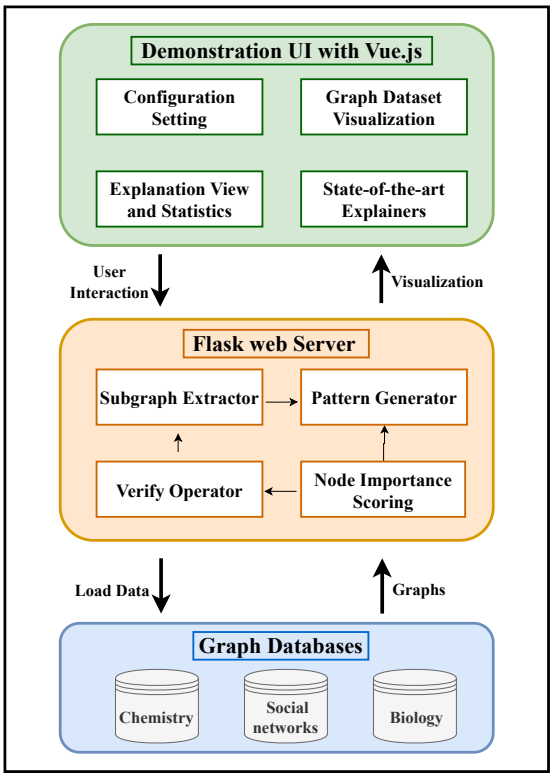

Figure 2: The architecture of the GVEX system.

statistical information on the explanation view and presents it graphically, facilitating the analysis of real-world scenarios. Moreover, our platform incorporates state-of-the-art GNN explainers for users to evaluate and compare explanation results. To the best of our knowledge, this is *the first demonstration proposal on generating user-friendly, interactive, and configurable explanations for* GNNs.

2 SYSTEM OVERVIEW

We provide a comprehensive overview of our GVEX system. Figure 2 presents the architecture of GVEX and the interaction between its frontend and backend. The frontend facilitates user interaction for configuration settings and provides visualization of the input graph data, output explanation views with statistical information, as well as generated explanations using state-of-the-art algorithms. The backend is responsible for node importance scoring, the verify operator, subgraph extractor, and the pattern generator. We implement the frontend using Vue.js (https://vuejs.org/), while the backend is developed with Flask – a micro web framework written in Python.

Node Importance Scoring module receives input graphs from the graph database and employs a random walk algorithm on the graphs to compute two explainability scores, *influence* and *neighborhood diversity*, following feature sensitivity and influence analysis in GNNs [9]. The overall importance of a node is derived by aggregating these explainability scores [1]. This process serves as a one-time precomputation for all explainability queries and configurations.

Verify module filters subgraphs constructed by important nodes that do not satisfy counterfactual causality. An explanation subgraph

G_s, which is a subgraph of G, satisfies counterfactual causality when it clarifies "why" $\mathcal{M}(G) = l$, that is, $\mathcal{M}(G_s) = l$ and $\mathcal{M}(G \setminus G_s) \neq l$.

Subgraph Extractor module extracts lower-tier explanation subgraphs constructed by important nodes following the greedy influence maximization approach, which also satisfy verification by invoking the *Verify* module.

Pattern Generator module exploits constrained graph pattern mining to efficiently explore the higher-tier patterns. It extracts a set of candidate patterns from the explanation subgraphs, and then adopts a greedy strategy to dynamically select patterns that maximize a gain while covering all nodes in explanation subgraphs.

Visualization Displayer is a user-friendly module, which facilitates interactive visualization. **(1)** It includes an interface that allows users to easily configure explanations based on their preferences on class labels of interest and explanation sizes. **(2)** It visualizes input graph datasets, output explanation views, and statistical information. **(3)** It displays explanations generated by state-of-the-art algorithms to facilitate comparison.

3 CORE ALGORITHM

We next describe the key technique, *Explain-and-Summarize* in our algorithm, which generates good-quality explanation views. For details, we refer to our research paper [1]. Our explanation view $\mathcal{G}^l_{\mathcal{V}} = (\mathcal{P}^l, \mathcal{G}^l_s)$ consists of "lower-tier" explanation subgraphs $\mathcal{G}^l_s$ and "higher-tier" explanation patterns $\mathcal{P}^l$ for a class label l.

Explain Step selects high-quality nodes to induce "lower-tier" explanation subgraphs for GNN-classifer $\mathcal{M}$ that **(1)** pass the verification ensuring both factual and counterfactual properties; **(2)** maximize the overall explainability score for the explanation view; and **(3)** satisfy the node size constraints in the configuration parameters. Recall that the explainability score contains *influence* and *neighorhood diversity*. The *explainability* of an explanation view $\mathcal{G}^l_{\mathcal{V}} = (\mathcal{P}^l, \mathcal{G}^l_s)$ for $\mathcal{M}$ over $\mathcal{G}^l$ is quantified as:

$$f(\mathcal{G}^l_{\mathcal{V}}) = \sum_{G_s \in \mathcal{G}^l_s} \frac{I(V_s) + \gamma D(V_s)}{|V|} \tag{1}$$

where **(i)** V_s is the node set of an explanation subgraph G_s of G ($G_s \in \mathcal{G}^l_s$, and $G \in \mathcal{G}$), and V is the node set of G ($V_s \subseteq V$); **(ii)** $I(V_s)$ is a *feature influence* function that quantifies the influence of features of the node set $\mathcal{V}_s$ via feature propagation in the inference process of $\mathcal{M}$, and **(iii)** $D(V_s)$ is a diversity measure to capture influence maximization. Here, a weight $\gamma \in [0, 1]$ is introduced to balance between feature influence and diversity.

Summarize Step produces, as "higher-tier" structure, a set of graph patterns $\mathcal{P}$ which cover the nodes of the explanation subgraphs $\mathcal{G}^l_s$. Meanwhile, it is desirable for $\mathcal{P}$ to cover the edge set of $\mathcal{G}^l_s$ as much as possible. Given a pattern $P \in \mathcal{P}$ and graphs $\mathcal{G}^l_s$ with node set V_S and edge set E_S, we denote the nodes and edges in $\mathcal{G}^l_s$ it covers as P_{V_S} and P_{E_S}, respectively. Let each P be "penalized" by a normalized weight (as the Jaccard distance) between E_S and P_{E_S}, i.e., $w(P) = 1 - \frac{|P_{E_S}|}{|E_S|}$ (note $P_{E_S} \subseteq E_S$). The above requirements are formulated as an optimization problem:

- **Input**: explanation subgraphs $\mathcal{G}^l_s$;
- **Output**: a pattern set $\mathcal{P}^l$, such that (1) $\bigcup_{P \in \mathcal{P}^l}(P_{V_S}) = V_S$ and (2) $\mathcal{P}^l = \arg\min \sum_{P \in \mathcal{P}^l} w(P)$.

The summarize step solves the above problem by conducting a constrained pattern mining on V_S and their corresponding induced explanation subgraphs $\mathcal{G}^l_s$. It first exploits the minimum description length (MDL) principle to iteratively generate a set of pattern candidates, and subsequently adopts a greedy strategy to select a pattern P^* that maximizes a gain ascertained by covered nodes $\mathcal{P}^*_{V_S}$ in V_S with the smallest weight. $\mathcal{P}^l$ is enlarged with $\mathcal{P}^*$ accordingly. It can be implemented by invoking some scalable pattern mining algorithms, e.g., gSpan [4]. Post the selection of the currently optimal patterns, the matched nodes in V_S are reduced; and the weights of the patterns are updated accordingly. This allows to gradually acquire the final explanation view and reduce the edges "missed" by $\mathcal{P}^l$.

The above algorithm requires generating all explanation subgraphs to compute explanation views. We also design an anytime method, StreamGVEX to *incrementally* maintain explanation views by scanning over $\mathcal{G}$ as nodes stream and their *parallel* versions [1].

4 DEMONSTRATION SCENARIO

We describe our demonstration scenarios using Figure 3, which is a snapshot of the GVEX frontend. Our web app, available at **https://github.com/TiyCHEN/GVEX-DEMO**, accompanies several real-world datasets from various domains [1] and popular GNN models. To illustrate the functionality, we present a workflow with the widely-used Mutagenicity (MUT) dataset [3]. MUT is a molecular dataset for binary classification tasks. Each graph within this dataset represents a chemical compound with nodes denoting atoms and undirected edges symbolizing bonds, and the one-hot node feature denotes the atom type. An example workflow to support effective analyses on MUT is as follows.

Step 1. Initial Selection: The users first select an existing dataset or upload their own via buttons found in the "Graph Dataset" panel. The users can click the "Show" button to visualize the chosen graph dataset (e.g., MUT in Figure 3). Once clicked, molecular graphs are presented in the "Input Graphs" panel, with distinct colors assigned to different node types (i.e., C, H, O, N, etc.). The dataset's information (Name, Nodes, Edges, Classes, Domain) is conveniently displayed in the right-side bar of this panel. Furthermore, the users have the flexibility to navigate and zoom on the graphs , allowing a closer inspection of the specific structures within the selected dataset. Also, the users can choose whether to use a static (ApproxGVEX) or stream (StreamGVEX) algorithm.

Step 2. Interactive Configuration: GVEX offers a configuration component that allows users to tailor algorithm settings to their preferences. This configuration interface, located in the bottom left corner of Figure 3, allows the users to fine-tune various parameters. **(1)** The "budget" parameter determines the desired size in the explanation views; **(2)** the "classes" parameter permits the user-specific class labels of interest and determines the explanation ratios between different labels; **(3)** the "influence" and "diversity" parameters control the threshold for feature influence and the scope of neighborhood diversity; **(4)** the "trade-off" parameter gauges the importance assigned to neighborhood diversity; **(5)** the self-regulating progress bar is designed for StreamGVEX to investigate and ad-hocly query for specific explanation structures. Specifically, users can incrementally maintain explanation views by setting a value between 0% to 100% of input graphs with the the self-regulating progress bar.

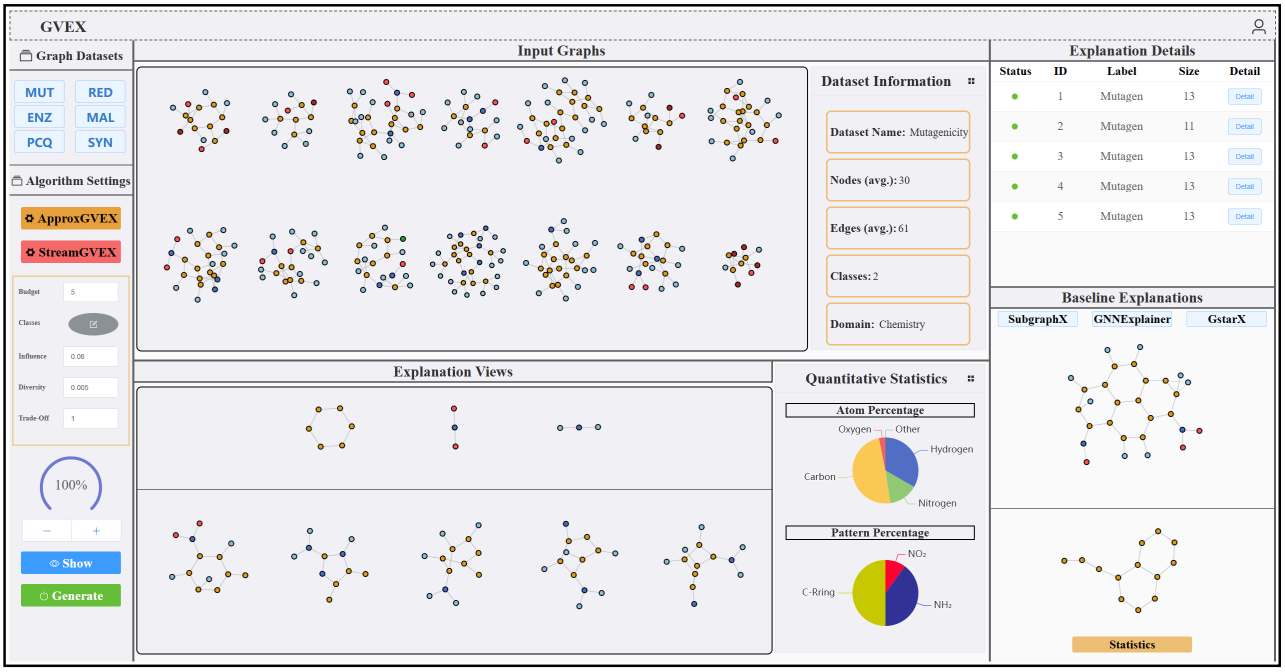

Figure 3: GVEX **frontend: left part – configuration panel, upper-center part – dataset displayer, lower-center part – explanation views and statistics, upper-right part – explanation details, and lower-right part – baseline explanation results.**

Step 3. Visualize Explanation Views and Statistics: When a user clicks the "Generate" button, GVEX executes the selected explanation algorithm in the backend. Once the process is complete, GVEX adds the results to the "Explanation Views" and "Explanation Details" panels. Recall that explanation views consist of graph patterns and a set of explanation subgraphs. The "lower-tier" explanation subgraphs explain the GNN w.r.t. the labels of interest, satisfying both factual and counterfactual properties. The "higher-tier" patterns serve as a concise summary to allow easy access, query, and inspection of the classification and explanation results. The details for explanation subgraphs are displayed in the "Explanation Details" panel. Furthermore, statistical information related to explanation views can be suggested to analysts for further inspection, or conveniently be utilized as graph queries for downstream analysis, e.g., "*which patterns occur only in mutagens, but not in nonmutagens?*".

Step 4. Visualize State-of-the-art Explanations: The GVEX system also supports recent GNN explanation algorithms, including SubgraphX [7], GNNExplainer [5], and GstarX [8], accessible through the "Baseline Explanations" panel. We randomly sample a graph from the graph dataset and after clicking a designated button, the corresponding algorithm is invoked to generate an explanation for it. This feature allows users to compare explanation subgraphs identified for each mutagen compound by different explainers. We also provide quantitative statistics of these baseline explanations.

5 CONCLUSION

We developed the GVEX system – a novel, user-friendly, interactive, and graph view-based two-tier structure to explain graph classification with GNNs. Our demonstration offers a visualization and configuration platform for human experts to access and inspect explanations for class labels of interest. The GVEX system assists to configure algorithm settings, and to compare with state-of-the-art explainers, making it flexible and usable to the audiences.

6 ACKNOWLEDGEMENT

Tingyang Chen and Dazhuo Qiu contributed equally to this research. Tingyang Chen, Xiangyu Ke, and Yunjun Gao are supported in part by the NSFC under Grants No. (62025206, U23A20296) and Yongjiang Talent Introduction Programme (2022A-237-G). Dazhuo Qiu and Arijit Khan acknowledge support from the Novo Nordisk Foundation grant NNF22OC0072415. Yinghui Wu is supported in part by NSF under CNS-1932574, ECCS-1933279, CNS-2028748 and OAC-2104007. Xiangyu Ke is the corresponding author.

REFERENCES

[1] Tingyang Chen, Dazhuo Qiu, Yinghui Wu, Arijit Khan, Xiangyu Ke, and Yunjun Gao. 2024. View-based explanations for graph neural networks. In *SIGMOD*.
[2] Wenfei Fan, Xin Wang, and Yinghui Wu. 2014. Answering graph pattern queries using views. In *ICDE*.
[3] Jeroen Kazius, Ross McGuire, and Roberta Bursi. 2005. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of Medicinal Chemistry* 48, 1 (2005), 312–320.
[4] Xifeng Yan and Jiawei Han. 2002. gspan: Graph-based substructure pattern mining. In *ICDM*.
[5] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*.
[6] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2023. Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 5 (2023), 5782–5799.
[7] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. In *ICML*.
[8] Shichang Zhang, Yozen Liu, Neil Shah, and Yizhou Sun. 2022. GStarX: Explaining graph neural networks with structure-aware cooperative games. In *NeurIPS*.
[9] Wentao Zhang, Zhi Yang, Yexin Wang, Yu Shen, Yang Li, Liang Wang, and Bin Cui. 2021. Grain: Improving data efficiency of graph neural networks via diversified influence maximization. *Proc. VLDB Endow.* 14, 11 (2021), 2473–2482.