ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

XAI and Bias of Deep Graph Networks

Michele Fontanesi, Alessio Micheli and Marco Podda *

University of Pisa - Department of Computer Science
Largo B. Pontecorvo 3, 56127 Pisa - Italy

Abstract. Generalization in machine learning involves introducing inductive biases that restrict the solution space of the learning problem, allowing for the inductive leap. In this paper, we show the existence of different inductive biases between convolutional and recursive Deep Graph Networks (DGN) by applying Explainable AI (XAI) methods as model inspection techniques. We show that different architectures can perfectly solve the given tasks by learning different labelling policies. Our results promote the usage of different architectures to address a task and raise warnings on the assessment of XAI techniques as their benchmarks may contain more ground truths than those provided.

1 Introduction

Deep Graphs Networks (DGNs) [1] are neural networks able to directly learn a function on complex graph-structured data. However, while able to achieve high performance, DGNs are usually regarded as black-boxes, since their complexity prevents recognizing the label assignment policies. As a consequence, to better understand their functioning and ensure their trustworthiness [2], different techniques to analyze DGNs' inner workings have been developed and collected as part of the field of XAI [3] for DGNs [4]. In this work, we deploy XAI techniques for DGNs as model inspection methods to reveal the presence of different inductive biases in different message-passing-based DGNs. Specifically, we focus our attention on recursive and convolutional DGNs as their architectural differences may focus on different aspects of the problem causing the learning of different policies to solve graph classification tasks [5, 6]. Our work empirically tests the following hypothesis: (i) architectures characterized by different inductive biases can encode in their opaque set of weights different label assignment policies: human-intelligible sets of rules able to associate each graph with its label; (ii) different policies may match different ground truth (GT) explanations when processed by XAI algorithms. We empirically verify our hypothesis using the plausibility metric, which quantifies the adherence of the computed explanations to the GTs, as a proximity measure between the label assignment policy learned by the models and the one used to label the XAI-ready datasets. We interpret discrepancies between plausibility metrics across DGNs as indicators of different learned label assignment policies and consequently of different inductive biases [7]. Our experiments, performed on multiple combinations of DGNs architectures, XAI datasets, and explainers, prove our hypothesis and

*Research partly funded by PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - ”FAIR - Future Artificial Intelligence Research” - Spoke 1 ”Human-centered AI”, funded by the European Commission under the NextGeneration EU programme.

reveal the existence of a second label assignment policy for three of the four tested datasets. Moreover, our results show that recursive DGNs feature an inductive bias aligned with this second policy; revealing a rich landscape of inductive biases and, in turn, multiple useful ways to learn and generalize on graph data. The contributions of this work are twofold: (i) We empirically show that different DGN architectures may learn different policies to solve the same problem, which might be attributed to the existence of different inductive biases for message-passing-based DGNs; (ii) We prove the existence of at least a second ground truth in commonly used datasets to benchmark XAI attribution methods. This raises concerns about the performance assessment of XAI techniques as low-quality explanation metrics may be due to using a wrong assumed GT with architectures that show very high predictive capabilities.

2 Background

2.1 Deep Graph Networks

Let $G = (V, E)$ be a graph with $V = \{v_1, ..., v_{N_G}\},\ N_G = |V|$ its set of vertices, $E \subseteq (V \times V) = \{(u, v) \mid u, v \in V\}$ its set of edges, and $\mathbf{A} \in \mathbb{R}^{N_G \times N_G}$ its *adjacency matrix*. We define as $\mathbf{x}_v \in \mathbb{R}^d,\ d \in \mathbb{N}$ the feature vector of a node v ($\mathbf{X} \in \mathbb{R}^{N_G \times d}$ in matrix form). A DGN is a parameterized function able to learn a mapping between input graphs $G \in \mathcal{G}$ and their associated labels $y \in \mathcal{C}$ by adhering to the message passing paradigm; a procedure that updates node embeddings $\mathbf{h}_v$ at each iteration l. This paper distinguishes two classes of Deep Graph Networks: convolutional and recursive. Convolutional DGNs map each iteration l of the message-passing paradigm to a different layer of the architecture; creating deep models. In this work, we focus on the Graph Isomorphism Network (GIN) [8]:

$$\mathbf{H}^{l+1} = \mathrm{MLP}\left((1 + \epsilon^{l+1})\mathbf{H}^l + \mathbf{A}\mathbf{H}^l\right) \tag{1}$$

and on the Graph Convolutional (GC) [9] operator:

$$\mathbf{H}^{l+1} = f(\mathbf{W}_1^l \mathbf{H}^l + \mathbf{W}_2^l \hat{\mathbf{A}} \mathbf{H}^l) \tag{2}$$

In GIN, MLP is a multilayer perceptron, and ϵ is a learnable or fixed parameter. In GC, $\mathbf{W}_1^l$ and $\mathbf{W}_2^l$ are learnable parameters and $\hat{\mathbf{A}}$ is a weighted adjacency matrix and f is an activation function. In both architectures $\mathbf{H}^1 = \mathbf{X}$. Recursive DGNs, instead, map each iteration l of the message-passing paradigm to an iterative step of a recursive layer. In this work, we focus on the Graph Echo State Networks (GESN) [10, 11], an efficient recursive DGN based on Reservoir Computing [12]:

$$\mathbf{H}^{l+1} = f(\mathbf{X}\bar{\mathbf{W}}^{l+1} + \mathbf{A}\mathbf{H}^l \mathbf{W}^{l+1}) \tag{3}$$

where $\mathbf{H}^1 = \mathbf{X}$, f is an activation function, $\bar{\mathbf{W}}^{l+1}$ is a weight matrix introducing residual connections, and $\mathbf{W}^{l+1}$ is the recursive weight matrix. The efficiency and further specific bias of this architecture come from its untrained weight matrices $\bar{\mathbf{W}}^{l+1}$ and $\mathbf{W}^{l+1}$ whose careful initialization determines the creation of a contractive/Markovian dynamical system able to provide meaningful node embeddings to solve a task.

2.2 XAI attribution methods

In this work, we use local post-hoc XAI techniques that yield an *importance score* to each node in a graph in the form of a mask $\hat{\mathbf{t}} \in \mathbb{R}^{N_G}$. Among the many methods that exist in the literature [13, 4], we focus on: (i) GNNExplainer (GNNE) [14], a perturbation-based method which optimizes $\hat{\mathbf{t}}$ by maximizing the mutual information between the original model output and the masked one; (ii) Integrated Gradients (IG) [15], a gradient-based method which computes the contributions of the input features to the output prediction relatively to a baseline point that encodes the complete absence of information; (iii) CAM [16], a DGN-specific technique based on the observation that the output logits of a DGN are the weighted sum of the contribution of each node. These contributions are the importance scores. In addition, we include, as a baseline, a random explainer returning random importance scores for each input graph.

3 Method

We use XAI techniques as model inspection methods to understand whether different DGN architectures learn different label assignment policies. We base our experiments on four different XAI benchmark datasets for which the label assignment policies are known, and corresponding GT explanations $\mathbf{t}$ for each input graph are provided. We quantify the overlap between the retrieved explanation masks $\hat{\mathbf{t}}$ and the GT explanations $\mathbf{t}$ with the plausibility metric which computes the AUROC score between $\hat{\mathbf{t}}$ and $\mathbf{t}$ [12]. Specifically, AUROC computes the Area Under the Receiver Operating Characteristics curve which is obtained by plotting the values of True Positive Rate (TPR) and False Positive Rates (FPR) while varying the classification threshold. As a consequence, it measures adherence of a retrieved explanation to its GT for a single input graph. We interpret an average low value of plausibility across multiple samples as a discrepancy indicator between a model learned label assignment policy and the one used to generate each dataset. As a further step, we manually analyze the retrieved explanations trying to understand which policy each model learns to solve the tasks.

4 Experiments and discussion

We tested our approach on four different synthetic binary classification graph datasets with known label assignment policies. In particular, the BA2Motif datasets assign class 1 to Barabási-Albert (BA) graphs linked to a house motif and class 0 to BA graphs linked to a 5-nodes cycle motif; BA2grid assigns class 1 to BA graphs linked to a 3x3 grid motif and class 0 to plain BA graphs; GridHouse assigns class 1 to BA graphs linked to a 3x3 grid and a house motif, class 0 to BA graphs linked to either the grid or the house; finally HouseColors assigns class 1 to BA graphs linked to at least one green house motif and class 0 to BA graphs linked to at least one blue house motif. This latter dataset is the only one with meaningful node feature vectors as they one-hot encode colors (blue, green, red); the others have constant and meaningless input features. We partitioned each dataset into training (80%) and test (20%) sets stratifying the splits following the label distributions and we trained a GESN, GIN, and GC model performing model selection with 5-fold cross-validation on the training set and model assessment on the hold-out test set. The best model of each trained architecture achieved perfect performance over training, validation, and test splits (accuracy of 1), meaning that a perfect label assignment policy was learned by each model. Afterward, we applied the XAI attribution methods and computed the plausibility metric to measure the alignment between the learned label assignment policies and the ones used to generate each dataset.

	RandomExplainer			GNNExplainer			IntegratedGradient			CAM		
	GESN	GIN	GC	GESN	GIN	GC	GESN	GIN	GC	GESN	GIN	GC
BA2grid	0.495	0.495	0.495	0.557	0.545	0.564	0.808	0.863	0.904	0.806	0.954	1.0
BA2Motif	0.512	0.512	0.512	0.367	0.704	0.843	0.776	1.0	0.893	0.769	0.911	0.843
GridHouse	0.505	0.505	0.505	0.602	0.543	0.599	0.777	0.834	0.823	0.696	0.903	0.953
HouseColors	0.505	0.505	0.505	0.887	0.517	0.843	0.963	0.994	0.947	0.994	0.99	0.996

Table 1: Plausibility of each model and explainer pair with respect to the label assignment policies used to generate each dataset.

Results, as introduced in Table 1, allow for the examination of multiple factors. First, GNNExplainer-based plausibility values are most often closer to the ones obtained with the RandomExplainer than to those obtained with IG or CAM. As a consequence, we did not further analyze its explanations as we consider them unreliable. Second, even if all models were able to perfectly distinguish the two classes, only the pairs IG-GIN on BA2Motif and CAM-GC on BA2grid perfectly retrieved the expected explanation (plausibility value equals 1). Lastly, the plausibility values obtained with IG and CAM on the HouseColor dataset all reach up to 0.94. We consider this result an indicator that all tested architectures learned the label assignment policy used to generate the dataset. However, this consideration does not hold for the other three datasets where consistent gaps have been found between the plausibility values associated with GESN and the ones associated with GIN and GC. We traced back this behavior to the existence of a secondary label assignment policy. Specifically, it is possible to generate perfect classifiers for the BA2grid, BA2Motif, and GridHouse datasets by only using as a feature the average degree of the input graphs. In Table 2 we show that the maximum average degree characterizing class 0 graphs is always lower than the minimum average degree characterizing class 1 graphs for these three datasets; proving that a threshold between these two values is enough to distinguish the classes. Moreover, as the minimum average degree for class 1 graphs is always above 2, we can associate the novel policy with a new set of GT explanations. Specifically, we identify the nodes with a degree greater than 3 as relevant, since they move the average toward class 1. Plausibility values for the novel GT are shown in Table 3 and they clearly reveal that GESN was solving the BA2grid, BA2Motif, and GridHouse tasks by learning this second policy. In fact, all values based on CAM and IG increased

	Class 0		Class 1	
	min	max	min	max
BA2grid	1.87	1.93	2.20	2.4
BA2Motif	2	2	2.08	2.08
HouseColors	2.03	2.35	2.03	2.35
GridHouse	2.06	2.3	2.34	2.5

Table 2: Average degrees across all datasets grouped by target class.

up to a score of 1 with the only exception of IG-GESN on the GridHouse dataset (0.987). Moreover, the perfect scores achieved by the pairs IG-GIN on BA2Motif and CAM-GC on the BA2grid datasets significantly dropped; a sign that GIN and GC were effectively solving these tasks based on the first label assignment policy. In addition, the CAM-based scores generally decreased in contrast to the IG-based scores. We attribute this latter behavior to the usage of the gradient as a mean to compute importance scores for nodes with meaningless input features as in these tasks the most influential nodes on the model output would be the ones having a higher degree.

	RandomExplainer			Integrated Gradient			CAM		
	GESN	GIN	GC	GESN	GIN	GC	GESN	GIN	GC
BA2grid	0.49	0.49	0.49	1.0	1.0	0.952	1.0	0.822	0.731
BA2Motif	0.496	0.496	0.496	1.0	0.412	0.958	1.0	0.812	0.965
GridHouse	0.498	0.498	0.498	0.987	1.0	1.0	1.0	0.847	0.9

Table 3: Plausibility of each model and explainer pair with respect to the newly discovered label assignment policy.

We interpret these results as experimental proof that, for these datasets with meaningless node features, recursive (with the GESN implementation), and convolutional architecture can solve the same task by learning different label assignment policies that we attribute to the existence of different inductive biases.

5 Conclusions

In this work, we used XAI attribution methods as model inspection techniques to empirically verify the existence of different inductive biases in convolutional and recursive DGNs. Our results show that three different tasks can be solved based on two different label assignment policies and that recursive DGNs (GESN) feature an inductive bias that makes them learn one of such policies while the other is preferred by convolutional (GIN, GC) DGNs; revealing multiple opportunities to learn and generalize on graph data. This research impacts the Machine Learning field for graph-structured data and the XAI field for DGNs. In particular, showing that different message-passing-based architectures can learn different label assignment policies should encourage practitioners to test multiple architectural variants to solve a given problem and use XAI techniques to check whether those variants have learned different policies to solve the task. On the XAI field, instead, our results raise concerns about the current benchmarking processes of XAI attribution methods as even simple synthetic tasks may involve multiple GT explanations, some of which may not be known but could be retrieved by an explainer. Consequently, some explainers may underperform with some architectures if the used GT does not match the learned label assignment procedure. Future works include the extension to real-world datasets, other architectures, and XAI attribution methods.

References

[1] Davide Bacciu, Federico Errica, Alessio Micheli, and Marco Podda. A gentle introduction to deep learning for graphs. *Neural Networks*, 129, 2020.

[2] Luca Oneto, Nicolò Navarin, Battista Biggio, Federico Errica, and Alessio Micheli et al. Towards learning trustworthily, automatically, and with guarantees on graphs: An overview. *Neurocomputing*, 493, 2022.

[3] David Gunning and David Aha. DARPA's explainable artificial intelligence (XAI) program. *AI magazine*, 40(2), 2019.

[4] H. Yuan, H. Yu, S. Gui, and S. Ji. Explainability in Graph Neural Networks: A Taxonomic Survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 45(05), 2023.

[5] Antonio Longa, Steve Azzolin, Gabriele Santin, Giulia Cencetti, and Pietro Liò et al. Explaining the explainers in graph neural networks: a comparative study. *arXiv:2210.15304*, 2022.

[6] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, and Bo Zong et al. Parameterized explainer for graph neural network. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., (2020).

[7] Tom M Mitchell. The need for biases in learning generalizations. *Rutgers CS tech report*, 1980.

[8] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[9] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, and Jan Eric Lenssen et al. Weisfeiler and leman go neural: Higher-order graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), Jul. 2019.

[10] Claudio Gallicchio and Alessio Micheli. Graph echo state networks. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, (2010).

[11] Claudio Gallicchio and Alessio Micheli. Fast and deep graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, (2020).

[12] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, and Chengqi Zhang et al. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1), 2021.

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

[14] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating Explanations for Graph Neural Networks. In *NeurIPS*, 2019.

[15] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017.

[16] Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.