Explaining Graph Neural Networks

Steffen Backmann[†] Rushan Wang[†] Kenza Amara[†]

[†] Department of Computer Science, ETH Zurich

{sbackmann,ruswang}@student.ethz.ch,kenza.amara@ai.ethz.ch

Abstract

Graph Neural Networks (GNNs) achieve state-of-the-art performance in various graph-related tasks. However, their black-box nature often limits the interpretability and trustworthiness. Numerous explainability methods have been proposed to uncover the decision-making logic of GNNs, by generating underlying explanatory substructures. In this paper, we propose an interactive dashboard for domain experts that want to better understand the reasons behind toxic molecules and shed light on the most important graph entities. This dashboard is also meant for ML experts to better understand and compare current explainability methods. In this same dashboard, we allow users to manipulate explanations and eventually correct them based on their scientific prior knowledge. We hope that future work stemming from this dashboard will lead to new discoveries in molecules' toxicity but also lead to the development of more appropriate explainability methods.

1 Introduction

1.1 Graph neural networks

Graph Neural Networks (GNNs) have emerged as a powerful tool for studying graph-structured data in various applications, including social networks, drug discovery, and recommendation systems [4–6, 8, 24, 27, 35]. The explainability and trustworthiness of GNNs are crucial for their successful deployment in real-world scenarios, especially in high-stake applications, such as anti-money laundering, fraud detection, and healthcare forecasting [1, 15, 31]. Explanations for GNNs aim to discover the reasoning logic behind their predictions, making them more understandable and transparent to users. Explanations also help identify potential biases and build trust in the decision-making process of the model. Furthermore, they aid users in understanding complex graph-structured data and improve outcomes in various applications through better feature selection [7, 28, 34].

1.2 Explainability for GNN

Numerous explanation methods have been extensively studied for GNNs, including gradient-based attribution methods [3, 16, 20], perturbation-based methods [10, 17, 26, 29, 32], *etc.* However, most of these methods optimize individual explanations for a specific instance, lacking global attention to the overall dataset and the ability to generalize to unseen instances. To tackle this challenge, generative explainability methods have emerged recently, which instead formulate the explanation task as a distribution learning problem. Generative explainability methods aim to learn the underlying distributions of the explanatory graphs across the entire graph dataset [12, 14, 25, 30], providing a more holistic approach to GNN explanations.

1.3 Explain molecular toxicity

In the scope of this project, we focus on explainability for molecular graphs. In particular, we try to explain why a molecule is predicted as toxic or non-toxic. The explanation is a subgraph of the initial molecular graph that contains the graph entities, i.e., nodes, edges and node features, that contribute the most to the toxicity of the molecule.

The dataset used for this interface is **MUTAG**, a collection of 188 nitroaromatic compounds and it includes binary labels on their mutagenicity on Salmonella typhimurium. The chemical fragments -NO2 and -NH2 in mutagen graphs are labeled as ground-truth explanations [13].

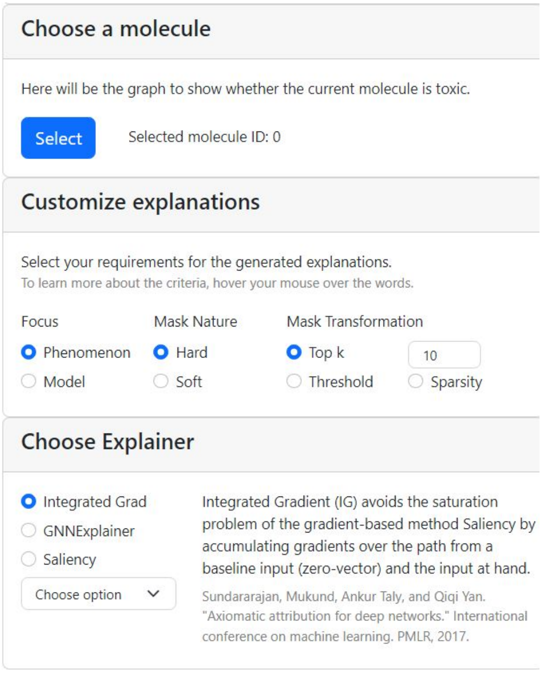
The network structure of the GNN model for graph classification is a series of 3 GAT [21] layers with ReLU activation, followed by a max pooling layer to get graph representations before the final fully connected layer. We split the train/validation/test with 80/10/10% for all datasets and adopt the Adam optimizer with an initial learning rate of 0.001. Each model is trained for 200 epochs with early stopping.

The interface includes explanations generated by different classes of explainers including gradient/feature-based methods and perturbation-based methods. It compares the following methods: **Saliency (SA)** measures node importance as the weight on every node after computing the gradient of the output with respect to node features [3]; **Integrated Gradient (IG)** avoids the saturation problem of the gradient-based method Saliency by accumulating gradients over the path from a baseline input (zero-vector) and the input at hand [20]; **Grad-CAM** is a generalization of class activation maps (CAM) [18]; **Occlusion** attributes the importance of an edge as the difference of the model's initial prediction on the graph after removing this edge [33]; **GNNExplainer** computes the importance of graph entities (node/edge/node feature) using the mutual information [29]; We also use **Basic GNNExplainer** that considers only edge importance; **PGM-Explainer** perturbs the input and uses probabilistic graphical models to find the dependencies between the nodes and the output [22]. We include the pre-computed masks in the interface backend.

1.4 Our Contribution

The GNN Explainer interface provides access and visualizations to the most common explainability methods. It tests them on the molecular dataset MUTAG. This interface is made for two types of users:

- Domain experts in biology or chemistry can increase their understanding on molecule toxicity, but also compare the generated explanations to their prior knowledge. They can correct current explainability methods by editing the importance, i.e. the weight that is assigned to each bond in the molecules. They can also explore completely new atom combinations and think out of the box.



Choose a molecule

Here will be the graph to show whether the current molecule is toxic.

Select | Selected molecule ID: 0

Customize explanations

Select your requirements for the generated explanations.
To learn more about the criteria, hover your mouse over the words.

Focus: ● Phenomenon ○ Model

Mask Nature: ● Hard ○ Soft

Mask Transformation: ● Top k ○ Threshold ○ Sparsity; 10

Choose Explainer

● Integrated Grad
○ GNNExplainer
○ Saliency
Choose option

Integrated Gradient (IG) avoids the saturation problem of the gradient-based method Saliency by accumulating gradients over the path from a baseline input (zero-vector) and the input at hand.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks." International conference on machine learning. PMLR, 2017.

Figure 1: The left panel of the interface contains all the elements to set the prior requirements for the explanation to be visualized in the main component. Here, users can choose the *explainee*, i.e., the molecule to be explained, the prior criteria for the explanation and the explainability method.

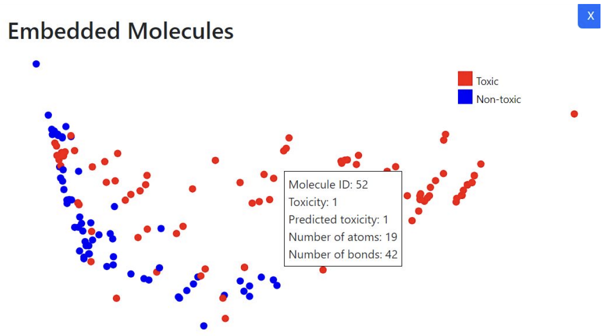
- ML experts familiar with the research of xAI get a better understanding of the existing methods by trying them out on a concrete, real-world graph classification dataset and accessing the properties of each explanatory mask. With the provided short description of the methods, they can observe their theoretical differences in practice and potentially discover methods previously unknown to them.

2 User Interface

Dashboard Overview The dashboard is divided into a left, a center and a right part. The left part contains components for selecting the molecule instance to explain, as well as the explanation criteria expected by the users. It also contains the elements to set the environment for explanations, called users' explanation criteria. Finally, in the bottom of the left part we find the set of explainability methods with their description and reference. The main part in the center is the visualization panel: It displays the graph visualization of the selected molecule with its explanation. The right component shows the description of the selected molecule and the scores and properties of the generated explanation. Beside the main dashboard, another selection window contains a scatter plot with all available data points.

2.1 Users selection

Selection of an instance On the top left of the dashboard in Figure 1, the user can click on the ”SELECT” button to choose a molecule to explain. A window shown in Figure 2 pops up with all the molecules of the dataset embedded in the GNN latent space projected to 2 dimensions using principal component analysis (PCA). By hovering the mouse on the data points, the user can see details about the respective molecule. The colors blue and red encode the toxicity of the molecules. Thus, users can easily distinguish toxic from non-toxic molecules and focus on choosing to explain toxic molecules in particular. After clicking on a molecule, the window closes and the selected molecule is displayed in the center part of the dashboard.



Molecule ID: 52
Toxicity: 1
Predicted toxicity: 1
Number of atoms: 19
Number of bonds: 42

Figure 2: On this scatter plot, users can select the molecule they want to explain. Blue data points correspond to non-toxic molecules while red encodes toxic molecules. By hovering on the data points, they get information about each molecule. Clicking on one of the data point closes the window.

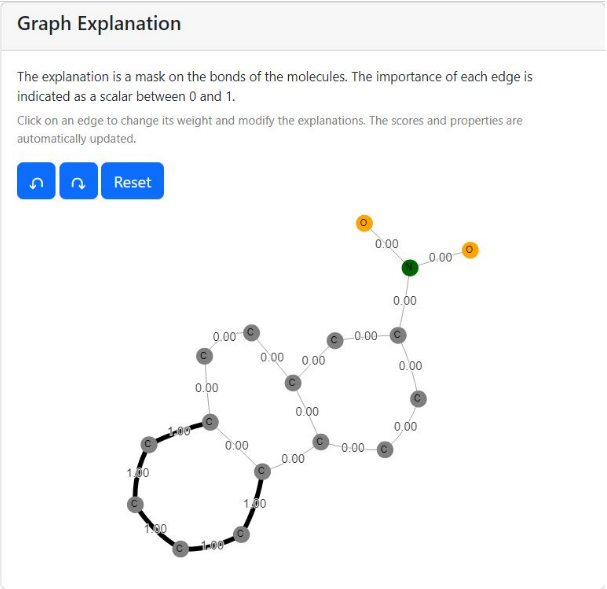
Selection of the explanation criteria Users specify their explanation criteria in the left-center component displayed in Figure 1. Following the systematic evaluation framework GraphFramEx [2], we distinguish three main selection criteria: the focus, the size and the nature of the explanation. Each criterion is explained when we hover the mouse over the concept. We provide the definition of each term as well as the description of each option for each criterion. We use radio buttons to be ticked for the selection. Users choose between ”phenomenon” or ”model” for the focus of the explanation. They choose if they want a hard, i.e. non-weighted, or a soft, i.e. weighted, explanatory subgraph. Finally, they can decide on the size of the expected explanation by selecting one out of the three mask transformation strategies: top-k, threshold or sparsity, and indicate the level of sparsity they want.

Selection of the explainability method Users can select one of the seven explainability methods available in the interface. The three methods with the highest scores are directly displayed on the dashboard. More options can be viewed when clicking on the dropdown arrow. For domain expert users, we have added a description of the methods to help them understand how each method works. We also add the main reference paper for them to learn more about the methods. Even if we assume that ML experts have prior knowledge in xAI, we still believe the descriptions can be useful for them as well.

2.2 Visualization and analysis of the explanation

Molecule visualization Once the selection of the molecule is done and the explanation requirements are specified, users can visualize the chosen molecule in the center main component. The atoms are colored based on their nature, that is the chemical element they encode. The edges in the graph represent the bonds between two atoms. The explanation corresponds to the bold edges in the molecule. Explanation weights are values between 0 and 1 displayed on the edges to indicate the importance level of the bond with respect to the predicted toxicity. By default, the explainability method is Integrated Gradient.

Molecule description On the top right of the dashboard in Figure 4, the selected molecule is described by its number of atoms and bonds. We also communicate the ground truth label, i.e., whether the molecule is toxic or not, and the model prediction about the selected molecule's toxicity, which can be correct or wrong. This description changes when we select another molecule from the scatter plot.

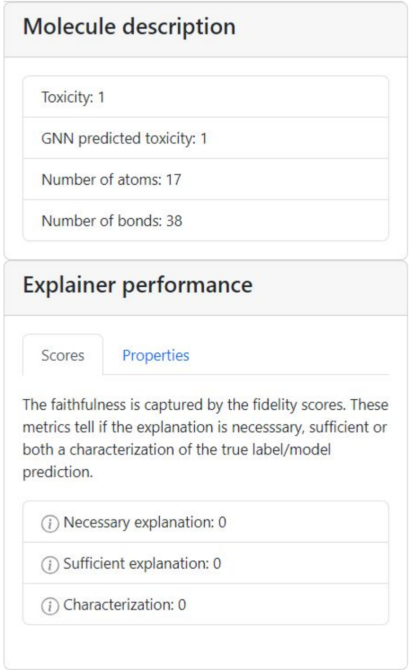


Graph Explanation

The explanation is a mask on the bonds of the molecules. The importance of each edge is indicated as a scalar between 0 and 1.

Click on an edge to change its weight and modify the explanations. The scores and properties are automatically updated.

Figure 3: The main panel in the center of the interface contains the visualization of the chosen molecule as a graph with nodes corresponding to the atoms and edges to the bonds between the,. The explanation is displayed as the bold edges. Edge weights indicate the importance scores of the explanatory bonds.



Molecule description

Toxicity: 1

GNN predicted toxicity: 1

Number of atoms: 17

Number of bonds: 38

Explainer performance

The faithfulness is captured by the fidelity scores. These metrics tell if the explanation is necesssary, sufficient or both a characterization of the true label/model prediction.

ⓘ Necessary explanation: 0

ⓘ Sufficient explanation: 0

ⓘ Characterization: 0

Figure 4: The right panel of the interface displays the description of the molecules with the ground truth, predicted toxicity and information about the structure, i.e., number of atoms and bonds. It also contains the explainer performance with the explanation quality evaluated on three metrics and the explanatory mask properties.

Evaluation of explanations Explanations are scored using the faithfulness metrics described in [2]. The scores appear in the right component of the dashboard in Figure 4. Definitions of the metrics are provided when the user clicks on the small info symbol next to the respective term. Users can thus easily grasp the meaning of concepts like ”necessary explanation”, ”sufficient explanation” and ”characterization”.

Considering the large spectrum of possible explanations, we propose to classify explanations in two categories based on their fidelity scores. Each category defines the role of the explanation in producing the observed outputs: the explanation can be necessary and/or sufficient.

- SUFFICIENT EXPLANATION An explanation is sufficient if it leads by its own to the initial prediction of the model explanation. Since other configurations in the graph may also lead to the same prediction, it is possible to have multiple sufficient explanations for the same prediction. A sufficient explanation has a fid_- score close to 0. We report $(1 - fid_-)$ in the score section of the interface.
- NECESSARY EXPLANATION An explanation is necessary if the model prediction changes when you remove it from the initial graph. Necessary explanations are similar to counterfactual explanations [23]. A necessary explanation has a fid_+ score close to 1.

An explanation is a characterization of the prediction if it is both necessary and sufficient. It can be interpreted as the certificate for a specific class or label. Explainability methods should aim at returning this type of explanation as they are the most informative and complete.

Mask properties Each explanation is a mask that has some interesting properties such as entropy, size and sparsity. Those are communicated to the users in the right component in Figure 4 when clicking on the ”Properties” option. Here the properties are described in a small paragraph and results appear next to the terms. The mask properties are meant especially for ML experts who want to compare explanatory masks generated by different methods on a structural level.

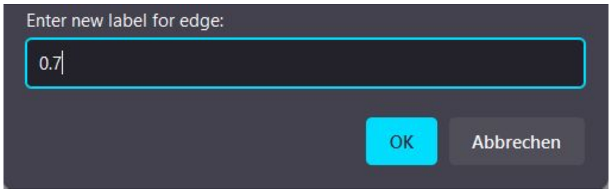
Both, scores and mask properties are computed on-demand: When users select a new method or a new molecule, the request is sent from the frontend to the backend. Scores and properties are computed after running the GNN model in the backend and sent back to the frontend.

2.3 Interaction with the explanation

Users are not only given generated explanations, but they can also interact with them. Our interface enables users to modify the explanations and observe the consequences of their changes on scores and properties.

Edge addition or removal When clicking on a bond of the molecule displayed in the main component, a window as displayed in Figure 5 pops up and users are asked to enter a new weight for the selected edge. After entering the weight and clicking on the ”OK” button, the explanatory mask is updated with a new value for this edge. Scores and properties are automatically re-computed with the GNN model running in the backend.

Edition history The interface also allows users to edit the modifications they have done. They can move back to the previous graph state with the ”Undo” button and reverse this action moving towards their most recent modification with the ”Redo” button. Thus, they can travel back and forth between previous and more recent modifications. The edit history allows domain experts to observe the changes in the scores and find the optimal explanatory bonds in the molecule. The ”Reset” button allows to go back to the initial explanation generated by the method, thereby erasing the entire edition history.



Enter new label for edge:

0.7

Figure 5: Pop-up window when clicking on a bond of the molecule displayed in the center panel. Users can decide to edit the weight of the edge with a new value between 0 and 1. Clicking on the "OK" button updates the edge weight and leads to a re-computation of the explanation's scores and properties.

3 Discussion and Outlook

The interface allows people to compare diverse explainability methods on a very common molecular graph dataset. Users have also the possibility to directly interact with the generated explanations when editing the edge weights. There are regular communications between frontend and backend. Each time users select a new molecule, a retrieval request is sent from the frontend to the backend. The molecular graph and re-computed scores and mask properties are then sent back to the frontend almost instantly. The same applies to the selection of different explainability methods as well as explanation customizations. Every action taken on the interface is realised with no time latency.

Our future objective is to extend the interface to user-specified datasets. Users will be able to add their own graph dataset in the correct format. Explainability methods will then run in the backend and return the explanatory masks for their graphs and classification task. The selection of molecules is also limited now to choosing a data point on a scatter plot only; it could be improved with a more complex search engine, e.g., searching for molecules with more than 20 bonds. We also plan to give users the option to select more GNN models: : GCN [11], GIN [9], GAT [21], and GraphTransformer [19]. To make the interface more interactive, we will also enable users to select multiple explainability methods at once and aggregate their outputs. Another future direction is to customize and extend the edit history: We want to visualize the evolution of the scores and properties when modifying the explanation. With an appropriate visualization, users would easily observe the score improvement and general changes in the explanation properties. Finally, as of now the system does not guide the users in the editing process. It would be interesting to give users suggestions about what edge to add next to the explanation to improve the explanation quality.

Acknowledgments

The authors wish to thank Gen Li, teaching assistant in the Interactive Machine Learning course, for supervising our project and sharing his insights and advice. We also thank Menna El Assady and Furui Cheng for their suggestions on how to improve the interface.

References

[1] C. Agarwal, O. Queen, H. Lakkaraju, and M. Zitnik. Evaluating explainability for graph neural networks. *Scientific Data*, 10(1):144, 2023.

[2] K. Amara, R. Ying, Z. Zhang, Z. Han, Y. Shan, U. Brandes, S. Schemm, and C. Zhang. Graphframex: Towards systematic evaluation of explainability methods for graph neural networks. *arXiv preprint arXiv:2206.09677*, 2022.

[3] F. Baldassarre and H. Azizpour. Explainability techniques for graph convolutional networks. *CoRR*, abs/1905.13686, 2019.

[4] P. Bongini, M. Bianchini, and F. Scarselli. Molecular generative graph neural networks for drug discovery. *Neurocomputing*, 450:242–252, 2021.

[5] A. Chaudhary, H. Mittal, and A. Arora. Anomaly detection using graph neural networks. In *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 346–350. IEEE, 2019.

[6] D. Cheng, F. Yang, S. Xiang, and J. Liu. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121:108218, 2022.

[7] E. Dai, T. Zhao, H. Zhu, J. Xu, Z. Guo, H. Liu, J. Tang, and S. Wang. A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability. *arXiv preprint arXiv:2204.08570*, 2022.

[8] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

[9] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec. Strategies for pre-training graph neural networks, 2020.

[10] Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang. Graphlime: Local interpretable model explanations for graph neural networks. *CoRR*, abs/2001.06216, 2020.

[11] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[12] W. Li, Y. Li, Z. Li, J. Hao, and Y. Pang. Dag matters! gflownets enhanced explainer for graph neural networks. *arXiv preprint arXiv:2303.02448*, 2023.

[13] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang. Parameterized explainer for graph neural network. In *NeurIPS*, 2020.

[14] J. Ma, R. Guo, S. Mishra, A. Zhang, and J. Li. Clear: Generative counterfactual explanations on graphs. *arXiv preprint arXiv:2210.08443*, 2022.

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

[16] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann. Explainability methods for graph convolutional neural networks. In *CVPR*, pages 10772–10781, 2019.

[17] M. S. Schlichtkrull, N. D. Cao, and I. Titov. Interpreting graph neural networks for NLP with differentiable edge masking. *CoRR*, abs/2010.00577, 2020.

[18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017.

[19] Y. Shi, Z. Huang, W. Wang, H. Zhong, S. Feng, and Y. Sun. Masked label prediction: Unified massage passing model for semi-supervised classification. *CoRR*, abs/2009.03509, 2020.

[20] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *ICML*, volume 70, pages 3319–3328, 2017.

[21] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks, 2018.

[22] M. N. Vu and M. T. Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. In *NeurIPS*, 2020.

[23] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. Nov. 2017.

[24] J. Wang, S. Zhang, Y. Xiao, and R. Song. A review on graph neural network methods in financial applications. *arXiv preprint arXiv:2111.15367*, 2021.

[25] X. Wang, Y. Wu, A. Zhang, F. Feng, X. He, and T. Chua. Reinforced causal explainer for graph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

[26] X. Wang, Y.-X. Wu, A. Zhang, X. He, and T.-S. Chua. Towards multi-grained explainability for graph neural networks. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

[27] O. Wieder, S. Kohlbacher, M. Kuenemann, A. Garon, P. Ducrot, T. Seidel, and T. Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

[28] B. Wu, J. Li, J. Yu, Y. Bian, H. Zhang, C. Chen, C. Hou, G. Fu, L. Chen, T. Xu, et al. A survey of trustworthy graph learning: Reliability, explainability, and privacy protection. *arXiv preprint arXiv:2205.10014*,

2022.

[29] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*, pages 9240–9251, 2019.

[30] H. Yuan, J. Tang, X. Hu, and S. Ji. XGNN: towards model-level explanations of graph neural networks. In R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, editors, *KDD*, pages 430–438, 2020.

[31] H. Yuan, H. Yu, S. Gui, and S. Ji. Explainability in graph neural networks: A taxonomic survey. *CoRR*, 2020.

[32] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji. On explainability of graph neural networks via subgraph explorations. *ArXiv*, 2021.

[33] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[34] H. Zhang, B. Wu, X. Yuan, S. Pan, H. Tong, and J. Pei. Trustworthy graph neural networks: Aspects, methods and trends. *arXiv preprint arXiv:2205.07424*, 2022.

[35] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

A Contribution statement

A.1 Steffen Backmann

- Creation of the API to fetch the different data sources.
- Implementation of the main visualization.
- Implementation of the scatter plot.
- Contribution to keeping the repository README updated.
- Final polish and deployment of the application.

A.2 Kenza Amara

- Implementation of the backend python codes.
- Implementation of the evaluation modules in the frontend.
- Contribution to writing the report.

A.3 Rushan Wang

- Implementation of the dashboard design.
- Implementation of the explanation criteria and explainability methods selection.
- Contribution to creating the poster.