A Review of Paper 2: D4Explainer — Research Seminar Data Science Task 1

Daan Brugmans
Radboud University
daan.brugmans@ru.nl

Marieke van Vreeswijk
Radboud University
marieke.vanvreeswijk@ru.nl

1 Objective of the Paper

The primary goal of this paper is to address the challenges associated with generating in-distribution, diverse, and robust explanations for Graph Neural Networks (GNNs). Existing explanation methods for GNNs often struggle with providing effective explanations, especially in the context of counterfactual and model-level scenarios. Counterfactual explanations aim to identify changes in the input graph that lead to different outcomes, specifically focusing on the transition between classes while remaining within the distribution. Model-level explanations, on the other hand, seek to uncover the patterns specific to a particular class, requiring the consideration of multiple instances for a comprehensive understanding. The overarching objective is to improve the interpretability and trustworthiness of GNNs in diverse applications.

2 Proposal of the Paper

The proposed solution, termed D4Explainer [1], introduces a novel approach that leverages denoising diffusion models to generate explanations for GNNs. Their proposal can be divided into counterfactual and model-level explanations.

2.1 Counterfactual Explanations

D4Explainer's counterfactual explanation architecture is visualized in figure 1. In their pursuit of minimal modifications to the original input graph G_0, Chen et al. initiate the counterfactual explanation process by introducing noise to the graph at timestamp t. This noisy graph G_t is then subjected to the denoising model p_θ, which is based on the Provably Powerful Graph Network (PPGN) architecture by Maron et al. [4]. Specifically, the current timestamp t is put through a Multilayer Perceptron (MLP) that returns a vector containing latent timestamp-related information. This vector is then concatenated with the discrete adjacency matrix of the noisy graph A_t and the feature vector of the original input graph X_0. The concatenated representation is used to obtain a dense mask, containing the probabilities of each edge's presence in the reconstructed graph $\tilde{G}_0$. Subsequently, edges deemed irrelevant for the counterfactual explanation (edges with low probabilities in the dense adjacency matrix) are removed. At this stage, a candidate counterfactual explanation has been generated, which undergoes evaluation using both a counterfactual loss $\mathcal{L}_{cf}$ and an in-distribution loss $\mathcal{L}_{dist}$. The counterfactual loss ensures a transition to a different class, while the in-distribution loss ensures adherence to the original data distribution. If a valid counterfactual is not yet found, the process proceeds to the next timestamp by introducing additional noise, repeating iteratively until a valid counterfactual is identified.

2.2 Model-level Explanations

For the generation of model-level explanations, aimed at identifying patterns specific to a particular class, the process begins with the sampling of a purely noisy graph G_T^r, as illustrated in figure 2. The

Preprint. Under review.

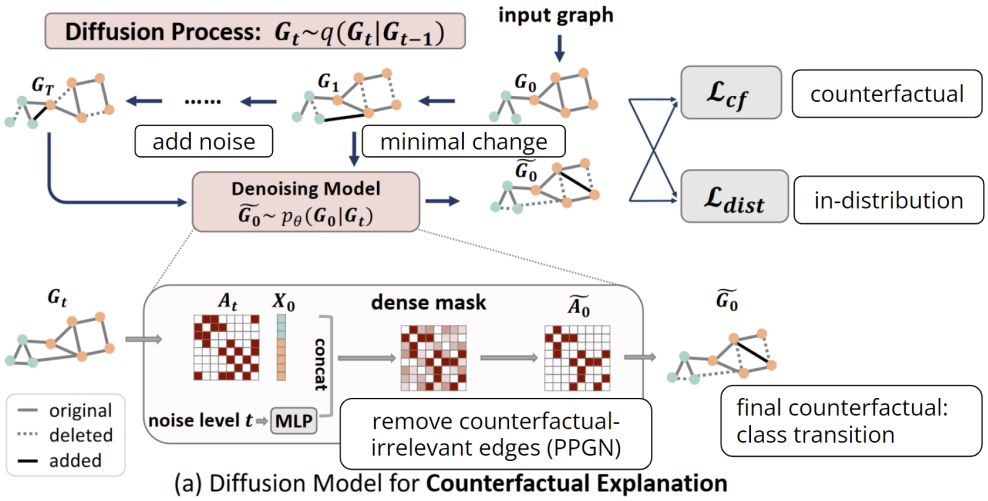

Figure 1: D4Explainer's counterfactual explanation process visualized.

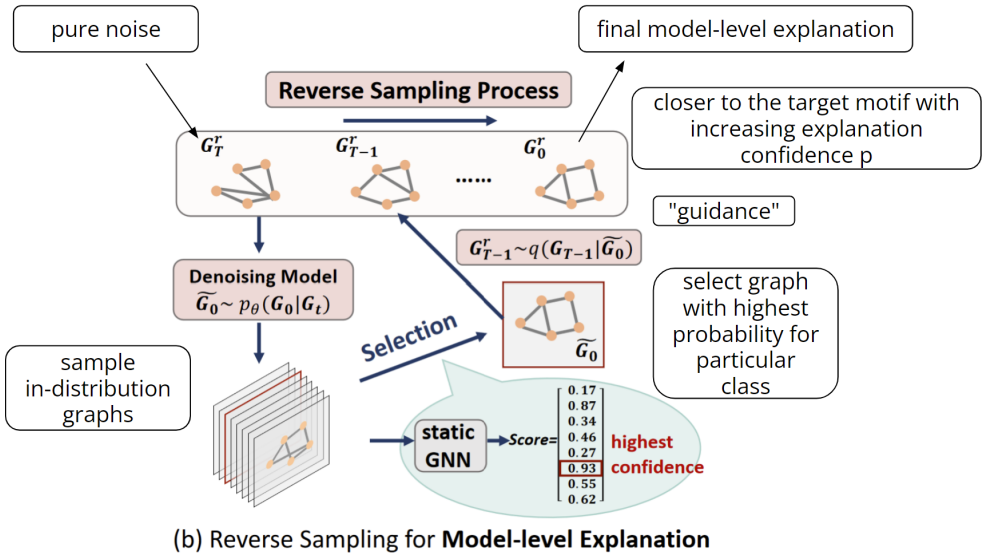

Figure 2: D4Explainer's model-level explanation process visualized.

denoising model p_θ is then utilized to sample multiple in-distribution graphs. Subsequently, these generated graphs undergo evaluation by the static, well-trained GNN for which the explanations are generated, in order to obtain a probability score for each graph concerning the specific class of interest. The graph with the highest probability of belonging to the class of interest is taken to be the best temporary model-level explanation $\tilde{G}_0^r$. This candidate explanation is then taken as the input graph for a forward diffusion process that generates a noisy graph with a noise level/timestamp t that is one level lower than the previous noisy graph G_{T-1}^r. This is called the Reverse Sampling Process and is repeated until timestamp 0 is reached. This approach ensures a progressive approximation towards the target motif, with an increasing level of explanation confidence (denoted as p) during the reverse sampling process. At timestamp 0, representing a completely denoised graph, the model-level explanation is considered to be successfully obtained. This systematic process ensures the extraction of patterns representative of the specified class, contributing to a comprehensive model-level explanation.

3 Evidence Given

D4Explainer's efficacy is thoroughly assessed through a comprehensive evaluation involving various benchmarks and baseline methods. The evaluation encompasses both node classification and graph classification tasks, aiming to validate D4Explainer's proficiency in providing counterfactual and model-level explanations. In this context, synthetic datasets (3 for node classification and 1 for graph classification) and real-world datasets (3 for graph classification) are utilized. Synthetic datasets focus on classifying nodes or graphs based on specific types or shapes, such as cycles, trees, or houses. Real-world datasets, on the other hand, pertain to the representation of molecules as graphs, with atoms serving as nodes.

For counterfactual explanations, a dual assessment is conducted concerning both the in-distribution and counterfactual properties, alongside evaluations of robustness and diversity. Metrics employed include counterfactual accuracy, fidelity, modification ratio, and maximum mean discrepancy. Robustness is measured through Top-K accuracy, emphasizing the ability of counterfactual explanations to remain consistent even in the presence of noise. Although diversity is qualitatively assessed, D4Explainer's unique capability to create edges during the counterfactual explanation process is highlighted, introducing a novel layer of diversity.

Model-level explanations are evaluated qualitatively, focusing on the probability assigned by the GNN for correct class attribution and edge density within the explanation. D4Explainer consistently outperforms baselines in qualitative assessments, demonstrating superior accuracy over existing baselines, achieving over 90% accuracy, across 7 out of 8 datasets. This signifies a substantial advancement in model-level explanation generation, particularly for complex graphs. Furthermore, D4Explainer consistently exhibits lower levels of density in its model-level explanations than existing baselines, showing that it is capable of generating explanations that are simpler to interpret than the baselines' explanations.

In the realm of model-level explanations, D4Explainer maintains consistent superiority over baselines. It produces explanations with higher prediction confidences by the model while concurrently simplifying explanations through reduced edge density. Overall, the qualitative evaluations affirm D4Explainer's robustness and efficacy in providing both counterfactual and model-level explanations.

4 Shoulders of Giants

Chen et al. [1]'s paper builds upon the foundation established by two main prior methods, CF-GNNexplainer [2] and CLEAR [3], dedicated to producing counterfactual explanations for GNNs. While CF-GNNexplainer utilizes edge deletion for counterfactual subgraph generation, CLEAR goes a step further by considering edge addition as well. However, a shared limitation of these methods is the oversight of the distribution constraint for the generated explanations, potentially leading to out-of-distribution and unreliable results. Recognizing the significance of in-distribution considerations is paramount, especially when altering a graph out-of-distribution. In such cases, the model is prone to random guessing since it has never encountered similar instances before. This randomness can lead to unreliable counterfactual explanations, emphasizing the necessity of ensuring explanations align with the original data distribution for interpretability and trustworthiness.

D4Explainer addresses these limitations and underscores the importance of in-distribution counterfactuals. Chen et al.'s novel approach leverages denoising diffusion models for the generation of in-distribution, diverse, and robust counterfactual and model-level explanations for GNNs. The denoising model in D4Explainer is based on Maron et al. [4]'s PPGN architecture and addresses this concern by incorporating distribution constraints during the generation process, enhancing the reliability of counterfactual explanations.

5 Impact

D4Explainer's main impact is the possibility of generating counterfactual and model-level explanations for GNNs that are faithful to the distribution of the input data the GNN was trained on. Prior counterfactual GNN explainability methods, such as those of Lucic et al. [2] and Ma et al. [3], may generate out-of-distribution explanations. When an explanation is out-of-distribution, it may fail to adhere to certain constraints present in the distribution of a dataset. Molecules, for example, must

adhere to specific structures in order to be considered valid. When represented by graphs, prior counterfactual explanation methods may generate molecular structures that could not exist in the real world. D4Explainer ensures that GNNs trained on graphs of the molecule domain can be explained counterfactually using explanations that represent molecular structures that could realistically exist. This, of course, also extends beyond the domain of molecules.

On top of faithfulness to the in-distribution property, D4Explainer also offers explanations that are both robust and diverse. Since D4Explainer uses a discrete denoising diffusion process to generate explanations, it is possible to not only generate many diverse explanations for the same GNN and target class, but to have those explanations be robust despite the presence of noise. The robustness and diversity of D4Explainer's explanations is reflected in its model-level explanation generation process, see figure 2: D4Explainer is capable of generating in-distribution explanations from a purely noisy graph, and does so many times in order to get a diverse set of candidate model-level explanations.

In their paper, Chen et al. [1] themselves state that "[...] D4Explainer can generate more reliable explanations with better in-distribution property, diversity and robustness", and that "[...] the ability to generate counterfactual explanations for GNNs can enhance transparency and interpretability, empowering users to understand and trust the decisions made by these models. By shedding light on the features and interactions that contribute to specific predictions, this work can facilitate the identification of biases, discriminatory patterns, and vulnerabilities present in GNNs".

6 Reproducibility

In assessing the reproducibility of Chen et al. [1]'s work, we take as our guideline *The Machine Learning Reproducibility Checklist* by Pineau et al. [5], which may be accessed at the URL `https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf`.

For all **models** and **algorithms** presented, the authors...

- **do** provide a clear description of the mathematical setting, algorithm, and model. The mathematical setting is described in chapters 3 and 4 and appendices B and C, while the algorithms and the D4Explainer model are described in chapter 4 and appendix D.
- **do** provide a clear description of any assumptions. For example, for the model-level explanation process, the authors state that it is assumed that the GNN is static and well-trained. Limitations are also discussed in appendix F.
- **do** offer an analysis of complexity in paragraphs 4.3 and E.6.

For any **theoretical claim**, the authors **do** provide clear statements *and* complete proofs. The authors' claims are given in the main body of the paper and are supported by partial proofs. Full proofs of their claims can be found in the appendices of their work.

For all **datasets** used, the authors...

- **do** offer the relevant statistics. These are given in appendix E, in table 6.
- **do not** provide the details of train, validation, and test splits.
- **do not** provide an explanation of which data were excluded or preprocessed.
- **do not** provide a link to a downloadable version of the datasets, only references to the papers in which they originated, nor to their simulation environment. However, the authors **do** provide these links in their shared code.
- **do not** provide descriptions for newly collected data, since they do not collect new data in their work.

For all shared **code** related to their work, the authors...

- **do** provide a specification of dependencies by providing a `requirements.txt`.
- **do** provide training code for both GNNs and explaination models.
- **do** provide evaluation code, for robustness and the in-distribution property.
- **do not** provide pre-trained models.

- **do not** provide results in the README, but **do** provide instructions to run their program.

For all reported **experimental results**, the authors...

- **do not** provide the range of hyperparameters considered, but **do** provide the best hyperparameter configuration in appendix E.3.
- **do not** provide the exact number of training and validation runs.
- **do** provide clear definitions of the specific measures used to report results. They may be found in paragraphs 5.3, E.2 and E.4.
- (mostly) **do not** provide results with central tendency and variation.
- **do** provide average runtime for each result, in table 9.
- **do not** provide a description of the computing infrastructure used.

7 Summary of Received Queries and In-Class Discussions

The received queries posted on the Brightspace forum mostly concerned the measures that were (or were not) used. For example, the lack of a measure for the diversity of explanations was questioned. We agree that the lack of a measure for diversity of explanations dampens the strength of the authors' claims regarding D4Explainer's diversity capabilities, and that such a measure should have been included. Multiple queries regarded the application of D4Explainer in real-world scenarios. We argue that the example of molecules given earlier is an example of how D4Explainer can be applied in the real world in, for example, the designing and testing of drugs. In order to compare D4Explainer to real-world explanations, the explanations generated by D4Explainer for a molecule graph dataset could be compared to actual molecular structures known to cause certain phenomena. The main way in which D4Explainer stands out from prior baselines is questioned, which we argue would be faithfulness to the in-distribution property. Finally, the implications of D4Explainer are queried, which we argue would be improved transparency in decision-making of GNN-based systems.

During the in-class discussion, the limitations of D4Explainer came to light. The assumption that a GNN would have to be well-trained for model-level explanations to work properly was addressed, for example; the fact that GNNs can be confidently wrong is a limitation of the proposed method. Further limitations of D4Explainer were also discussed, such as scalability issues on large input graphs and dependency on the specific architecture of the GNN analyzed, preventing generalized explanations that apply to all GNN architectures. Furthermore, we also discussed the results of the author's experiments in more detail, going over the specific scores that were achieved by D4Explainer and the baselines and providing additional examples of D4Explainer's explanation-generation capability.

References

[1] Jialin Chen, Shirley Wu, Abhijit Gupta, and Zhitao Ying. D4explainer: In-distribution explanations of graph neural network via discrete denoising diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=GJtP1ZEzua`.

[2] Ana Lucic, Maartje A. Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 4499–4511. PMLR, 28–30 Mar 2022. URL `https://proceedings.mlr.press/v151/lucic22a.html`.

[3] Jing Ma, Ruocheng Guo, Saumitra Mishra, Aidong Zhang, and Jundong Li. CLEAR: Generative counterfactual explanations on graphs. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=YR-s5leIvh`.

[4] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett,

editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bb04af0f7ecaee4aae62035497da1387-Paper.pdf.

[5] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Lariviere, Alina Beygelzimer, Florence d'Alche Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research(a report from the neurips 2019 reproducibility program). *Journal of Machine Learning Research*, 22(164):1–20, 2021. URL http://jmlr.org/papers/v22/20-303.html.