How Faithful are Self-Explainable GNNs?

Marc Christiansen[*]
CS Department, Aarhus University
mc@marcs.dk

Lea Villadsen[*]
CS Department, Aarhus University
lea@villadsen.one

Zhiqiang Zhong
CS Department, Aarhus University
zzhong@cs.au.dk

Stefano Teso
CIMeC & DISI, University of Trento
stefano.teso@unitn.it

Davide Mottin
CS Department, Aarhus University
davide@cs.au.dk

Abstract

Self-explainable deep neural networks are a recent class of models that can output *ante-hoc* local explanations that are *faithful to the model's reasoning*, and as such represent a step forward toward filling the gap between *expressiveness* and *interpretability*. Self-explainable graph neural networks (GNNs) aim at achieving the same in the context of graph data. This begs the question: *do these models fulfill their implicit guarantees in terms of faithfulness?* In this extended abstract, we analyze the faithfulness of several self-explainable GNNs using different measures of faithfulness, identify several limitations – both in the models themselves and in the evaluation metrics – and outline possible ways forward.

1 Introduction

A number of *self-explainable* deep models for image data have been recently developed that sport *high performance* without giving up on *interpretability* [1–6]. At a high level, they follow a two-level architecture that first maps input images to a set of "concepts" – *i.e.*, interpretable, higher-level representations of the input, such as what objects are visible in the image and their attributes – using a neural network and then infers predictions based on the extracted concepts only. This step is implemented with a transparent inference layer (*e.g.*, a sparse linear map) from which concept-based explanations can be easily derived. This makes local explanations *cheap* to compute, *expressive* [7, 8], and *faithful* to the model's inference process [9]. We refer the reader to [10] for an overview.

These developments have spurred interest into self-explainable GNN architectures that follow a similar setup and aim at achieving the same feat for graph data. A brief overview of these architectures is given in Section 2. This begs the question: *do self-explainable GNN architectures uphold the promise of producing explanations that are faithful to the model's reasoning?*

In this preliminary investigation, we *focus on graph classification* and provide an initial answer to this question for a representative selection of self-explainable GNN architectures. Namely, we empirically analyze the faithfulness of explanations output by one information-constrained architecture (`GISST` [11]) and two prototype-based architectures (`ProtGNN` [12] and `PIGNN` [13]) on four different data sets, and according to different metrics. Our results highlight several interesting phenomena. First, these architectures fail to *guarantee* faithfulness, thus falling short of the original desideratum. Furthermore, their explanations exhibit widely different degrees of faithfulness depending on the data set. Naturally, this hinders trustworthiness. Second, and equally importantly, we find that well-known

[*] Equal contribution.

M. Christiansen et al., How Faithful are Self-Explainable GNNs? (Extended Abstract). Presented at the Second Learning on Graphs Conference (LoG 2023), Virtual Event, November 27–30, 2023.

measures of faithfulness *lack a natural reference* and therefore can paint an incomplete picture of explanation quality. Summarizing, our results raise doubts on the interpretability of self-explainable GNN architectures and call for stricter scrutiny of their properties and of the measures proposed to evaluate them.

2 Post-hoc and Ante-hoc Explanations for Graph Neural Networks

Explainability and faithfulness. Graph Neural Networks (GNNs) are discriminative classifiers that, given an input graph $G = (V, E)$, annotated with node features X, output a conditional distribution over the labels $p_\theta(Y \mid G)$ parameterized by θ. In this abstract, we focus on *graph classification*, in which case a prediction is readily obtained by solving $\operatorname{argmax}_y p_\theta(y \mid G)$. The inference process of GNNs relies on opaque operations – chiefly, message passing of uninterpretable embedding vectors – and as such it is difficult to interpret for laypeople and experts alike. Motivated by this, a number of *post-hoc* explainers have been developed for translating the reasoning process of GNNs into human-understandable terms. Roughly speaking, these algorithms extract an (attributed) *subgraph* $\mathcal{E}$ of G highlighting those elements – nodes, edges, features, or some selection thereof – that are most *responsible* for a GNN's prediction. For details, please see [14–16].

A key desideratum is that of *faithfulness*, that is, explanations should portray an accurate picture of the model's decision making process [15, 17, 18]. Unless explanations are faithful, they can fool users into, *e.g.*, wrongly trusting misbehaving models, and can complicate debugging [8]. Metrics for evaluating faithfulness will be discussed in Section 3. Post-hoc approaches give limited guarantees about faithfulness [14, 15], prompting the development of *self-explainable* GNNs able to produce explanations that are – at least on paper – faithful by design, see [16, Section 3.2].

Self-explainability via information constraints. A first group of approaches encourage interpretability by ensuring the information used for prediction crosses an *information bottleneck*, oftentimes implemented as a sparse subgraph, and read off an explanation from this [19–23]. We focus on a representative of this group, `GISST` [11], a model-agnostic approach for turning any GNN into an information-constrained self-explainable model. In short, `GISST` introduces an attention layer on the model embeddings and encourages this layer – via an appropriate sparsification penalty over the features and the edges of the graph – to focus on a subgraph $\mathcal{E}$ of G (and a subset of features $\mathcal{X}$) capturing all information necessary to infer a label. As for explanations, `GISST` derives the relevance of each edge in the graph from the attention layer, and constructs an explanation subgraph by filtering the most relevant edges that connect a predefined maximum number of nodes.

Self-explainability via structure constraints. Another class of self-explainable GNNs seek interpretability by imposing some form of architectural bias [24–26]. A popular strategy involves augmenting GNNs with prototypes. These are (embeddings of) concrete training examples or parts thereof, and can be used to implement case-based reasoning [27]. One example is `ProtGNN` [12], which first embeds inputs using a GCN [28] or a GAT [29], and then infers a prediction by matching the input graph with learned prototypes. In turn, these prototypes are learned so as to activate highly on training examples corresponding to a specific target class. Explanations are obtained by taking those elements (*i.e.*, nodes, edges) of the input that have triggered the activation of the prototypes most responsible for the prediction. `PIGNN` [13] is similar, in that it infers concept activations based on the similarity between the input graph and learned prototypes. The similarity is however computed at the level of individual node embeddings. Node predictions are then computed directly from the prototype activations, while graph predictions make use of a max pooling operator. `PIGNN` depends on an underlying prototype-based neural network. This can be either ProtoPNet [2] (yielding `PIGNN+P`) or TesNet [30] (yielding `PIGNN+T`). TesNet is a prototype architecture for images that encourages disentanglement between concepts by enforcing orthonormality of the subspaces spanned by the concepts. Both models return the subgraphs that activate the prototypes associated to a given class the most by adapting existing image-based ProtoPNets by substituting convolutional layers with a GCN and adding a max-pooling before the classification layer to aggregate node-prototypes into subgraphs.

3 Empirical Analysis

We address the following research questions: **Q1**: Do self-explainable GNNs guarantee faithfulness in practice? **Q2**: Do existing faithfulness metrics properly gauge the quality of the model's explanations? To this end, we integrated the implementations of `GISST`, `ProtGNN` and `PIGNN` from the original papers into a unified framework built on Pytorch Geometric [31]. We publish the code and the data for reproducibility at `https://anonymous.4open.science/r/SEGNNEval`.

Data sets. We consider four graph classification data sets, cf. Table 1. `BBBP` [32] and `MUTAG` [33, 34] are data sets of chemical compounds labeled according to their blood-brain barrier penetration ability and mutagenicity, respectively. `Ba2Motif` [35] and BAMultiShapes (`BaMS` for short) [36] are synthetic data sets. Similarly to BA-Shapes [37], graphs are obtained by linking a random Barabasi-Albert (BA) graph [38] to one or more motifs that cannot be generated by the BA process. In `Ba2Motif`, the motif is either a house or a 5-node cycle, and it fully determines the label. In `BaMS`, it is either a house, a wheel or a grid, and the label is determined by the presence of exactly two out of three motifs. For the BA data sets, node features encode the number of edges and triangles the node participates in, and the BA subgraphs were generated using a fixed seed, for reproducibility.

Faithfulness metrics. We evaluate faithfulness using three metrics. Let $\mathcal{E}$ be the explanation subgraph induced by the nodes, edges, and features deemed *relevant* by the model, and $\mathcal{C}$ be its complement. *Unfaithfulness* [15] measures how strongly a prediction Y depends on *irrelevant* (i.e., not part of the explanation) nodes, edges, and features in G:

$$\mathsf{Unf} = 1 - \exp\big(- \mathsf{KL}\big(p_\theta(Y \mid G) \parallel p_\theta(Y \mid \mathcal{E})\big)\big) \tag{1}$$

Unfaithfulness is close to zero only if the Kullback-Leibler divergence KL is small, that is, if G and $\mathcal{E}$ yield similar label distributions[2]. *Fidelity* [17, 18] includes two metrics assessing to what degree explanations are *necessary* (Fid^+) and *sufficient* (Fid^-). These are defined as[3]:

$$\mathsf{Fid}^+ = \big|\mathbb{1}\{\widehat{y} = y\} - \mathbb{1}\big\{\widehat{y}^{\mathcal{C}} = y\big\}\big|, \quad \mathsf{Fid}^- = \big|\mathbb{1}\{\widehat{y} = y\} - \mathbb{1}\big\{\widehat{y}^{\mathcal{E}} = y\big\}\big| \tag{2}$$

Here, $\widehat{y}^{\mathcal{E}} = \operatorname{argmax}_y \; p_\theta(y \mid \mathcal{E})$ and similarly for $\widehat{y}^{\mathcal{C}}$. High Fid^+ means that $\mathcal{E}$ is *necessary*, in the sense that the prediction does change when the model is fed the irrelevant subgraph $\mathcal{C}$ only, while low Fid^- means the explanation is *sufficient*, in that the prediction remains unchanged when considering the relevant subgraph $\mathcal{E}$ only.

These measures differ in that Fid^+ and Fid^- look at the hard predictions and depend on the ground-truth label, whereas Unf considers the change in label distribution only. As such, it can be viewed as a smoother version of Fid^- and behaves similarly in our experiments.

Q1: Faithfulness of self-explainable GNNs is not perfect and varies across tasks. Table 2 shows the average result of 5 runs of the algorithm with 80/10/10 train, validation, and test split. Most architectures attained good ($> 80\%$) and stable ($< 10\%$ std. dev.) graph classification accuracy on all data sets. The exception is `ProtGNN`, which fared very poorly on the synthetic tasks. Since it tends to over-predict the majority class (especially in `Ba2Motif`), faithfulness measures are unreliable. Hence, we excluded `ProtGNN` from our analysis for these data sets.

In stark contrast, their faithfulness –indicated as $\mathsf{Unf}(\mathcal{E})$, $\mathsf{Fid}^+(\mathcal{E})$, and $\mathsf{Fid}^-(\mathcal{E})$ – is far from perfect. All models tend to perform better at Unf and Fid^- (best results are 0 and 0.07, respectively, the lower the better) than at Fid^+ (best result is 0.58, the higher the better), indicating they are better suited at generating *necessary*, rather than *sufficient*, explanations. Note that in the case of `ProtGNN` on `MUTAG`, Unf might be zero due to negligible changes in the prediction vector and rounding in the results, while Fid^-, although being very low, is 0.1 indicating that some of the predictions swaps classes while the overall distribution is stable. This example shows how Fid^- and and Unf captures the same phenomenon but with different granularity.

Table 1: Overview of data sets used in our experiments. Numbers are averaged.

Dataset	# Graphs	Avg. # Nodes	Avg. # Edges	# Features	# Classes
`MUTAG` [33, 34]	188	17.9	39.6	7	2
`BBBP` [32]	2039	24.0	51.9	9	2
`Ba2Motif` [35]	1000	30.0	57.8	2	2
`BaMS` [36]	1000	40.0	87.4	2	2

[2] In [15], $\mathcal{E}$ is obtained by zeroing-out the irrelevant features from G. In this abstract, $\mathcal{E}$ is obtained by also deleting irrelevant nodes and edges, so as to take the relevance of topological elements into account.

[3] Our definition pertains to graph classification. Fidelity is also defined for node classification [17].

Table 2: Results for all models and data sets. We report average and std. dev. of classification accuracy, faithfulness of the model explanations $\mathcal{E}$, and faithfulness of random subgraphs $\mathcal{R}$. **Bold** entries indicate best result, **red** ones that the random subgraph is at least as faithful as the model's explanation, **orange** ones that it is at most 0.10 worse, and **green** the rest.

Dataset	Model	Acc (↑)	Unf($\mathcal{E}$)(↓)	Unf($\mathcal{R}$)(↓)	Fid$^-$($\mathcal{E}$)(↓)	Fid$^-$($\mathcal{R}$)(↓)	Fid$^+$($\mathcal{E}$)(↑)	Fid$^+$($\mathcal{R}$)(↑)
BBBP	GISST	$\mathbf{0.87 \pm 0.01}$	0.22 ± 0.15	0.15 ± 0.02	0.29 ± 0.25	0.21 ± 0.04	0.16 ± 0.01	0.20 ± 0.03
	PIGNN+P	0.86 ± 0.01	$\mathbf{0.07 \pm 0.02}$	0.16 ± 0.03	$\mathbf{0.07 \pm 0.03}$	0.20 ± 0.03	$\mathbf{0.33 \pm 0.06}$	0.22 ± 0.03
	PIGNN+T	0.86 ± 0.01	0.10 ± 0.01	0.15 ± 0.01	0.14 ± 0.02	0.22 ± 0.02	0.30 ± 0.05	0.27 ± 0.03
	ProtGNN	0.85 ± 0.01	0.15 ± 0.08	0.27 ± 0.05	0.14 ± 0.02	0.23 ± 0.05	0.21 ± 0.06	0.20 ± 0.05
MUTAG	GISST	$\mathbf{0.86 \pm 0.04}$	0.11 ± 0.12	0.13 ± 0.06	0.31 ± 0.17	0.41 ± 0.16	$\mathbf{0.50 \pm 0.19}$	0.38 ± 0.13
	PIGNN+P	0.82 ± 0.02	0.12 ± 0.08	0.18 ± 0.08	0.26 ± 0.26	0.59 ± 0.08	0.42 ± 0.17	0.52 ± 0.22
	PIGNN+T	0.84 ± 0.07	0.14 ± 0.13	0.08 ± 0.07	0.40 ± 0.30	0.51 ± 0.12	0.37 ± 0.05	0.40 ± 0.15
	ProtGNN	0.82 ± 0.02	$\mathbf{0.00 \pm 0.00}$	0.00 ± 0.00	$\mathbf{0.10 \pm 0.07}$	0.18 ± 0.04	0.34 ± 0.05	0.20 ± 0.05
Ba2Motif	GISST	0.98 ± 0.02	0.13 ± 0.16	0.32 ± 0.09	$\mathbf{0.14 \pm 0.16}$	0.32 ± 0.08	0.51 ± 0.02	0.52 ± 0.02
	PIGNN+P	$\mathbf{1.00 \pm 0.00}$	0.50 ± 0.04	0.42 ± 0.01	0.59 ± 0.06	0.50 ± 0.00	0.01 ± 0.01	0.50 ± 0.00
	PIGNN+T	0.98 ± 0.01	$\mathbf{0.03} \pm 0.02$	0.08 ± 0.05	0.21 ± 0.06	0.57 ± 0.06	$\mathbf{0.58 \pm 0.14}$	0.56 ± 0.08
	ProtGNN	0.51 ± 0.11	–	–	–	–	–	–
BaMS	GISST	0.79 ± 0.13	$\mathbf{0.11} \pm 0.07$	0.13 ± 0.06	$\mathbf{0.20 \pm 0.13}$	0.27 ± 0.12	0.42 ± 0.20	0.39 ± 0.19
	PIGNN+P	$\mathbf{0.96 \pm 0.00}$	0.32 ± 0.00	0.32 ± 0.00	0.54 ± 0.00	0.54 ± 0.00	0.54 ± 0.00	0.54 ± 0.00
	PIGNN+T	0.92 ± 0.00	0.13 ± 0.01	0.30 ± 0.00	0.46 ± 0.01	0.56 ± 0.00	$\mathbf{0.56 \pm 0.00}$	0.56 ± 0.00
	ProtGNN	0.64 ± 0.10	–	–	–	–	–	–

Moreover, explanation faithfulness varies widely across learning tasks. For instance, PIGNN+P attains excellent Unf and Fid$^-$ in BBBP (0.07 for both), but fares poorly in Ba2Motif (0.50 and 0.59, respectively). A 10-15% difference in unfaithfulness also exists for the other models. As for Fid$^+$, the delta across data sets is 34% for GISST, 0.53 for PIGNN+P, 0.18 for PIGNN+T, and 0.13 for ProtGNN. Summarizing, no model yields consistent faithfulness across tasks for any of the metrics. This makes it difficult to fully trust the explanations output by these models and raises doubts on their applicability to high-stakes tasks.

Q2: Absolute faithfulness measures can be misleading. It is true, however, that faithfulness is excellent for some combinations of models and tasks. This is especially the case for ProtGNN on MUTAG and for PIGNN+T on Ba2Motif, which both attain near-perfect Unf and Fid$^-$. Here, we argue that the absolute values in faithfulness can lead to erroneous interpretations. In a nutshell, faithfulness measures require a baseline method to establish the improvement. We offer such a term of reference by compare each measure by the same measure computed on a random explanation model that generates subgraphs $\mathcal{R}$ of G of the same size of the explanation $\mathcal{E}$. That is, we replace $\mathcal{E}$ with $\mathcal{R}$ in Eq. (1) and Eq. (2). We indicate the results as Unf($\mathcal{R}$), Fid$^-$($\mathcal{R}$) and Fid$^+$($\mathcal{R}$) in Table 2.

We can see that oftentimes the model's explanations are in fact more faithful than the random ones, especially for MUTAG data. Yet, in many cases the models produce worse or comparable explanations to a random graph. Some results are especially interesting. For instance, ProtGNN is the most faithful method in MUTAG, while our analysis displays the same behavior on random explanations. Another interesting case is that of GISST, which performs the best on BaMS while exhibiting Unf 45% worse than random. In conclusion, while faithfulness measures offer an intuitive idea of the model's explanations, they provide only a partial view of their overall quality.

4 Discussion and Conclusion

Our initial results indicate that, at least when it comes to graph classification, it is *hard to assert that self-explainable GNNs are faithful by design*. Specifically, faithfulness of these models seems to be very data set dependent. This becomes even more clear when we compare the faithfulness of their explanations to that of a completely uninformed baseline.

Naturally, our results are preliminary. In follow-up work, we will verify to what extent they carry over to other self-explainable GNN architectures, like [23], and to extend our analysis to *node classification*. Another major issue is whether the *concepts* (*e.g.*, prototypes) learned by these models are in fact guaranteed to be interpretable. Yet, this issue is shared by self-explainable models for image data as well, and needs a separate investigation to be understood properly [5].

References

[1] David Alvarez-Melis and Tommi S Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7786–7795, 2018. 1

[2] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. *Advances in Neural Information Processing Systems*, 32:8930–8941, 2019. 2

[3] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[4] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

[5] Emanuele Marconato, Andrea Passerini, and Stefano Teso. Glancenets: Interpretabile, leak-proof concept-based models. In *NeurIPS*, 2022. 4

[6] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Liò, and Mateja Jamnik. Concept embedding models. 2022. 1

[7] Subbarao Kambhampati, Sarath Sreedharan, Mudit Verma, Yantian Zha, and Lin Guan. Symbols as a lingua franca for bridging human-ai chasm for explainable and advisable ai systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12262–12267, 2022. 1

[8] Stefano Teso, Öznur Alkan, Wolfang Stammer, and Elizabeth Daly. Leveraging explanations in interactive machine learning: An overview. *Frontiers in Artificial Intelligence*, 2023. 1, 2

[9] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. 1

[10] Gesina Schwalbe. Concept embedding analysis: A review. *arXiv preprint arXiv:2203.13909*, 2022. 1

[11] Chris Lin, Gerald J Sun, Krishna C Bulusu, Jonathan R Dry, and Marylens Hernandez. Graph neural networks including sparse interpretability. *arXiv preprint arXiv:2007.00119*, 2020. 1, 2

[12] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. Protgnn: Towards self-explaining graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9127–9135, 2022. 1, 2

[13] Alessio Ragno, Biagio La Rosa, and Roberto Capobianco. Prototype-based interpretable graph neural networks. *IEEE Transactions on Artificial Intelligence*, 2022. 1, 2

[14] Antonio Longa, Steve Azzolin, Gabriele Santin, Giulia Cencetti, Pietro Liò, Bruno Lepri, and Andrea Passerini. Explaining the explainers in graph neural networks: a comparative study. *arXiv preprint arXiv:2210.15304*, 2022. 2

[15] Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating explainability for graph neural networks. *Scientific Data*, 10(1):144, 2023. 2, 3

[16] Jaykumar Kakkad, Jaspal Jannu, Kartik Sharma, Charu Aggarwal, and Sourav Medya. A survey on explainability of graph neural networks. *arXiv preprint arXiv:2306.01958*, 2023. 2

[17] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5): 5782–5799, 2022. 2, 3

[18] Kenza Amara, Zhitao Ying, Zitao Zhang, Zhichao Han, Yang Zhao, Yinan Shan, Ulrik Brandes, Sebastian Schemm, and Ce Zhang. Graphframex: Towards systematic evaluation of explainability methods for graph neural networks. In *Learning on Graphs Conference*, pages 44–1. PMLR, 2022. 2, 3

[19] Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*, pages 15524–15543. PMLR, 2022. 2

[20] Siqi Miao, Yunan Luo, Mia Liu, and Pan Li. Interpretable geometric deep learning via learnable randomness injection. In *The Eleventh International Conference on Learning Representations*, 2022.

[21] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. Graph information bottleneck for subgraph recognition. In *International Conference on Learning Representations*, 2020.

[22] Junchi Yu, Jie Cao, and Ran He. Improving subgraph recognition with variational graph information bottleneck. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19396–19405, 2022.

[23] Lucie Charlotte Magister, Pietro Barbiero, Dmitry Kazhdan, Federico Siciliano, Gabriele Ciravegna, Fabrizio Silvestri, Mateja Jamnik, and Pietro Lio. Encoding concepts in graph neural networks. *arXiv preprint arXiv:2207.13586*, 2022. 2, 4

[24] Enyan Dai and Suhang Wang. Towards self-explainable graph neural network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 302–311, 2021. 2

[25] Enyan Dai and Suhang Wang. Towards prototype-based self-explainable graph neural network. *arXiv preprint arXiv:2210.01974*, 2022.

[26] Aosong Feng, Chenyu You, Shiqiang Wang, and Leandros Tassiulas. Kergnns: Interpretable graph neural networks with graph kernels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6614–6622, 2022. 2

[27] Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. *Advances in neural information processing systems*, 27, 2014. 2

[28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 2

[30] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *ICCV*, pages 895–904, 2021. 2

[31] PyG team. Pytorch geometric. URL `https://pytorch-geometric.readthedocs.io/en/stable/`. 3

[32] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018. 3

[33] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2):786–797, 1991. 3

[34] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020. 3

[35] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631, 2020. 3

[36] Steve Azzolin, Antonio Longa, Pietro Barbiero, Pietro Lio, and Andrea Passerini. Global explainability of gnns via logic combination of learned concepts. In *The Eleventh International Conference on Learning Representations*, 2022. 3

[37] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019. 3

[38] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002. 3