A framework for Counterfactual Explainability in Graph Neural Networks

Maria Myrto Villia[1,2,*]

[1]Computer Science Department, University of Crete, Greece
[2]Institute of Computer Science, FORTH, Greece

Abstract

This PhD research focuses on counterfactual explainability for graph neural networks (GNNs), with an emphasis on local-level, model-agnostic, post-hoc explanations. A novel pipeline is proposed to enable more targeted and controllable graph edits. The approach combines ideas from factual explainability with edge prediction models inspired by link prediction. The aim is to enhance the quality, robustness, and interpretability of counterfactual explanations while keeping the method computationally tractable. Experiments are conducted on real-world and synthetic graph classification benchmarks, and the approach is compared against existing state-of-the-art methods across multiple evaluation metrics.

Keywords

counterfactuals,, explainability, graph neural networks

1. Description and motivation of the problem

Counterfactual explanations identify the minimal changes needed in an input to alter a model's prediction and offer a promising way to make complex AI systems more transparent [1, 2]. My PhD research investigates their utility in the context of Graph Neural Networks (GNNs), where explaining predictions is particularly difficult because both features and graph structure affect the outcome [3]. For example, in drug discovery, where molecules are represented as graphs, a counterfactual explanation can show that a small change in a molecule, such as adding or removing a bond, would be enough to change the prediction of the model from non-toxic to toxic or from inactive to active.

A significant part of this PhD research is devoted to the evaluation of counterfactual explainability methods, which is not straightforward. One major challenge is the limited availability of datasets with ground-truth explanations, which leads the XAI community to rely on synthetic benchmarks. Moreover, evaluation practices in the literature remain relatively heterogeneous, with different works adopting different metrics and protocols, which complicates direct and consistent comparison across methods.

Factual explainers [3, 4], which, given a trained GNN model (oracle) and an input graph, aim to extract the subgraph that plays a crucial role in the oracle's prediction, have dominated the field of GNN explainability for years. Counterfactual explanations on graphs have a much shorter history [2, 5], however, they provide a complementary advantage, as they indicate not only why a prediction was made, but also what should be changed to alter it.

2. Related Work

Most GNN explainers focus on factual, local, and post-hoc explanations. Early works in counterfactual explainability, such as CF-GNNExplainer [6], CFExplainer [7], and CF^2 [8] focus on edge removal, leading to explanations that are substructures of the input graph, significantly limiting their expressive power. The first counterfactual explainers that supported edge addition each have their own generalization limitations. For instance, LEGIT [9] and MEG [10] rely on domain knowledge to guide a

Doctoral Consortium of the 23rd International Conference on Principles of Knowledge Representation and Reasoning (KR 2026 DC), July 20-23, 2026, Lisbon, Portugal

*Corresponding author.
mvillia@ics.forth.gr (M. M. Villia)

reinforcement learning module. GREASE [11] is tailored to recommendation tasks, while CLEAR [12] assumes a causal model of the data. C2Explainer [13], in contrast, uses a domain-independent edge masking approach, but limits possible edge additions to those defined in a predefined supergraph.

To address these limitations, several counterfactual explainers have been proposed in recent years such as RSGG-CE [14], ATEX-CF [15], D4Explainer [16] and GIST [17]. RSGG-CE and ATEX-CF [15] draw on ideas from adversarial learning to model edge probability distributions, which is a sensible choice given the ability of adversarial methods to introduce targeted graph perturbations. However, these approaches are often computationally demanding, although RSGG-CE reduces this cost through an optimized sampling strategy. D4Explainer uses a denoising diffusion model to learn the distribution of explanation graphs. It achieves strong performance and can produce both factual and counterfactual explanations, but its very high computational cost and its focus on diffusion over discrete features limit its applicability. GIST [17] introduces a backtracking mechanism based on spectral style transfer to move across the decision boundary of the oracle. While the method performs better on binary classification tasks, it tends to generate large explanations and, more importantly, cannot control the target prediction in multi-label classification settings.

3. The progress to date

3.1. The proposed framework

The first of the two main phases of this PhD work has been completed and focused on the development of a model-agnostic, post-hoc, local-level counterfactual explanation method for graph classification tasks; where counterfactual graphs are constructed through edge removals and edge additions to the original graph. The counterfactual explanation is then defined as the set of added and removed edges. Edge removal and edge addition were perceived as conflicting operations; therefore, they were optimized independently before integrating their outputs into a unified framework to form a counterfactual explanation of a graph. In the proposed method, edges are first removed through a deconstruction step, and then new edges are added during a reconstruction step.

During the deconstruction step, a state-of-the-art factual explainer, SubgraphX [18], is used to identify the factual subgraph that contains the edges that contribute the most to the original prediction. The factual subgraph preserves the original prediction. Combinations of edges from the factual subgraph are selectively removed, thereby destroying the pattern that led to the initial prediction. This step produces multiple deconstructed graphs. Then, in the reconstruction step, link prediction methods are leveraged, as they are designed to infer future, missing, or unobserved relationships in complex networks. Interestingly, this line of research has been largely overlooked in the context of counterfactual explanations for GNNs.

Link prediction models are typically trained on a single graph and evaluated either on the same nodes (transductive setting [19, 20]) or on previously unseen nodes within the same underlying graph structure (inductive setting) [21, 22]. One contribution of this work is the adaptation of an inductive approach to GNN explainability for graph classification. Thus, to support counterfactual generation in graph classification, link prediction models are trained on datasets containing multiple graphs, so that patterns learned from the training graphs can be generalized to unseen test graphs. These predicted edges are then used to construct counterfactual explanations.

Training For the edge addition process, an Encoder-Decoder architecture was developed. To adapt inductive learning on multiple training graphs, the edges of each graph are split into two categories: message passing edges, used by the encoder to propagate information between nodes, and supervision edges, used by the decoder to learn edge existence. The supervision set consists of edges from the graph corresponding to real observed connections, referred to as positive edges, together with negative samples (non-edges) generated through negative sampling to represent non-existent connections, referred to as negative edges. To train the model, the positive edges are labeled with 1, and the negative edges with 0. The decoder is trained to distinguish between these two cases using the binary cross-entropy loss.

Importantly, rather than just training a model to predict a potentially missing link between any pair of nodes, the goal is to adopt a pattern-driven approach, prioritizing links that lead to an alternative, yet predefined pattern. This is a crucial requirement for many multi-label real-world classification problems, e.g. protein function prediction, yet largely overlooked by most existing explainers. Thus, to equip the model with target-specific counterfactual generation, a one-hot vector of the target class is first projected through an MLP and injected into the embedding before it is passed to the decoder.

Inference During inference, all non-existent candidate edges of the deconstructed graph are fed into the model to estimate their likelihood. Only edges whose likelihood exceeds a threshold are retained. It is noteworthy that this process can naturally be enriched with domain knowledge by enforcing edge-level constraints during supervision. Specifically, edges that must be present, such as chemical bonds, can be fixed as positive training supervision edges and always be included in the training signal, while edges that are known to be impossible or invalid can be explicitly enforced as negative training supervision edges. In this way, inference will reflect the impact of the prior knowledge on the predictions and, ultimately, promote plausible explanations that avoid the recommendation of changes that are not actionable.

Denoising Furthermore, the already trained reconstruction model described above can also be used for denoising. This step is performed before the deconstruction process. Given the original graph, the model produces an edge probability matrix, which is then used to remove a number of edges with the lowest probabilities. The cutoff is determined on the basis of a small fraction of the area under the curve of the sorted edge probabilities.

Sampling To determine how many edges to remove from the factual subgraph and how many edges to add from those proposed by the reconstruction model above the threshold, a number x of edges are sampled from each set according to the following probability function:

$$p(x) = e^{-\alpha(x-\beta)^4}, \tag{1}$$

where $\alpha \in \mathbb{R}^+$ controls the rate at which probability decays as the number of edges deviates from a center of distribution β, with $\beta = \{0, 1, ..., \}$. Intuitively, the hyperparameter β encodes the intrinsic distance between graph patterns, i.e., the number of edge edit operations that can lead from one pattern to another if such information can be extracted from the domain. This sampling process is performed iteratively within a fixed time limit. At each iteration, a number of edges to remove and a number of edges to add are sampled according to the probability function above, and the corresponding graph edits are applied to the original graph. If the resulting graph changes the prediction of the model, it is retained as a valid counterfactual graph, and the modifications are maintained as a counterfactual explanation. In this way, multiple counterfactual explanations can be generated for the same input graph.

3.2. Evaluation

The proposed method has already been extensively evaluated against state-of-the-art baselines on both synthetic and real-world benchmarks. Specifically, four variants of the method - with and without the one-hot projection embedding, as well as with and without the denoising mode are evaluated - against the state-of-the-art counterfactual explainers, namely **CF**^{2} [8], one of the most prominent explainers that only remove edges, **D4Explainer** [16], **RSGNN-CE** [14], and **GIST** [17]. In addition, a global-level explainer, **GCFExplainer** [23], and a naive exhaustive-search **Random** baseline, in which edges are randomly added and removed, are included for reference.

The framework is evaluated on a collection of real-world and synthetic graph classification datasets, covering both binary and multi-class settings. The benchmarks include synthetic motif-based datasets, namely **BA-2Motifs**, and two variants constructed in this work, **BA-3Motifs** and **BA-4Motifs**, as well

as real-world sentiment datasets (**Graph-Twitter**, **Graph-SST5**) and a molecular property prediction dataset (**BBBP**). Additional experiments include the **MUTAG** dataset, which contains ground-truth explanations. In addition, motivated by the lack of explicit evaluation of GNN explainers under incomplete data, a variant of **BA-2Motifs**, named **BA-2Motifs-3classes**, is constructed to include an explicitly incomplete class.

A diversity of metrics are considered to achieve a multi-faceted comparative analysis: **Validity**: calculates the proportion of input graphs for which there is at least one counterfactual. **Explanation Size**: measures the number of edge edits that constitute the explanation. **Fidelity:** measures the decrease in the oracle confidence after the CF modifications. **Motif Proximity**: Following [24] and [25], this metric captures the ability of the explainer to modify the edges that adhere to the motif that generates the original prediction. It is applicable when ground-truth is available. **Minimality**: evaluates whether subsets of the edits of an explanation can also flip the oracle's prediction. **Validity after Noise (VaN)**: assessing robustness, it defines the fraction of counterfactual graphs that remain valid when a small noise σ is added to the input graphs without changing the prediction. **Edge Consistency after Noise (ECaN)**: given that VaN alone fails to reveal the impact the noise has on the quality of the explanation, ECaN quantifies the structural stability of CfXs using the Jaccard similarity between the edge-edit sets obtained from the original and noisy inputs.

Although the multi-faceted nature of the problem renders a universally best-performing approach difficult to achieve, the comparative analysis elucidates the importance of exploring a variety of qualitative criteria. RSGG-CE, for instance, stands out for its capacity to generate compact CfXs with high validity, especially considering its time efficiency, but presents moderate performance in all qualitative metrics. D4Explainer, on the other hand, seems to owe its impressive performance on real data to the computationally very intensive diffusion process, which furthermore seems to hurt performance in smaller, simpler graphs. The GCFExplainer, with its design to offer a global view, achieves its goal for high coverage and robustness, but at the same time its poor qualitative scores make clear why a deeper view is crucial for CfXs. In comparison to SoTA, the proposed method manages to consistently produce highly impactful and qualitative explanations, with characteristically compact sizes, and exceptionally high minimality and motif proximity performance, while maintaining a competitive level in coverage and robustness. This result underlines the accuracy of the graph edit selection model, considering that apart from size and fidelity, none of the other metrics is included in the optimization process.

Overall, the first phase of this PhD has been completed. A manuscript has been submitted to ECML PKDD 2026 and is currently under review. The research is now transitioning to the second phase, as outlined in the future work.

4. Future work

Building on the first phase of this PhD, two promising future directions naturally emerge.

The first is the study of counterfactual explanations for temporal graph neural networks that operate on dynamic graphs. Dynamic graphs evolve over time with a structure that is not fixed; nodes and edges may appear, disappear, or change. This makes the generation of explanations more challenging. It is therefore important to study how counterfactual explanations can be adapted to temporal settings. In particular, an important question is the following: when the temporal dimension is taken into account, what needs to change at a past time step in order to alter a prediction in the present? A counterfactual explanation identifies the minimal change in the temporal graph, either in its structure or in the timing of events, that would alter the prediction of the model [26]. In particular, it will be studied whether the edge-addition model described above, with suitable adaptations for temporal graphs, can appropriately add edges and produce meaningful counterfactual explanations.

The second is to build on recent advances in neuro-symbolic AI. In this context, models such as [27] incorporate knowledge-enhancement layers to enable formal verification of prediction consistency. These layers allow structured knowledge to be integrated during the learning process. Neuro-symbolic

AI is generally considered a very promising direction, as it combines the strengths of neural learning with symbolic reasoning, enabling more interpretable, consistent, and reliable models.

5. Research Impact

My research direction seeks to contribute to the broader effort of improving the transparency and trustworthiness of Graph Neural Networks. A domain where counterfactual explanations could be particularly useful is biomedical research, i.e. drug discovery. In this setting, adding an edge to the original graph may indicate that if an interaction between two drugs existed (represented as a link), the classifier would predict a higher risk for the patient. Thus, counterfactual explanations can assist researchers to better understand which relationships in the graph influence the model's prediction. Furthermore, combining link prediction with counterfactual generation opens an interesting research direction, especially in large or incomplete graphs, where realistic edge additions are important.

6. Acknowledgments

I would like to sincerely thank Theodore Patkos, Filippos Gouidis, and Panos Trahanias for their guidance, support, and valuable discussions throughout this work.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] J. Jiang, F. Leofante, A. Rago, F. Toni, Robust counterfactual explanations in machine learning: a survey, in: IJCAI, 2024.
[2] Z. Guo, Z. Wu, T. Xiao, C. Aggarwal, H. Liu, S. Wang, Counterfactual Learning on Graphs: A Survey, Machince Intelligence Research 22 (2025) 17–59.
[3] H. Yuan, H. Yu, S. Gui, S. Ji, Explainability in GNNs: A Taxonomic Survey , IEEE Trans. on Pattern Analysis & Machine Intelligence 45 (2023) 5782–5799.
[4] C. Agarwal, O. Queen, H. Lakkaraju, M. Zitnik, Evaluating explainability for graph neural networks, Scientific Data 10 (2023) 144.
[5] J. Kaddour, A. Lynch, et al., Causal Machine Learning: A Survey and Open Problems, Foundations and Trends in Optimization 9 (2025) 1–247.
[6] A. Lucic, M. A. Ter Hoeve, G. Tolomei, M. De Rijke, F. Silvestri, Cf-gnnexplainer: Counterfactual explanations for gnns, in: AISTATS, 2022.
[7] Z. Chu, Y. Wan, et al., Graph neural networks for vulnerability detection: A counterfactual explanation, in: ISSTA, 2024.
[8] J. Tan, S. Geng, Z. Fu, Y. Ge, S. Xu, Y. Li, Y. Zhang, Learning and evaluating gnn explanations based on counterfactual and factual reasoning, in: WWW, 2022.
[9] D. Bacciu, D. Numeroso, Explaining deep graph networks via input perturbation, IEEE Transactions on NN and Learning Systems 34 (2023) 10334–10345.
[10] D. Numeroso, D. Bacciu, Meg: Generating molecular counterfactual explanations for deep graph networks, in: IJCNN, 2021.
[11] Z. Chen, J. Huang, et al., Joint factual and counterfactual explanations for top-k gnn-based recommendations, ACM Transactions on Recommender Systems (2025).
[12] J. Ma, R. Guo, S. Mishra, A. Zhang, J. Li, Clear: generative counterfactual explanations on graphs, in: NeurIPS, 2022.

[13] J. Ma, I. Takigawa, A. Yamamoto, C2explainer: Customizable mask-based counterfactual explanation for graph neural networks, in: FAccT, 2025.
[14] M. A. Prado-Romero, B. Prenkaj, G. Stilo, Robust stochastic graph generator for counterfactual explanations, in: AAAI, 2024.
[15] Y. Zhang, S. B. Yang, A. Khan, C. G. Akcora, Atex-cf: Attack-informed counterfactual explanations for graph neural networks, in: ICLR, 2026.
[16] J. Chen, S. Wu, A. Gupta, R. Ying, D4explainer: in-distribution gnn explanations via discrete denoising diffusion, in: NeurIPS, 2023.
[17] B. Prenkaj, E. Zaradoukas, G. Kasneci, Graph inverse style transfer for counterfactual explainability, in: ICML, 2025.
[18] H. Yuan, H. Yu, J. Wang, K. Li, S. Ji, On explainability of graph neural networks via subgraph explorations, in: ICML, 2021.
[19] J. Li, H. Shomer, et al., Evaluating gnns for link prediction: Current pitfalls and new benchmarking, Adv. in Neural Information Processing Systems 36 (2023).
[20] M. Zhang, Y. Chen, Link prediction based on graph neural networks, Advances in neural information processing systems 31 (2018).
[21] Y. Hao, X. Cao, Y. Fang, X. Xie, S. Wang, Inductive link prediction for nodes having only attribute information, arXiv preprint arXiv:2007.08053 (2020).
[22] X. Huang, M. Galkin, M. M. Bronstein, I. I. Ceylan, Hyper: A foundation model for inductive link prediction with knowledge hypergraphs, arXiv preprint arXiv:2506.12362 (2025).
[23] Z. Huang, et al., Global counterfactual explainer for gnn, in: WSDM, 2023.
[24] Z. Ying, D. Bourgeois, J. You, M. Zitnik, J. Leskovec, Gnnexplainer: Generating explanations for gnns, Advances in neural information processing systems 32 (2019).
[25] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, X. Zhang, Parameterized explainer for graph neural network, in: NeurIPS, 2020.
[26] Z. Qu, D. Gomm, M. Färber, Cody: Counterfactual explainers for dynamic graphs, arXiv preprint arXiv:2403.16846 (2024).
[27] L. Werner, N. Layaïda, P. Genevès, S. Chlyah, Knowledge Enhanced Graph Neural Networks for Graph Completion, in: DSAA, 2023.