ChemGraph Explainer: A Graphical User Interface for Explaining Predictions of Graph Neural Networks in Chemistry

Ali Can Kara[1], Debanjan Rana[2], Frank Glorius[2], and Xiaoyi Jiang[1]

[1] Faculty of Mathematics and Computer Science, University of Münster, Münster, Germany {akara2,xjiang}@uni-muenster.de, alican.kara@meb.k12.tr
[2] Organic Chemistry Institute, University of Münster, Münster, Germany {debanjan.rana,glorius}@uni-muenster.de

Abstract. Since Graph Neural Networks (GNNs) made a big impact on graph structured datasets, they are widely utilized in the field of chemistry. However, the reasons behind the prediction of GNNs are not always obvious, so they are considered as black-box models. In this paper, we introduce a graphical user interface (GUI) which can be used for explaining the predictions of GNNs. We aim to integrate our GUI into the user's research directly to make the predictions of GNNs more understandable in both classification and regression tasks. Furthermore, we offer the option to use the built-in GNN models to train custom datasets directly. Additionally, the system incorporates several explainable artificial intelligence (XAI) techniques, and also allows users to assess the accuracy of explanation findings using various assessment metrics and thus to compare the explanation outcomes. Using the well-known datasets in the field, this tool can also be used for education purposes. The interface provides a comprehensive platform for examining and interpreting the predictions provided by the GNNs and merging several GNN models with XAI approaches. This will facilitate a deeper understanding and possibly lead to new discoveries in researchers' respective domains in understanding the underlying elements that influence the model's explainability. The code is made publicly available at https://github.com/ChemGraphExplainer/ChemGraphExplainer.

Keywords: Artificial Intelligence · Graph Neural Networks · Explainable Artificial Intelligence · Chemistry · Graphical User Interface

1 Introduction

Graph Neural Networks (GNNs) have seen a sharp rise in usage in the last several years for practical uses, including fraud detection [13], drug design [16, 24, 27, 1, 29], healthcare [40, 41], and recommender systems [9]. Several graph-related tasks, such as node [10, 14], link [5, 36], and graph classifications [37], have been thoroughly studied. As GNNs become more and more common, more attention is being given to their explanation. Due to the wide use of GNNs in different fields

Explainable Artificial Intelligence (XAI) has become increasingly important in the context of GNNs as its applications in delicate and important fields increase. XAI is a research area in which methods are developed that enable human users to understand and trust the results and outputs generated by machine learning algorithms. This explanation is not directly achievable in traditional AI methods [2, 8, 20]. The goal of XAI for GNNs is to increase network transparency so that users can comprehend the reasoning behind a given prediction.

While the majority of XAI researches has been investigated for classification problems [3, 19], regression problems have received little attention despite having a very important place in machine learning [12]. Generally, the XAI methods developed for classification problems are not appropriate to apply in studies on regression problems. Only recently, such methods have been adapted to deal with regression problems [12].

There are some libraries that enable the use of GNN models and the evaluation of predictions made with XAI methods. Libraries frequently used in the field of chemistry include DIG [15], DeepChem [18], and PyTorch Geometric [7]. However, these libraries do not provide a user interface and do not offer options suitable for using explanation methods for regression problems. We aim to develop a user-friendly library with the user interface by using and developing the DIG library. The reason we use the DIG library is that it comes ready to integrate some XAI methods directly into GNN models. In addition, our user interface, named ChemGraph Explainer, offers the opportunity to easily evaluate and compare the decisions made by the GNN model with different XAI methods.

The remainder of the paper is structured as follows. In Section 2 general information about GNNs will be given and the GNN models included in ChemGraph Explainer will be explained. Additionally, we also present information about the categorization of XAI methods, the methods included in the user interface, and the evaluation metrics used to evaluate XAI methods. In Section 3, detailed information will be given about ChemGraph Explainer and its use, and also our contribution will be mentioned. In Section 4 we show three case studies using ChemGraph Explainer. Finally, Section 5 concludes the paper with some discussion.

2 Methods

In this section, we will give technical information about some GNN models and also explanation methods included in the user interface. The GNNs and explanation methods we will talk about in this section are the models and methods that are available in the DIG library or that we are adding to the DIG library.

2.1 Graph Neural Networks Models

There are many studies carried out with deep learning methods in Euclidean space such as natural language processing, image recognition or video processing.

In addition, recently there has been an increasing interest in examining graph data with deep learning approaches. However, significant difficulties have arisen due to the complexity of graph-structured data. For example, graph data has a complex structure, causing difficulties in machine learning algorithms [30].

Graph Convolutional Networks (GCNs) [38] are algorithms that work with graph structures to simulate the relationships between edges, nodes, or graphs. GCNs compile data from a node's neighbors to update its representation. This procedure transforms and gathers characteristics from a node's local neighborhood in an attempt to capture the graph structure. Using graph signal processing techniques and transformations over the graph Laplacian, spectral approaches carry out convolution operations in the spectrum domain. In contrast, spatial techniques use the structure of the graph to directly aggregate characteristics from nearby nodes.

Graph Isomorphism Networks (GIN) [31] are developed to improve the representation capacity of graph topology. By simulating the strength of the Weisfeiler-Lehman (WL) graph isomorphism test, GIN seeks to optimize the discriminative power of graph architectures. As a result of this aggregation process, GIN changes the representation vector of the node by combining the features of its neighbors. This process is comparable to the WL test's labeling and summarizing steps. In order to make sure that various node neighbors are transferred to diverse representations, GIN uses injective functions to aggregate neighbor features. In addition, GIN aggregates neighbor features using multi-layer perceptrons.

Graph Attention Network (GAT) [25] is a neural network design that uses masked self-attention layers to overcome the drawbacks of existing graph convolution models. GATs are able to pay attention to the properties of these nodes and provide varying weights to different nodes within their neighborhoods.

2.2 XAI Methods

Explanation techniques can be grouped in two different ways. The first of these is based on the application period of the explanation method and the type of AI model to which it can be applied, see Figure 1. According to their explanation periods, they are classified into: transparent and post-hoc. Transparent methods are methods, in which the internal structure of the model and the decision-making process can be directly understood. K-nearest Neighbor, Linear-logistic regression, and decision Trees are examples of transparent methods [8]. Post-hoc methods are explanation methods applied to complex models that can be applied after the training of the AI model is completed. Post-hoc methods are further categorized into two as model-agnostic and model-specific, depending on whether they can be applied to any AI model. Model-agnostic methods are explanation

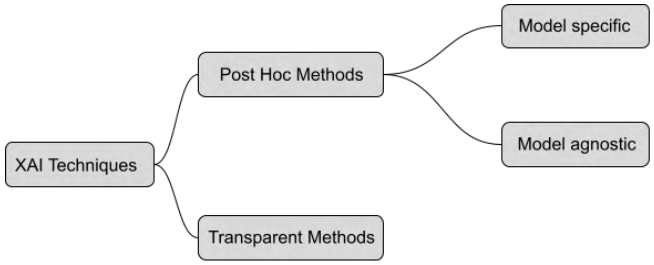

Fig. 1. Categorization of XAI techniques according to their application stages (during training or after training) and their applicability to models [8].

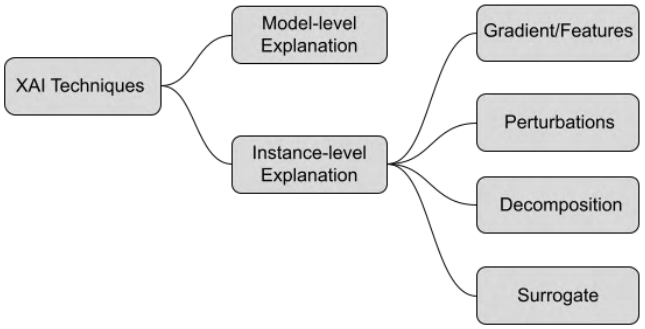

Fig. 2. Categorization of XAI techniques according to their application methods [15].

methods that can be applied to any machine learning method without being specific to a particular model. Model-specific methods are methods that can be used on certain model types only.

Alternatively, the categorization can also be made according to the application methods of explanation methods, see Figure 2. Explanation techniques are categorized into two main ways to evaluate the model. Model-level explanation methods aim to explain the model in general, independent of a specific input [35]. Since model-level explanation methods are very unexplored methods, only instance-level explanation methods have been added to ChemGraph Explainer so far, therefore only instance-level explanation methods are discussed in the following part.

Gradients-Based Methods [15] are employed for deep learning models operating on graph data to improve the explainability of these models. The base of these techniques is the computation of gradients by back propagation, which establishes the significance of input information about the target prediction. Here, the main idea is to use gradients or hidden feature maps to quantify the sensitivity between the target prediction and the input characteristics.

Gradient-weighted Class Activation Mapping (Grad-CAM) [22] is a method that calculates the prediction gradients related to the node embeddings in the final layer of the model. These gradients are averaged to obtain a weight for each feature map, and these weights are used to highlight important nodes in the feature maps. Thus, a heatmap is created that shows the nodes with the most influence on the prediction.

Perturbation-Based Methods [15] examine how the output predictions alter in response to various input perturbations. The main notion of perturbation-based approaches is to assess how the model's predictions are affected by keeping or changing specific input properties. If an important part of the input for the prediction is changed, a dramatic change in the prediction is observed. In order to explain deep neural networks the perturbation-based methods are generally used [6, 34].

GNNExplainer [32] is a model-agnostic method that aims to generate understandable explanations for predictions. It finds important node properties and subgraph structures that have a big impact on the predictions. GNNExplainer maximizes the mutual information between the distribution of possible subgraph structures and the GNN's predictions by rephrasing the explanation generation as an optimization task. This approach is extremely flexible and may be used for any GNN model as well as any graph-based task, including graph classification, link prediction, and node classification. GNNExplainer considerably surpasses baseline techniques in producing succinct and consistent explanations, improving model transparency, interpretability, and confidence by displaying pertinent graph structures and offering insights into model flaws.

Decomposition Methods [15] divide the initial prediction scores into many components, which are subsequently interpreted as the importance scores of matching input features, in an attempt to explain the predictions of deep graph models. By examining model parameters, this technique reveals the connections between input space features and output predictions.

Graph Neural Network Layer-wise Relevance Propagation (GNN-LRP) [21] is a method that offers higher-order explanations for the predictions made by GNNs. Using a hierarchical attribution approach, it breaks down the prediction into relevance scores for various network walks at each stage by applying proven technique named Layer-wise Relevance Propagation (LRP) [4]. This method finds sets of edges that together contribute to the prediction, capturing the intricate relationships between the network's layers. Graph Neural Network - Gradient Integration (GNN-GI) [21] uses the same working logic as the GNN-LRP method, but is a simplified version of it.

Deep Learning Important FeaTures (DeepLift) [23] sets a reference value and this reference value is used to compare with the normal operation of the machine learning model. The input value is compared with the reference value and the working logic of the model is tried to be calculated.

Surrogate Methods [15] are described as using a simple surrogate model to approximate the AI model's predictions for neighboring regions of the input. The results obtained from the interpretable surrogate model are applied to explain the original prediction. Some methods such as GraphLime [11], PGM-Explainer [26], RelEx [39] have been proposed to explain deep graph models.

2.3 Evaluation Metrics

In this section, some information will be given about evaluation metrics to evaluate explanation methods. When the predictions made by AI models are explained, it is necessary to evaluate whether the explanation made is logical or not. However, there may be cases where researchers cannot visually detect how meaningful explanation methods are or how accurate results XAI methods show. For this reason, some evaluation criteria are needed in XAI methods. In this section, the metrics used to evaluate XAI methods will be discussed.

Fidelity+ [17] assesses how well a model can recognize important features. The basic idea is that the model's predictions need to drastically drop if the elements the model deems critical are eliminated. Stated differently, a model has correctly identified the crucial features if it recognizes some nodes or edges as significant and removing these features results in a lower prediction accuracy.

$$\text{Fidelity}+_{\text{acc}} \;=\; \frac{1}{N}\sum_{i=1}^{N}\left(\mathbb{1}(\hat{y}_i = y_i) - \mathbb{1}(\hat{y}_{1-m_i} = y_i)\right) \tag{1}$$

In Equation (1), N is the total number of samples evaluated, $\hat{y}_i$ is the GNN's prediction for the i^{th} graph, y_i is the true label of the molecule, $\hat{y}_{1-m_i}$ is the prediction made by the GNN model after removing important features, and the indicator function $\mathbb{1}$ returns 1 if the condition is true and 0 otherwise.

Fidelity- [33] assesses how important feature preservation affects model predictions. It gauges the degree to which the model's predictions hold up when the important features are kept and the rest are eliminated. The truly significant information has been accurately captured by the model if its predictions remain mostly unchanged while the features that it considers vital are kept.

$$\text{Fidelity-}_{\text{acc}} \;=\; \frac{1}{N}\sum_{i=1}^{N}\left(\mathbb{1}(\hat{y}_i = y_i) - \mathbb{1}(\hat{y}_{m_i} = y_i)\right) \tag{2}$$

In Equation (2), $\hat{y}_{m_i}$ is the prediction made by the GNN model when keeping important features only.

Sparsity [17] assesses the degree of sparsity in the explanation findings. Effective explanations should focus on the most significant details while ignoring the less significant ones. This measure evaluates the importance of the features that the explanation method chooses and establishes how few features it chooses.

$$\text{Sparsity} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{|m_i|}{|M_i|}\right) \tag{3}$$

In Equation (3), m_i indicates the number of important features in the i^{th} sample and M_i represents the total number of features in the specific sample. In ChemGraph Explainer, we determine the sparsity value as an external input. In this way, the user will be able to calculate fidelity+ and fidelity- values by determining how many features are important in the input.

3 Graphical User Interface ChemGraph Explainer

In this section, some information will be given about the user interface we are developing. ChemGraph Explainer is a user interface developed based on the DIG library. Our main motivation is to develop a user-friendly user interface for users working with GNN models in the field of chemistry to evaluate the results obtained by GNN models and to enable them to see the reasons for the predictions of these models. There are different libraries that can be used in this field. Our current implementation is based on DIG [17]. This library distinguishes from others in that different XAI methods useful for GNN models are available in the library. Additionally, the DIG library is ready to use with 2 GCN models (with 2-layers and 3-layers, respectively) and GIN. We are currently using DIG library methods for explanations, but we also want to integrate causality methods into the system. We have designed the user interface as well as the visualization of the results. We are also continuously adding various GNNs such as GAT to the system.

The appearance of ChemGraph Explainer can be seen in Figure 3. It has three main building elements: datasets, GNN models, and explanation methods. A number of pre-specified datasets are available to be selected for use and some GNN models and explanation methods are made available or are still being worked on. Given these building elements, ChemGraph Explainer is beneficial in different application scenarios. Some examples are:

- Use of a pre-specified dataset and a given GNN model. This simple use case enables researchers to get familiar with the tool before working on their own datasets. In addition, it can also serve as a means of education to train students for GNN-based applications in chemistry.
- Use of a user's dataset with a given GNN model. This allows researchers and students to explore their own data.
- Integration of new GNN models. Machine learning researchers can use this option to test their new GNN models for chemical applications.

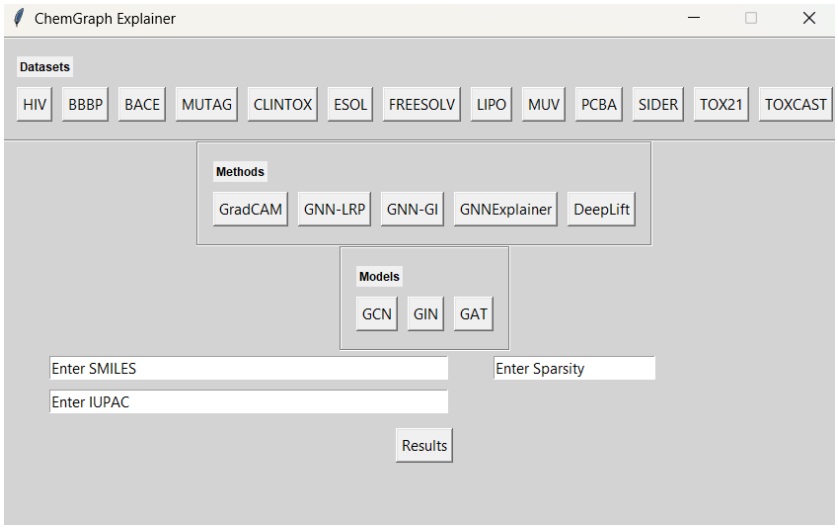

Fig. 3. ChemGraph Explainer Graphical User Interface.

– Integration of new explanation methods. Machine learning researchers can use this option to test their new explanation methods for chemical applications.

In addition, ChemGraph Explainer provides visualization functionality to make the explanation results visible.

The current implementation is restricted to pre-specified datasets and given GNN models and explanation methods. Working with ChemGraph Explainer thus has the following typical workflow. When the program is run, the user should select a dataset, a GNN model, and an explanation method. Afterwards, the user can select a particular molecule to explore it's classification and the related explanation. This can either be done by manually entering a SMILES text or by selecting a molecule from the panel. In addition, in order to calculate Fidelity+ and Fidelity- metrics, sparsity value (between 0-1) must be specified. The explanation can then be visualized by clicking the result button.

4 Case Studies

In this section, the explanation results we obtained with different datasets will be shown and evaluated. Particularly, using ChemGraph Explainer, we illustrate the reasons for the predictions for a molecule with different explanation methods of a molecule. Also, the obtained fidelity evaluation metrics will be evaluated. The datasets used in this study are BACE [28], BBBP [28], and MUTAG [28]. The BBBP dataset contains information about the ability of molecules to cross the blood-brain barrier (BBB). The BACE dataset is a dataset describing the

activity of BACE1 inhibitors. MUTAG is a dataset containing the activities of chemical compounds that cause mutation. All of these datasets are instances of binary classification problems.

We use a GCN model of 3 layers, each layer with 128-dimensional node features. The ReLU activation function is used between these layers. The dropout rate of the model was determined as 0.0. The learning rate was set to 0.001, the weight decay was set to 5e-4, and the batch size in each training epoch was set to 32. The output layer of the model is summarized by the maximum readout function.

To train the models, we divided all the datasets into 80% train, 10% evaluation, and 10% test dataset. Then, we trained GCN models for the different datasets. After the training phase, the average Fidelity+ and Fidelity- values of the test datasets were calculated at different sparsity values with five different explanation methods. When evaluating the fidelity metrics, an important criterion is that the Fidelity+ metric is expected to be as close to 1 while the Fidelity- metric as close to 0 as possible. When calculating the Fidelity+ value, the situation in which the most important features are removed from the input data is calculated, that is, when they are removed, it is expected to see a change in the result. On the other hand, when calculating the Fidelity- value, the least important features are removed and the prediction should not be affected by this change, that is, the Fidelity- value is expected to be close to 0. As can be seen in Figure 4, while the DeepLift and GNN-GI method produced successful explanation results in all three datasets, the GNN-LRP method could not approach the expected scores for the fidelity metrics for all three datasets. In addition, while the GradCam method produced successful fidelity metrics in the MUTAG dataset, it was one of the top three most successful methods for the Fidelity+ metric for the BACE dataset and was one of the top three methods for the Fidelity- metric in the BBBP dataset. As can be observed in all three dataset tests, as the sparsity value increases, the explanation methods move away from the targeted success metric values.

When values close to 0 are chosen for sparsity, such as 0.2, it means that 80% of the nodes within the selected molecule are important. In this case, it is more possible to produce successful fidelity metrics than low sparsity values. On the other hand, at high sparsity values, many nodes are marked as less important and therefore the expected success metric values are gradually moved away. By selecting the sparsity value to 0.5 with ChemGraph Explainer, Figure 5, 6, and 7 show the explanation results and visualization of a molecule selected from the BACE, BBBP, and HIV dataset, respectively, according to five different explanation methods. The red circles in the images of GradCam, DeepLift, and GNNExplainer show the parts that have a positive impact on the decision when the GCN model makes predictions. In GNN-LRP and GNN-GI images, red circles show the parts that have a positive impact on the decision when the GCN model makes predictions, while blue circles show the parts that have a negative impact. In addition, the upper left part of the visual output produced by ChemGraph

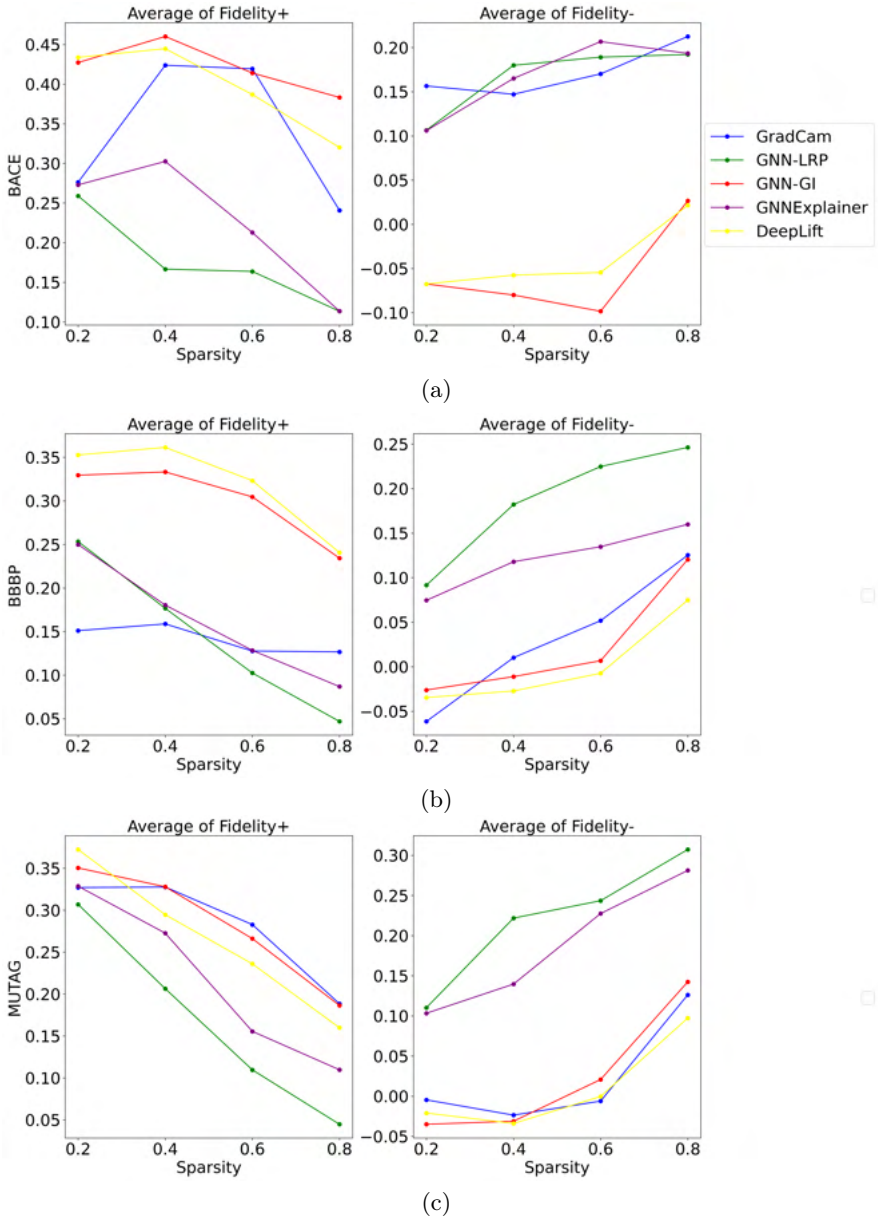

Fig. 4. Average of Fidelity+ (Left) and Fidelity- (Right) scores of the explanation results produced for the test dataset after training: (a) BACE, (b) BBBP, (c) MUTAG.

Explainer also shows the classification prediction made by the GNN model, the fidelity metrics, and the explanation method used for the evaluation.

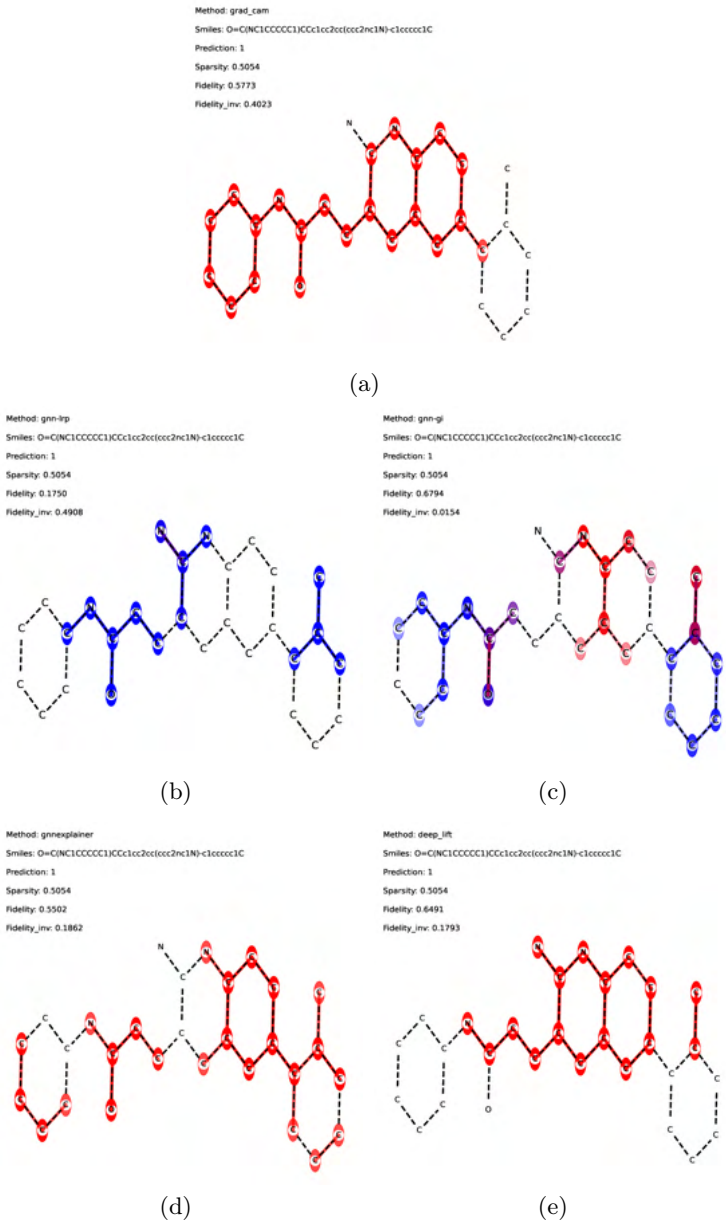

Fig. 5. Explanations of the prediction of a GCN model trained on the BACE dataset for a molecule selected from the BACE dataset using different XAI methods. (a) Grad-CAM, (b) GNN-LRP, (c) GNN-GI, (d) GNNExplainer, (e) DeepLift. Note that "Fidelity" and "Fidelityinv" correspond to the Fidelity+ and Fidelity- metrics, respectively.

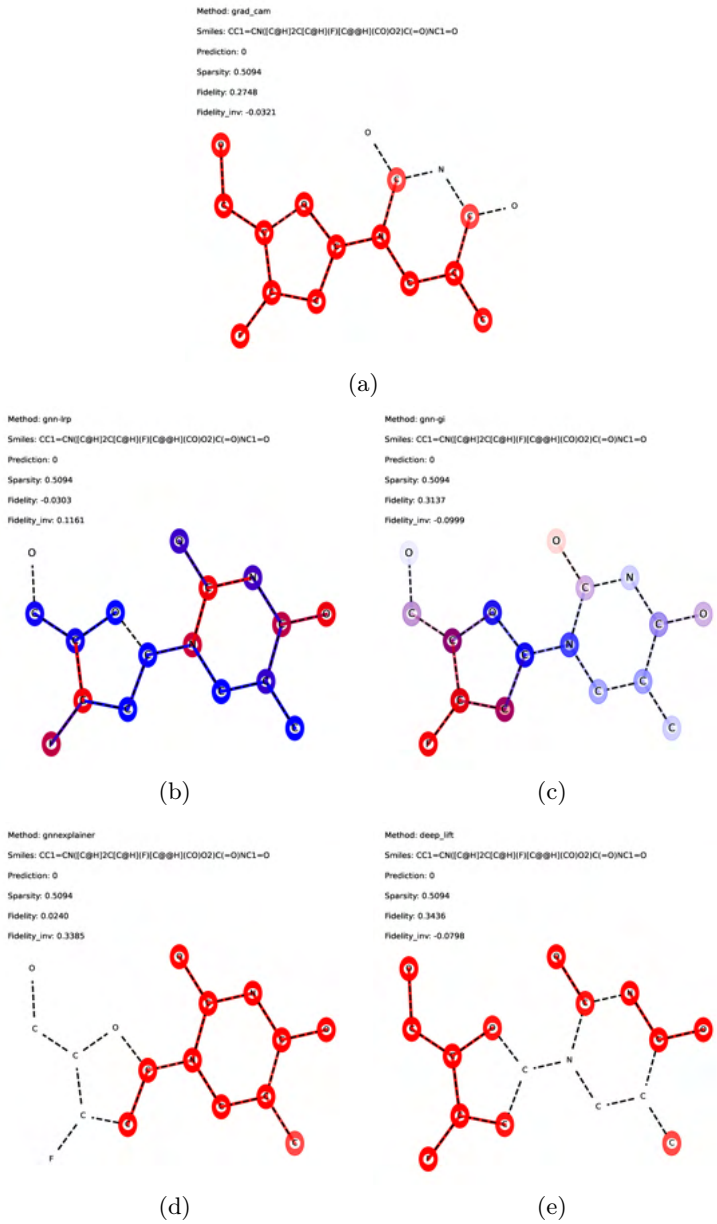

Fig. 6. Explanations of the prediction of a GCN model trained on the BBBP dataset for a molecule selected from the BBBP dataset using different XAI methods. (a) Grad-CAM, (b) GNN-LRP, (c) GNN-GI, (d) GNNExplainer, (e) DeepLift. Note that "Fidelity" and "Fidelityinv" correspond to the Fidelity+ and Fidelity- metrics, respectively.

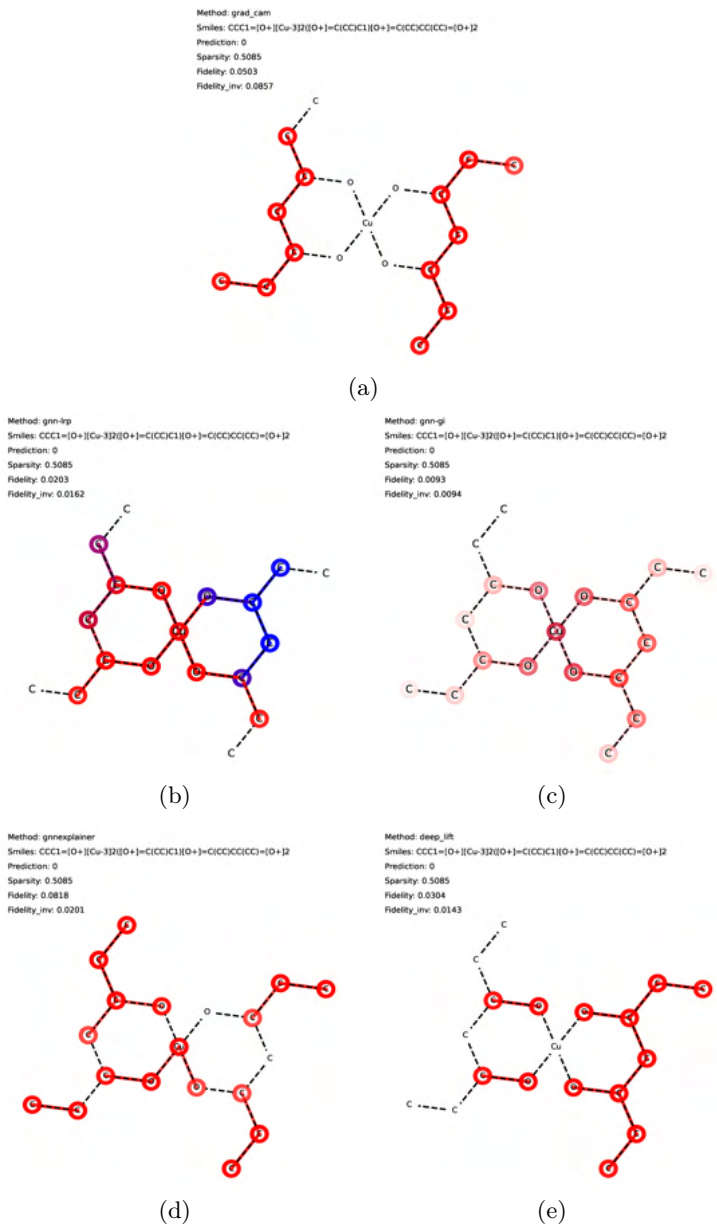

Fig. 7. Explanations of the prediction of a GCN model trained on the HIV dataset for a molecule selected from the HIV dataset using different XAI methods. (a) GradCAM, (b) GNN-LRP, (c) GNN-GI, (d) GNNExplainer, (e) DeepLift. Note that "Fidelity" and "Fidelityinv" correspond to the Fidelity+ and Fidelity- metrics, respectively.

5 Conclusion

With the ChemGraph Explainer, we offer researchers who conduct research using GNNs in the field of Chemistry the opportunity to evaluate the decisions made by the models they have developed and to see the reasons for the decisions taken. In addition to research, this tool can also be used for education purposes to support teaching about GNNs and their applications in the field of chemistry. The development of this user interface is in progress and the current implementation is restricted to pre-specified datasets and given GNN models and explanation methods. Additional functionalities will be integrated in future. our contributionsfuture goal is for users to upload custom GNN models and custom datasets to ChemGraph Explainer to receive the explanation results. This continued development includes, among others, further GNN models and XAI techniques such as causality-based methods, and handling of regression tasks.

References

1. An, H., Liu, X., Cai, W., Shao, X.: Explainable graph neural networks with data augmentation for predicting pka of c–h acids. Journal of Chemical Information and Modeling **64**(7), 2383–2392 (2024)
2. Arrieta, A.B., Rodríguez, N.D., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82–115 (2020)
3. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Müller, K.: How to explain individual classification decisions. Journal of Machine Learning Research **11**, 1803–1831 (2010)
4. Binder, A., Montavon, G., Lapuschkin, S., Müller, K., Samek, W.: Layer-wise relevance propagation for neural networks with local renormalization layers. In: 25th International Conference on Artificial Neural Networks (ICANN), Part II. pp. 63–71 (2016)
5. Cai, L., Ji, S.: A multi-scale approach for graph link prediction. In: AAAI Conference on Artificial Intelligence. pp. 3308–3315 (2020)
6. Dabkowski, P., Gal, Y.: Real time image saliency for black box classifiers. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NIPS). pp. 6967–6976 (2017)
7. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
8. Gohel, P., Singh, P., Mohanty, M.: Explainable AI: current status and future directions. CoRR **abs/2107.07045** (2021), https://arxiv.org/abs/2107.07045
9. Hamilton, W.L.: Graph Representation Learning. Springer (2020)
10. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. CoRR **abs/1506.05163** (2015), http://arxiv.org/abs/1506.05163
11. Huang, Q., Yamada, M., Tian, Y., Singh, D., Chang, Y.: Graphlime: Local interpretable model explanations for graph neural networks. IEEE Trans. on Knowledge and Data Engineering **35**(7), 6968–6972 (2023)
12. Letzgus, S., Wagner, P., Lederer, J., Samek, W., Müller, K.R., Montavon, G.: Toward explainable artificial intelligence for regression models: A methodological perspective. IEEE Signal Processing Magazine **39**(4), 40–58 (2022)

13. Liu, G., Tang, J., Tian, Y., Wang, J.: Graph neural network for credit card fraud detection. In: International Conference on Cyber-Physical Social Intelligence (ICCSI). pp. 1–6 (2021)
14. Liu, M., Gao, H., Ji, S.: Towards deeper graph neural networks. In: 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 338–348 (2020)
15. Liu, M., Luo, Y., Wang, L., Xie, Y., Yuan, H., Gui, S., Yu, H., Xu, Z., Zhang, J., Liu, Y., Yan, K., Liu, H., Fu, C., Oztekin, B.M., Zhang, X., Ji, S.: DIG: A turnkey library for diving into graph deep learning research. Journal of Machine Learning Research **22**(240), 1–9 (2021)
16. Liu, Y., Wang, Y., Vu, O., Moretti, R., Bodenheimer, B., Meiler, J., Derr, T.: Interpretable chirality-aware graph neural network for quantitative structure activity relationship modeling. In: The First Learning on Graphs Conference (2022), https://openreview.net/forum?id=W2OStztdMhc
17. Pope, P.E., Kolouri, S., Rostami, M., Martin, C.E., Hoffmann, H.: Explainability methods for graph convolutional neural networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10764–10773 (2019)
18. Ramsundar, B., Eastman, P., Walters, P., Pande, V., Leswing, K., Wu, Z.: Deep Learning for the Life Sciences. O'Reilly Media (2019)
19. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1135–1144 (2016)
20. Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.R.: Explaining deep neural networks and beyond: A review of methods and applications. Proceedings of the IEEE **109**(3), 247–278 (2021)
21. Schnake, T., Eberle, O., Lederer, J., Nakajima, S., Schütt, K.T., Müller, K., Montavon, G.: XAI for graphs: Explaining graph neural network predictions by identifying relevant walks. CoRR **abs/2006.03589** (2020), https://arxiv.org/abs/2006.03589
22. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. Int. Journal of Computer Vision **128**(2), 336–359 (2020)
23. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: International Conference on Machine Learning, (ICML). pp. 3145–3153 (2017)
24. Sun, M., Zhao, S., Gilvary, C., Elemento, O., Zhou, J., Wang, F.: Graph convolutional networks for computational drug development and discovery. Briefings in bioinformatics **21**(3), 919–935 (2019)
25. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: 6th International Conference on Learning Representations (ICLR) (2018)
26. Vu, M.N., Thai, M.T.: PGM-Explainer: Probabilistic graphical model explanations for graph neural networks. In: Annual Conference on Neural Information Processing Systems (NeurIPS) (2020)
27. Wellawatte, G.P., Gandhi, H.A., Seshadri, A., White, A.D.: A perspective on explanations of molecular prediction models. Journal of Chemical Theory and Computation **19**(8), 2149–2160 (2023)
28. Wu, Z., Ramsundar, B., Feinberg, E., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., Pande, V.: MoleculeNet: a benchmark for molecular machine learning. Chemical Science **9**, 513–530 (2018)

29. Wu, Z., Wang, J., Du, H., Jiang, D., Kang, Y., Li, D., Pan, P., Deng, Y., Cao, D., Hsieh, C.Y., Hou, T.: Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. Nature Communications **14**(1), 2585 (2023)
30. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Networks and Learning Systems **32**(1), 4–24 (2021)
31. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: 7th International Conference on Learning Representations (ICLR) (2019)
32. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: GNNExplainer: Generating explanations for graph neural networks. In: Annual Conference on Neural Information Processing Systems (NeurIPS). pp. 9240–9251 (2019)
33. Yuan, H., Yu, H., Gui, S., Ji, S.: Explainability in graph neural networks: A taxonomic survey. IEEE Trans. on Pattern Analysis and Machine Intelligence **45**(5), 5782–5799 (2023)
34. Yuan, H., Cai, L., Hu, X., Wang, J., Ji, S.: Interpreting image classifiers by generating discrete masks. IEEE Trans. on Pattern Analysis and Machine Intelligence **44**(4), 2019–2030 (2022)
35. Yuan, H., Tang, J., Hu, X., Ji, S.: XGNN: towards model-level explanations of graph neural networks. In: 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 430–438 (2020)
36. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. In: Annual Conference on Neural Information Processing Systems (NeurIPS). pp. 5171–5181 (2018)
37. Zhang, M., Cui, Z., Neumann, M., Chen, Y.: An end-to-end deep learning architecture for graph classification. In: AAAI Conference on Artificial Intelligence (2018)
38. Zhang, S., Tong, H., Xu, J., Maciejewski, R.: Graph convolutional networks: a comprehensive review. Computational Social Networks **6** (2019)
39. Zhang, Y., DeFazio, D., Ramesh, A.: Relex: A model-agnostic relational model explainer. In: AAAI/ACM Conference on AI, Ethics, and Society (AIES). pp. 1042–1049 (2021)
40. Zitnik, M., Agrawal, M., Leskovec, J.: Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics **34**(13), i457–i466 (2018)
41. Zitnik, M., Nguyen, F., Wang, B., Leskovec, J., Goldenberg, A., Hoffman, M.M.: Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. Information Fusion **50**, 71–91 (2019)