
Global Explainability of GNNs via Logic Combination of Learned Concepts

Anonymous Author(s)

Anonymous Affiliation

Anonymous Email

Abstract

1 While instance-level explanation of GNN is a well-studied problem with plenty of
2 approaches being developed, providing a *global* explanation for the behaviour of a
3 GNN is much less explored, despite its potential in interpretability and debugging.
4 Existing solutions either simply list local explanations for a given class, or generate
5 a synthetic prototypical graph with maximal score for a given class, completely
6 missing any combinatorial aspect that the GNN could have learned. In this work
7 we propose GLGExplainer (Global Logic-based GNN Explainer), the first Global
8 Explainer capable of generating explanations as arbitrary Boolean combinations
9 of learned graphical concepts. GLGExplainer is a fully differentiable architecture
10 that takes local explanations as inputs and combines them into a logic formula
11 over graphical concepts, represented as clusters of local explanations. Contrary
12 to existing solutions, GLGExplainer manages to provide accurate and human-
13 interpretable global explanations in both synthetic and real world datasets.
14

15 1 Introduction

16 Graph Neural Networks (GNNs) have become increasingly popular for predictive tasks on graph
17 structured data. However, as many other deep learning models, their inner working remains a black
18 box. The ability to understand the reason for a certain prediction represents a critical requirement for
19 any decision-critical application, thus representing a big issue for the transition of such algorithms
20 from benchmarks to real-world critical applications.

21 Over the last years, many works proposed Local Explainers [1–9] to explain the decision process
22 of a GNN in terms of factual explanations often represented as subgraphs for each sample in the
23 dataset. Overall, they shed light over *why* the network predicted a certain value for a specific input
24 sample. However, they still lack a global understanding of the model. Global Explainers, on the other
25 hand, are aimed at capturing the behaviour of the model as a whole, abstracting individual noisy local
26 explanations in favor of a single robust overview of the model. However, despite their potential in
27 interpretability and debugging little has been done in this direction [10]. GLocalX [11] is a general
28 solution to produce global explanations of black-box models by hierarchically aggregating local
29 explanations into global rules via an heuristic-based iterative procedure. This solution is however not
30 readily applicable to GNNs as it requires local explanations to be expressed as logical rules. Yuan et
31 al. [10] proposed to frame the Global Explanation problem for GNN as a form of input optimization,
32 similarly as done for some vision models [12], using policy gradient to generate synthetic prototypical
33 graphs for each class. The approach requires prior domain knowledge, which is not always available,
34 to drive the generation of valid prototypes. Additionally, it cannot identify any compositionality in
35 the returned explanation, and has no principled way to generate alternative explanations for a given
36 class.

37 Concept-based Explainability [13–15] is a parallel line of research where explanations are constructed
38 using “concepts” i.e., intermediate, high-level and semantically meaningful units of information com-
39 monly used by humans to explain their decisions. Concept Bottleneck Models [16] and Prototypical
40 Part networks [17] are two popular architectures that leverage concept learning to learn explainable-

41 by-design neural networks. Both approaches have been recently adapted to GNNs [18, 19]. However,
 42 these solutions are not conceived for explaining already learned GNNs.

43 **Our contribution** consists in the first Global Explainer for GNNs which *i)* provides a Global
 44 Explanation in terms of logic formulas, extracted by combining in a fully differentiable manner
 45 graphical concepts derived from local explanations; *ii)* is faithful to the data domain, i.e., the logic
 46 formulas, being derived from local explanations, are intrinsically part of the input domain without
 47 requiring any prior knowledge. We validated our approach on both synthetic and real-world datasets,
 48 showing that our method is able to accurately summarize the behaviour of the model to explain, while
 49 providing explanations in terms of concise logic formulas.

50 2 Proposed Method

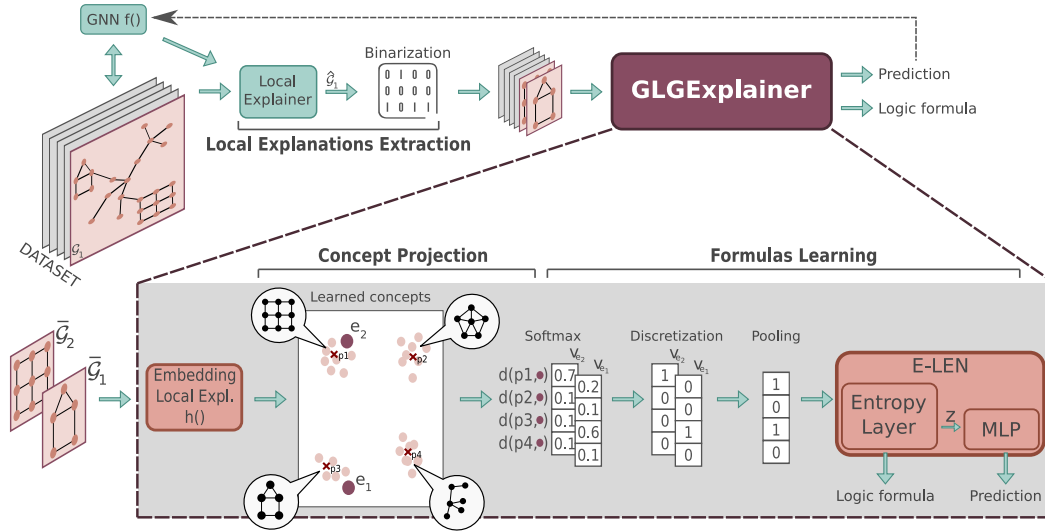


Figure 1: Illustration of the proposed method for a task of binary classification. Each step is described in detail in Section 2

51 Our proposed Global Explainer, named GLGExplainer (Global Logic-based GNN Explainer), is
 52 summarized in Figure 1. In the following we will describe each step in greater detail.

53 **Local Explanations Extraction:** The first step of our pipeline consists in extracting local explanations.
 54 Let $\text{LEXP}(f, \mathcal{G}) = \hat{\mathcal{G}}$ be the weighted graph obtained by applying the local explainer LEXP to
 55 generate a local explanation for the prediction of the GNN f over the input graph \mathcal{G} . In principle,
 56 every Local Explainer whose output can be mapped to a subgraph of the input sample is compatible
 57 with our pipeline [1–6]. Nonetheless, in this work, we relied on PGExplainer [2] since it allows
 58 the extraction of arbitrary disconnected motifs as explanations and it gave excellent results in our
 59 experiments. By binarizing the output of the local explainer $\hat{\mathcal{G}}$ with threshold $\theta \in \mathbb{R}$ we achieve a
 60 set of connected components $\bar{\mathcal{G}}_i$ such that $\bigcup_i \bar{\mathcal{G}}_i \subseteq \hat{\mathcal{G}}$. For convenience, we will henceforth refer to
 61 each of these $\bar{\mathcal{G}}_i$ as local explanation. Given that we want to emulate the behaviour of f on correctly
 62 predicted samples, we will discard every input graph \mathcal{G} belonging to wrongly predicted samples.
 63 The result of this extraction thus consists in a list D of local explanations. More details about the
 64 binarization are available in the Appendix.

65 **Embedding Local Explanations:** The following step consists in learning an embedding for
 66 each local explanation that allows to cluster together functionally similar local explanations. This
 67 can be achieved with a standard GNN h which maps any graph $\bar{\mathcal{G}}$ into a fixed-sized embedding
 68 $h(\bar{\mathcal{G}}) \in \mathbb{R}^d$. Since each local explanation $\bar{\mathcal{G}}$ is a subgraph of an input graph \mathcal{G} , in our experiments
 69 we used the original node features of the dataset. The outcome of this aggregation consists in a set
 70 $E = \{h(\bar{\mathcal{G}}), \forall \bar{\mathcal{G}} \in D\}$ of graph embeddings.

71 **Concept Projection:** Inspired by previous works on prototype learning [20, 21], we project each
 72 graph embedding $e \in E$ into a set P of $m \in \mathbb{N}$ prototypes $\{p_i \in \mathbb{R}^d | i = 1, \dots, m\}$ via a
 73 distance function $d(p_i, e) = \text{softmax} \left(\log \left(\frac{\|e - p_1\|^2 + 1}{\|e - p_1\|^2 + \epsilon} \right), \dots, \log \left(\frac{\|e - p_m\|^2 + 1}{\|e - p_m\|^2 + \epsilon} \right) \right)_i$. Prototypes are
 74 initialized randomly from a uniform distribution and are learned along with the other parameters
 75 of the architecture. As training progresses, the prototypes will align as prototypical representations
 76 of every cluster of local explanations, which will represent the final groups of graphical concepts.
 77 The output of this projection is thus a set $V = \{v_e, \forall e \in E\}$ where $v_e = [d(p_1, e), \dots, d(p_m, e)]$ is a
 78 vector containing the normalized probabilities of local explanation i belonging to the m concepts,
 79 and will be henceforth referred to as *concept vector*.

80 **Formulas Learning:** The final step consists of an E-LEN, i.e., a Logic Explainable Network [22]
 81 implemented with an Entropy Layer as first layer [23]. An E-LEN learns to map a concept activation
 82 vector to a class while encouraging a sparse use of concepts that allows to reliably extract Boolean
 83 formulas emulating the network behaviour. We train an E-LEN to emulate the behaviour of the
 84 GNN f feeding it with the graphical concepts extracted from the local explanations. Given a set of
 85 local explanations $\bar{\mathcal{G}}_a \dots \bar{\mathcal{G}}_{n_i}$ for an input graph \mathcal{G}_i and a corresponding set of the concept vectors
 86 $v_a \dots v_{n_i}$, we aggregate the concept vectors via a pooling operator and feed the resulting aggregated
 87 concept vector to the E-LEN, providing $f(\mathcal{G}_i)$ as supervision. In our experiments we used a max-
 88 pooling operator. Thus, the Entropy Layer learns a mapping from the pooled concept vector to (i)
 89 the embeddings z (as any linear layer) which will be used by the successive MLP for matching
 90 the predictions of f . (ii) a truth table T explaining how the network leveraged concepts to make
 91 predictions for the target class. Since the input pooled concept vector will constitute the premise
 92 in the truth table T , a desirable property to improve human readability is discreteness, which we
 93 achieved using the Straight-Through (ST) trick used for discrete Gumbel-Softmax Estimator [24]. In
 94 practice, we compute the forward pass discretizing each v_i via *argmax*, then, in the backward pass to
 95 favor the flow of informative gradient we use its continuous version.

96 **Supervision Losses:** Our proposed GLGExplainer is trained end-to-end with the following loss:
 97 $L = L_{surr} + \lambda_1 L_{R1} + \lambda_2 L_{R2}$, where L_{surr} corresponds to a Focal BCELoss [25] between the
 98 prediction of our E-LEN and the predictions to explain, while L_{R1} and L_{R2} are respectively aimed to
 99 push every prototype to be close to at least one local explanation and to push each local explanation
 100 to be close to at least one prototype [20]. The losses are defined as follows:

$$L_{surr} = -y(1 - p)^\gamma \log p - (1 - y)p^\gamma \log(1 - p) \quad (1)$$

101

$$L_{R1} = \frac{1}{m} \sum_{j=1}^m \min_{\bar{\mathcal{G}} \in D} \|p_j - h(\bar{\mathcal{G}})\|^2 \quad (2)$$

102

$$L_{R2} = \frac{1}{|D|} \sum_{\bar{\mathcal{G}} \in D} \min_{j \in [1, m]} \|p_j - h(\bar{\mathcal{G}})\|^2 \quad (3)$$

103 where p and γ represent respectively the probability for positive class prediction and the *focusing*
 104 parameter which controls how much to penalize hard examples.

105 3 Experiments

106 We tested our proposed approach on two datasets, namely:

107 **BAMultiShapes:** BAMultiShapes is a newly introduced extension of some popular synthetic bench-
 108 marks [1] aimed to assess the ability of a Global Explainer to deal with logical combinations of
 109 concepts. In particular, we created a dataset composed of Barabási-Albert (BA) graphs with attached
 110 in random positions the following network motifs: house, grid, wheel. Class 0 contains plain BA
 111 graphs and BA graphs enriched with a house, a grid, a wheel, or the three motifs together. Class 1
 112 contains BA graphs enriched with a house and a grid, a house and a wheel or a wheel and a grid.

113 **Mutagenicity:** The Mutagenicity dataset is a collection of molecule graphs where each graph is
 114 labelled as either having a mutagenic effect or not. Based on [26], the mutagenicity of a molecule is
 115 correlated with the presence of electron-attracting elements conjugated with nitro groups (e.g. *NO2*).

Table 1: Mean and standard deviation for Fidelity, Formula Accuracy and Concept Purity computed on the Test set over 5 runs with different random seeds. Since Concept Purity is computed for every cluster independently, here we report mean and standard deviation for the best run only.

Dataset	Fidelity	Formula Accuracy	Concept Purity
BAMultiShapes	0.99 ± 0.00	0.99 ± 0.00	0.85 ± 0.22
Mutagenicity	0.85 ± 0.01	0.85 ± 0.01	0.99 ± 0.01

116 For Mutagenicity we replicated the model accuracy and the local explanations presented in [2], while
 117 for BAMultiShapes we trained until convergence a 3-layers GCN. Details about the implementation
 118 and the pre-processing of local explanations, along with model accuracies, are in the Appendix.

119 In order to show the robustness of our proposed methodology, we have evaluated GLGExplainer on
 120 a number of metrics, namely: FIDELITY, FORMULA ACCURACY, and CONCEPT PURITY. A
 121 detailed description of those metrics can be found in the Appendix. Table 1 reports the results in terms
 122 of the three metrics, showing how GLGExplainer manages to provide reliable explanations under all
 123 these perspectives. Note that XGNN [10], the only available competitor for global explanations of
 124 GNN, cannot be evaluated according to these metrics. Figure 2 presents the final global explanations
 125 where we substituted each literal with its corresponding prototypical graphical concept, and report
 126 the explanations generated by XGNN for comparison. It’s easy to see that GLGExplainer produces
 127 highly interpretable explanations that match the ground-truth formula (for BAMultiShapes) and
 128 existing knowledge (for Mutagenesis) with remarkable accuracy. It is worth mentioning that the
 129 global explanations for Class 0 of BAMultiShapes do not comprise the case with all three motifs
 130 together. We observed that the reason resides in the GNN to explain failing at classifying every
 131 sample with such structure. So, GLGExplainer is effectively explaining the GNN f and not simply
 132 the dataset structure. Conversely, XGNN fails to generate interpretable explanations in most cases.
 133 Details about concepts compositions and formula extraction are available in the Appendix.

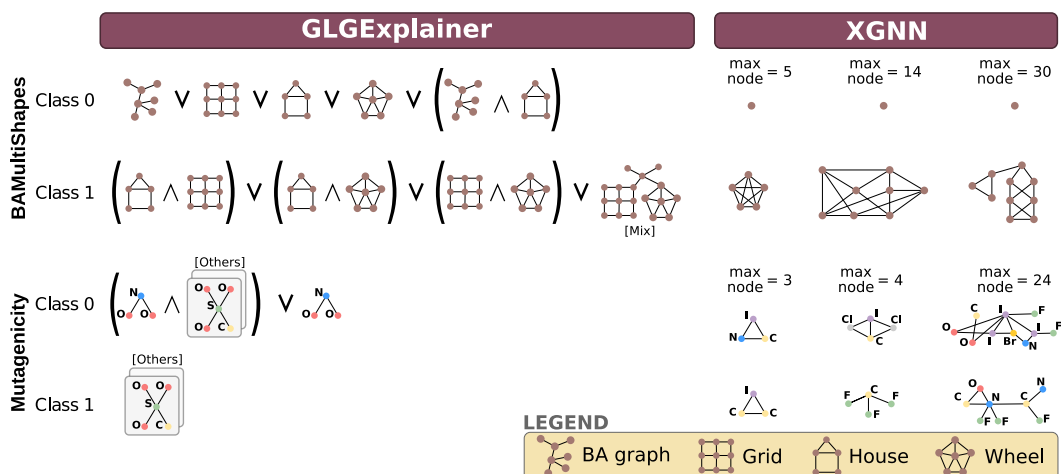


Figure 2: Global explanations of GLGExplainer (ours) and XGNN. The class probability predicted by XGNN for the generated explanations is around 1 for every explanation, except for Class 0 of BAMultiShapes where it was not able to generate a graph with confidence ≥ 0.5 .

134 4 Discussion & Conclusions

135 Given the results presented in the section above, it is worth noting that concept clusters emerge solely
 136 based on the supervision defined in Section 2, while no specific supervision was added to cluster local
 137 explanations based on their similarity. Further details about the clusters’ composition are available in
 138 the Appendix. Overall, the results confirm the ability of GLGExplainer in providing logic formulas,
 139 expressed over learned graphical concepts, which are accurately summarizing the global behaviour of
 140 the model, whereas the existing XGNN fails at providing concise and faithful explanations.

References

- 141
- 142 [1] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer:
143 Generating explanations for graph neural networks, 2019. URL [https://arxiv.org/abs/](https://arxiv.org/abs/1903.03894)
144 [1903.03894](https://arxiv.org/abs/1903.03894). 1, 2, 3
- 145 [2] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang
146 Zhang. Parameterized explainer for graph neural network, 2020. URL [https://arxiv.org/abs/](https://arxiv.org/abs/2011.04573)
147 [2011.04573](https://arxiv.org/abs/2011.04573). 2, 4, 6, 7
- 148 [3] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural
149 networks via subgraph explorations, 2021. URL <https://arxiv.org/abs/2102.05152>.
- 150 [4] Minh N. Vu and My T. Thai. Pgm-explainer: Probabilistic graphical model explanations for
151 graph neural networks, 2020. URL <https://arxiv.org/abs/2010.05788>.
- 152 [5] Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. Reinforcement learning
153 enhanced explainer for graph neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S.
154 Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*,
155 volume 34, pages 22523–22533. Curran Associates, Inc., 2021. URL [https://proceedings.](https://proceedings.neurips.cc/paper/2021/file/be26abe76fb5c8a4921cf9d3e865b454-Paper.pdf)
156 [neurips.cc/paper/2021/file/be26abe76fb5c8a4921cf9d3e865b454-Paper.pdf](https://proceedings.neurips.cc/paper/2021/file/be26abe76fb5c8a4921cf9d3e865b454-Paper.pdf).
- 157 [6] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoff-
158 mann. Explainability methods for graph convolutional neural networks. In *Proceedings of the*
159 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10772–10781, 2019.
160 2
- 161 [7] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. Graphlime: Local
162 interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge*
163 *and Data Engineering*, 2022.
- 164 [8] Wanyu Lin, Hao Lan, and Baochun Li. Generative causal explanations for graph neural networks.
165 In *International Conference on Machine Learning*, pages 6666–6679. PMLR, 2021.
- 166 [9] Wanyu Lin, Hao Lan, Hao Wang, and Baochun Li. Orphicx: A causality-inspired latent variable
167 model for interpreting graph neural networks. In *Proceedings of the IEEE/CVF Conference on*
168 *Computer Vision and Pattern Recognition*, pages 13729–13738, 2022. 1
- 169 [10] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. XGNN: Towards model-level explanations
170 of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference*
171 *on Knowledge Discovery & Data Mining*. ACM, aug 2020. doi: 10.1145/3394486.3403085.
172 URL <https://doi.org/10.1145/3394486.3403085>. 1, 4
- 173 [11] Mattia Setzu, Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca
174 Giannotti. GLocalX - from local to global explanations of black box AI models. *Artificial*
175 *Intelligence*, 294:103457, may 2021. doi: 10.1016/j.artint.2021.103457. URL [https://doi.](https://doi.org/10.1016/j.artint.2021.103457)
176 [org/10.1016/j.artint.2021.103457](https://doi.org/10.1016/j.artint.2021.103457). 1
- 177 [12] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R. Lyu, and Yu-Wing
178 Tai. Towards global explanations of convolutional neural networks with concept attribution.
179 In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages
180 8649–8658, 2020. doi: 10.1109/CVPR42600.2020.00868. 1
- 181 [13] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas,
182 and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept
183 activation vectors (TCAV). In *International Conference on Machine Learning (ICML)*,
184 volume 80 of *Proceedings of Machine Learning Research*, pages 2673–2682. PMLR, 2018. 1
- 185 [14] Amirata Ghorbani, James Wexler, James Y. Zou, and Been Kim. Towards automatic concept-
186 based explanations. In *Neural Information Processing Systems (NeurIPS)*, pages 9273–9282,
187 2019.
- 188 [15] Chih-Kuan Yeh, Been Kim, Sercan Ömer Arik, Chun-Liang Li, Tomas Pfister, and Pradeep
189 Ravikumar. On completeness-aware concept-based explanations in deep neural networks. In
190 *Neural Information Processing Systems (NeurIPS)*, 2020. 1
- 191 [16] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim,
192 and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*
193 *(ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR,
194 2020. 1

- 195 [17] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su.
196 This looks like that: Deep learning for interpretable image recognition. *Advances in Neural*
197 *Information Processing Systems*, 32:8930–8941, 2019. 1
- 198 [18] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. Protgnn: Towards
199 self-explaining graph neural networks. In *Proceedings of the AAAI Conference on Artificial*
200 *Intelligence*, volume 36, pages 9127–9135, 2022. 2
- 201 [19] Dobrik Georgiev, Pietro Barbiero, Dmitry Kazhdan, Petar Veličković, and Pietro Liò. Algorithmic
202 concept-based explainable reasoning. In *Proceedings of the AAAI Conference on Artificial*
203 *Intelligence*, volume 36, pages 6685–6693, 2022. 2
- 204 [20] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning
205 through prototypes: A neural network that explains its predictions, 2017. URL [https://](https://arxiv.org/abs/1710.04806)
206 arxiv.org/abs/1710.04806. 3
- 207 [21] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This
208 looks like that: Deep learning for interpretable image recognition. In H. Wallach, H. Larochelle,
209 A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information*
210 *Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL [https://proceedings.](https://proceedings.neurips.cc/paper/2019/file/adf7ee2dcf142b0e11888e72b43fcb75-Paper.pdf)
211 [neurips.cc/paper/2019/file/adf7ee2dcf142b0e11888e72b43fcb75-Paper.pdf](https://proceedings.neurips.cc/paper/2019/file/adf7ee2dcf142b0e11888e72b43fcb75-Paper.pdf). 3
- 212 [22] Gabriele Ciravegna, Pietro Barbiero, Francesco Giannini, Marco Gori, Pietro Lió, Marco
213 Maggini, and Stefano Melacci. Logic explained networks. *arXiv preprint arXiv:2108.05149*,
214 2021. 3
- 215 [23] Pietro Barbiero, Gabriele Ciravegna, Francesco Giannini, Pietro Lió, Marco Gori, and Stefano
216 Melacci. Entropy-based logic explanations of neural networks, 2021. URL [https://arxiv.](https://arxiv.org/abs/2106.06804)
217 [org/abs/2106.06804](https://arxiv.org/abs/2106.06804). 3, 13
- 218 [24] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax,
219 2016. URL <https://arxiv.org/abs/1611.01144>. 3
- 220 [25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense
221 object detection, 2017. URL <https://arxiv.org/abs/1708.02002>. 3
- 222 [26] Asim Kumar Debnath, Rosa L. Lopez de Compadre, Gargi Debnath, Alan J. Shusterman, and
223 Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic
224 nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal*
225 *of Medicinal Chemistry*, 34(2):786–797, 1991. doi: 10.1021/jm00106a046. URL [https:](https://doi.org/10.1021/jm00106a046)
226 [/doi.org/10.1021/jm00106a046](https://doi.org/10.1021/jm00106a046). 3
- 227 [27] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional
228 networks. *arXiv preprint arXiv:1609.02907*, 2016. 6, 13
- 229 [28] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural
230 networks? *arXiv preprint arXiv:1810.00826*, 2018. 7, 13
- 231 [29] Will Hamilton, Zhitaoying, and Jure Leskovec. Inductive representation learning on large
232 graphs. *Advances in neural information processing systems*, 30, 2017. 13
- 233 [30] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua
234 Bengio. Graph attention networks. *stat*, 1050:20, 2017. 13

235 A Appendix

236 A.1 Training the GNN f

237 For both BAMultiShapes and Mutagenicity we relied on the codebase provided by [2] for training
238 the GNN f to explain and to train the Local Explainer. For BAMultiShapes we trained a 3-layers
239 GCN [27] (20-20-20 hidden units) with mean graph pooling for the final prediction, whereas for
240 Mutagenicity we reproduced the results of [2]. A summary of model’s performance is available in
241 Table 2. Despite the high accuracy over BAMultiShapes, after a closer look we observed that the
242 network did not actually learn the *All* concept, i.e., the three motifs together. Such detailed view is
243 available in Table 3. This explains why the global explanations in Figure 2 Class 0 do not present
244 such concept.

Table 2: GNN accuracies for BAMultiShapes and Mutagenicity. The results for Mutagenicity are in line with the one reported in [2].

Split	BAMultiShapes	Mutagenicity
Train	0.94	0.87
Val	0.94	0.86
Test	0.99	0.86

Table 3: Accuracy of the model on the train set of BAMultiShapes with respect to every combination of motifs to be added to the Barabási-Albert base graph. H , G , W stand respectively for House, Grid, and Wheel.

	Class 0					Class 1		
Motifs	\emptyset	H	G	W	All	$H + G$	$H + W$	$G + W$
Accuracy (%)	1.0	1.0	0.85	1.0	0.0	1.0	0.98	1.0

245 A.2 Local Explanations Processing

246 As detailed in [2], the output of PGExplainer consists in a weighted edge mask $w_{ij} \in \mathcal{V} \times \mathcal{V}$ where
 247 each w_{ij} is the likelihood of the edge being an important edge. For Mutagenicity, we stuck to the
 248 original implementation which was correctly able to reproduce the results presented in the paper [2].
 249 The only difference resides in the procedure for cutting the explanation, which is needed to remove
 250 from the final local explanation the edges which were assigned low scores. The authors in [2] limited
 251 their analysis to graphs that were containing the ground truth motifs, and proposed to just keep the
 252 top-k edges. We, instead, selected the numeric threshold $\theta \in \mathbb{R}$ which maximises the F1 score of the
 253 explainer over all graphs. Afterwards, such threshold will be used to cut out the irrelevant edges, by
 254 applying the indicator function $\mathbf{1}_{w_{ij} \geq \theta}$ to the edge mask. The resulting edge mask is thus the binary
 255 adjacency matrix of the final explanation. For BAMultiShapes, however, we adopted a dynamic
 256 algorithm to select θ that does not require any prior knowledge about the ground truth motifs. This
 257 algorithm resembles the elbow-method, i.e., for each local explanation choose as θ the first value that
 258 is different enough from the previous ordered values. Figure 3 shows some examples for each dataset
 259 along with their local explanations in bold.

260 A.3 The GLGExplainer

261 The reference implementation of our Local Explanation Embedder h is constituted by a 2-layers
 262 GIN [28] network with 20 hidden units, followed by a non-linear combination of max, mean, and
 263 sum graph pooling. We chose a number m of 6 and 2 prototypes for, respectively, BAMultiShapes
 264 and Mutagenicity, keeping the dimensionality d to 10. We trained using ADAM optimizer with
 265 early stopping and with a learning rate for h and the prototypes P of $1e^{-3}$, while for the E-LEN of
 266 $5e^{-4}$. The batch size is set to 128, while the auxiliary loss coefficients λ_1 and λ_2 are chosen via
 267 cross-validation and set respectively to 0.09 and 0.00099, while the focusing parameter γ is kept fixed
 268 at 2. The E-LEN is constituted by the input Entropy Layer ($Entr.Layer : \mathbb{R}^m \rightarrow \mathbb{R}^{10}$), a hidden
 269 layer ($HiddenLayer : \mathbb{R}^{10} \rightarrow \mathbb{R}^5$), and the output layer with LeakyReLU activation function.

270 In the rest of this section we provide an ablation study to demonstrate the effectiveness of the Focal
 271 loss, the Discretization trick, and the impact of the number of prototypes in use.

272 **Focal loss:** Figure 4 presents a comparison of the learning curve for BAMultiShapes showing that
 273 using Focal loss with a *focusing* parameter of 2 helps to achieve a faster convergence while not being
 274 detrimental for the overall performances.

275 **Number of prototypes:** An effective approach to select an appropriate value m for the number of
 276 prototypes in use is via cross-validation, and by selecting the smallest m which achieves a competitive
 277 fidelity. In Figure 5 we show how different values of m impact the Fidelity and the Formula Accuracy.

278

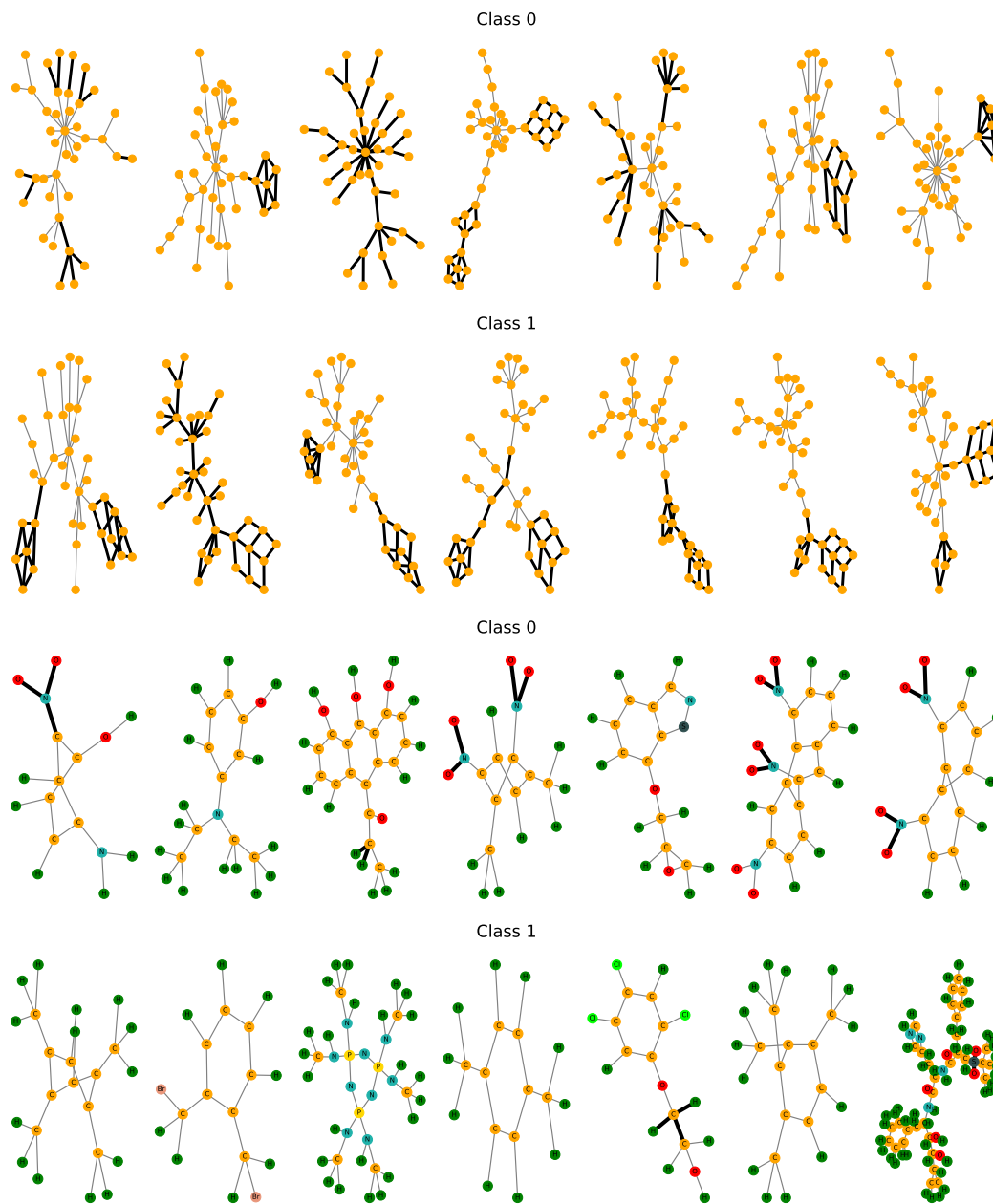


Figure 3: Random examples of input graphs along with their explanations in bold as extracted by PGExplainer, for respectively BAMultiShapes and Mutagenicity.

279 **Discretization trick:** The Discretization trick was introduced in Section 2 to enforce a discrete
 280 prototype assignment, something essential for an unambiguous definition of the concepts on which
 281 the formulas are based on. In Figure 6 we show for BAMultiShapes that this trick is also effective in
 282 improving the overall performance of GLGExplainer, since it forces the hidden layers of the E-LEN
 283 to just exploit the information relative to the closest prototype, while not relying on other positional
 284 information. Thus, the E-LEN’s predictions are much more aligned with the discrete formulas being
 285 extracted. In the Figure we further compare against a plain model without Discretization and against
 286 the addition to the overall loss of an entropy loss over the concept vector (Concept Entropy loss)
 287 with different scaling parameters $\lambda_3 \in \{0.01, 0.1\}$. This Concept Entropy loss (CE loss) pushes the
 288 pre-pooling concept vector to have low entropy, thus effectively pushing every local explanation to
 289 be assigned with confidence to just one prototype.

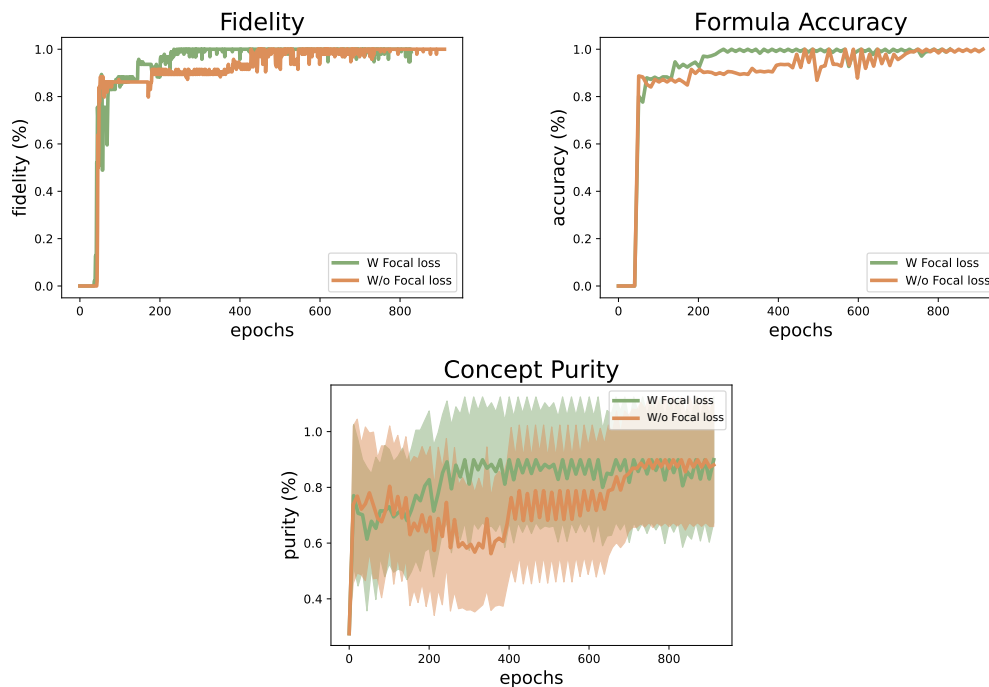


Figure 4: Learning curves for BAMultiShapes with and without Focal loss. Results show that the Focal loss, with a focusing parameter set to 2, helps to achieve faster convergence while not reducing the final performances.

Table 4: Raw formulas as extracted by the Entropy Layer. Each formula was rewritten following the Closed-World Assumption for convenience.

Dataset	Raw Formulas
BAMultiShapes	Class ₀ $\iff P_0 \vee P_5 \vee P_1 \vee P_4 \vee P_2 \vee (P_4 \wedge P_2)$
	Class ₁ $\iff P_3 \vee (P_5 \wedge P_2) \vee (P_5 \wedge P_1) \vee (P_2 \wedge P_1)$
Mutagenicity	Class ₀ $\iff P_1 \vee (P_0 \wedge P_1)$
	Class ₁ $\iff P_0$

290 A.4 Cluster Composition & Formulas Renaming

291 To effectively explore the content of each local explanations cluster, we plot in Figure 7 some random
 292 elements for each dataset. In most cases, the clusters contain atomic motifs (House, Grid, NO2,
 293 etc..) while in others the embedder h clustered together heterogeneous motifs. This is particularly
 294 evident for the cluster relative to the prototype p_3 of BAMultiShapes in which every local explanation
 295 comprising two atomic motifs are aggregated. The reason for this behaviour is that we are aggregating
 296 local explanation solely based on the ability of the E-LEN to emulate the predictions of f . Thus,
 297 since the simultaneous presence of two motifs appears only in Class 1, one single cluster aggregating
 298 all these *mixed* local explanations is enough for maximizing the performances. This is also the reason
 299 for the high variability in Concept Purity reported in Table 1, since it is computed considering the
 300 Purity in terms of labelled atomic motifs. For completeness, we additionally report in Figure 8 a 2D
 301 PCA-reduced view of the embedding space, annotated with the prototypes position.

302 Given that the default implementation of the Entropy layer returns formulas expressed in terms of the
 303 single concepts in input, Figure 7 is also useful to rename each literal into its corresponding graphical
 304 concept. Table 4 shows an example of such raw formulas, while Figure 2 presents the final formulas
 305 after replacing each raw name with the corresponding graphical concept.

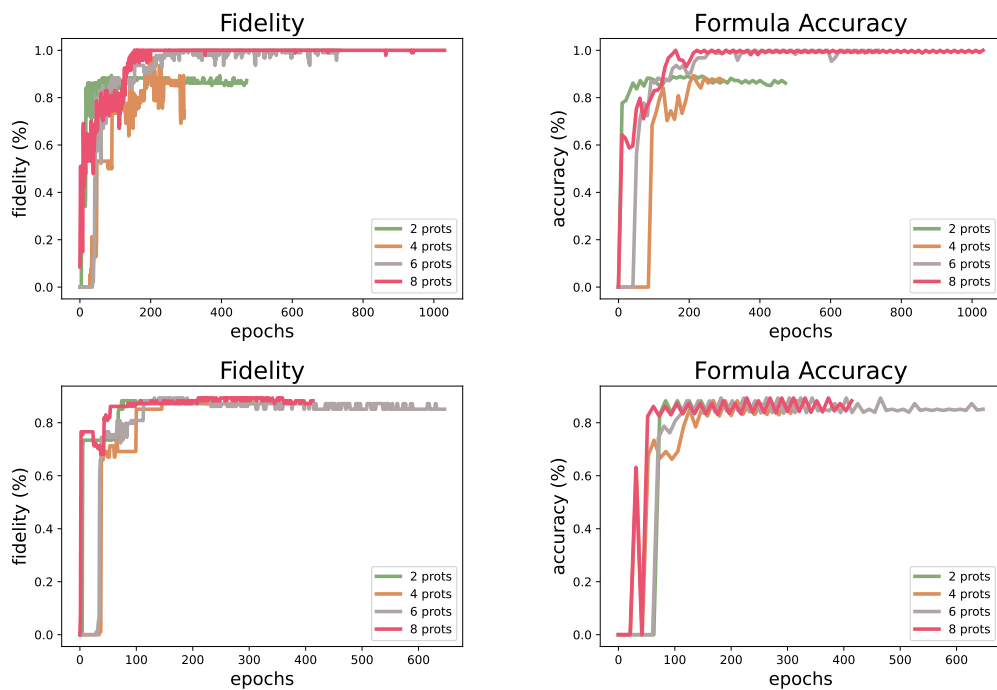


Figure 5: Ablation study on the number of prototypes to use. The first row is referred to BAMulti-Shapes, while the second to Mutagenicity.

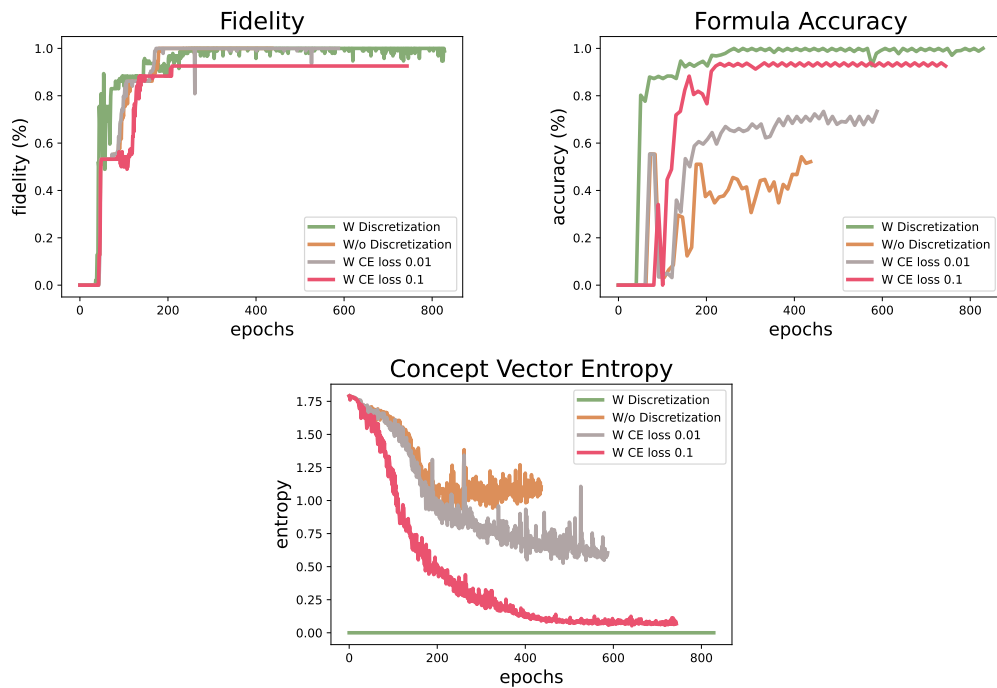


Figure 6: Ablation study for the impact of the Discretization trick, discussed in Section 2. We compared the performances with and without it, and against the addition of a Concept Entropy loss (CE loss) with different scaling parameters λ_3 .

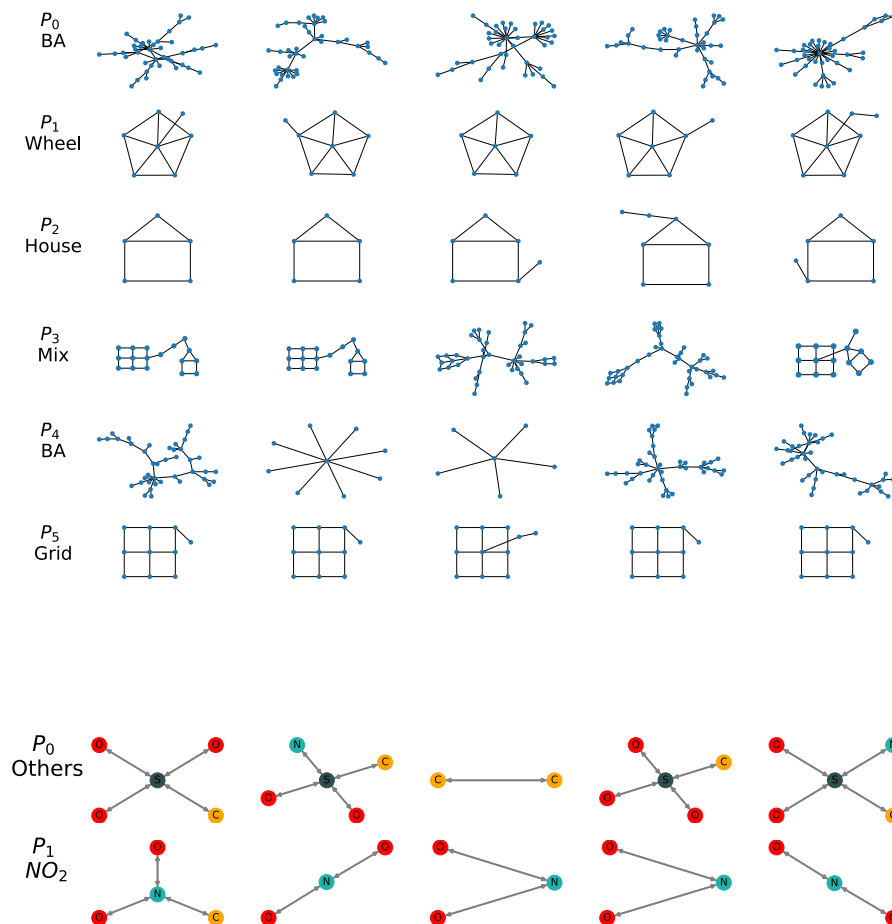


Figure 7: Random representative elements for each prototype in BAMultiShapes and Mutagenicity.

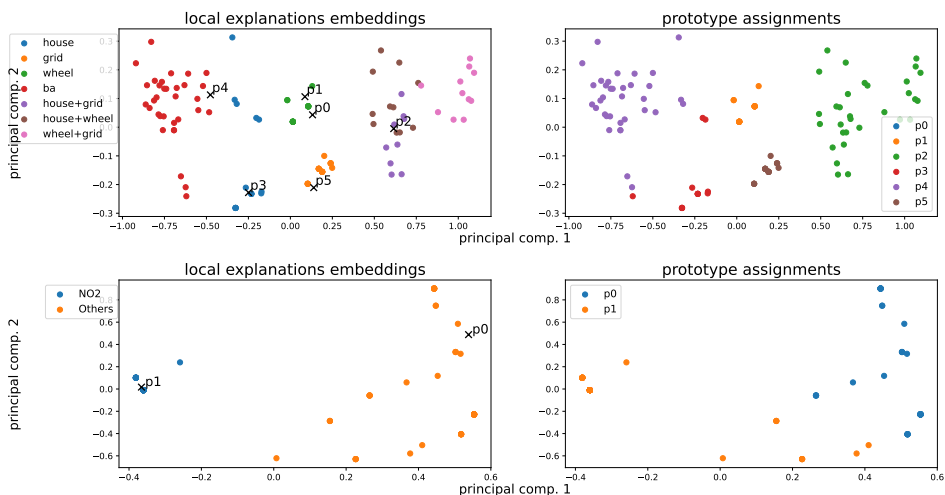


Figure 8: 2D view of the embedding space annotated with prototypes positions. The first line is referred to BAMultiShapes, while the second to Mutagenicity.

306 A.5 Benefits and Limitations

307 As previously discussed, the proposed GLGExplainer is inherently faithful to the data domain since
 308 it processes local explanations provided by a Local Explainer. However, the quality of those local
 309 explanations, in terms of representativeness and discriminability with respect to the task-specific class,
 310 has a direct effect on the Fidelity. If the generated concept vector does not exhibit any class-specific
 311 pattern, then the E-LEN will not be able to emulate the predictions of the model to explain. Despite
 312 being a potential limitation of GLGExplainer, this can actually open to the possibility of using the
 313 Formula Accuracy as a proxy of local explanations quality, which is notoriously difficult to assess.
 314 We leave this investigation to future work. Despite tailoring our discussion on graph classification,
 315 our approach can be readily extended to any kind of classification task on graphs, provided that a
 316 suitable Local Explainer is available.

317 A.6 Evaluation Metrics

318 Here we will describe in more detail the metrics briefly introduced in Section 3:

- 319 • Fidelity measures the accuracy between the prediction of the E-LEN and the one of the GNN
 320 to explain. It is computed as the accuracy between the class predictions of the E-LEN and the
 321 GNN f .
- 322 • Formula Accuracy represents how well the learned formulas can correctly predict the class
 323 labels. To compute this metric, we treat the final formulas as a classifier that given an input
 324 concept vector predicts the class corresponding to the clause evaluated to true. In the cases in
 325 which either no clause or more clauses of different classes are evaluated to be true, the sample is
 326 always considered as wrongly predicted.
- 327 • Concept Purity is computed for every cluster independently and measures how good the embed-
 328 ding is at clustering the local explanations. Specifically, it requires each local explanation to be
 329 annotated with a label, which in our cases corresponds to the typology of the motif represented
 330 by the local explanation. Then, the computation of the metric can be summarized by:

$$331 \text{ConceptPurity}(C_i) = \frac{\text{count_most_frequent_label}(C_i)}{|C_i|} \quad (4)$$

332 where C_i corresponds to the cluster having p_i as prototype (i.e., the cluster containing every local
 333 explanation associated to prototype p_i by the distance function $d(., .)$ described in Section 2).
 334 $\text{count_most_frequent_label}(C_i)$ instead returns the number of local explanations annotated
 335 with the most present label in cluster C_i . The Concept Purity results reported in Table 1 are
 computed by taking the mean and the standard deviation across the m clusters.

336 A.7 Additional Experiments

337 In this section, we report further experimental results in addition to those presented in Section 3.

338 A.7.1 Multi-Class Dataset

339 To challenge GLGExplainer also in a multi-class setting, we extended the previously introduced
 340 BAMultiShapes dataset with an additional class, constituted by the usual random BA base graph
 341 with attached house-like motifs and a cycle of length 6. We will henceforth refer to this dataset as
 342 *BAMultiShapesMC*. As for BAMultiShapes, node features are represented by a fixed vector with
 343 values 0.1. Similarly as done for BAMultiShapes, we trained an extension of the 3-layers GCN
 344 used before, using both a max and mean aggregator for graph pooling. The resulting accuracies are
 345 0.95% and 0.98% for respectively the train and test set. We kept the same setting for extracting and
 346 processing local explanations, with the only difference that we used 128 hidden units for PGExplainer,
 347 instead of the original 64, to favour the extraction of good local explanations which otherwise were
 348 of very low quality. For GLGExplainer, we kept almost all of the hyper-parameters presented in
 349 Appendix A.3. We found, however, to be very beneficial for the final embedding to use $m = 8$. Note
 350 that this is a reasonable modification, given that we added completely new motifs and the overall
 351 dataset composition is more complex.

352 In Table 5 we report the results over 5 different random seeds, while in Figure 10 we provide an
 353 illustration of the PCA 2D-reduced embedding for the best run along with the relative formulas

Table 5: Mean and standard deviation for Fidelity, Formula Accuracy and Concept Purity computed on the Test set over 5 runs with different random seeds. Since the Concept Purity is computed for every cluster independently, here we report mean and standard deviation for the best run only.

Dataset	Fidelity	Formula Accuracy	Concept Purity
BAMultiShapesMC	0.97 ± 0.02	0.97 ± 0.01	0.82 ± 0.23

Table 6: Raw formulas as extracted by the Entropy Layer along with their test Fidelity.

Dataset	Raw Formulas	Fidelity
BAMultiShapesMC	Class ₀ \iff $P_0 \vee P_2 \vee P_3 \vee P_5 \vee P_7$ $P_1 \vee P_6 \vee (P_0 \wedge P_2) \vee (P_0 \wedge P_5) \vee$ $(P_0 \wedge P_7) \vee (P_5 \wedge P_7) \vee (P_6 \wedge P_7) \vee$	0.98
	Class ₁ \iff $(P_5 \wedge P_6) \vee (P_0 \wedge P_4) \vee (P_0 \wedge P_6) \vee$ $(P_2 \wedge P_6) \vee (P_2 \wedge P_7) \vee (P_3 \wedge P_5 \wedge P_6)$	
	Class ₂ \iff $P_4 \vee (P_3 \wedge P_5) \vee (P_2 \wedge P_3) \vee$ $(P_3 \wedge P_4) \vee (P_2 \wedge P_3 \wedge P_5)$	

in Table 6. As it is possible to inspect from the output of the E-LEN, the resulting extracted raw formulas are still well representing the underlying ground truth modelling correctly the presence of the new class, despite containing some additional noise. For example, given the overlapping between some prototype assignments (like the cluster of P_5 that, even if representing the vast majority of BA base graphs, it contains some spurious houses, or similarly for P_3 that despite containing every local explanation representing a circle, contains a few BA graphs) the E-LEN learned some spurious clauses which are not correctly modelling the underlying ground truth. Those cases represent however the real minority of cases, being the overall formulas and Fidelity well aligned with the results obtained for BAMultiShapes, where the quality of local explanations allowed more distinctive clusters. Note indeed that as described in [23] and implemented in the official codebase¹, it is possible to rank the clauses in the truth table T created by the E-LEN by their support (for how many samples they hold), allowing to select only the top-ranked clauses either by evaluating on a validation set, or by specifying a minimum support. Nonetheless, we did not apply such filter in order to stick to the experimental setting previously defined. In the same vein, despite the possibility of arbitrarily augmenting the local explanations’ node features with any hand-crafted feature in order to better separate the clusters, we kept only the original datasets’ node features.

As mentioned in Appendix A.5, GLGExplainer can be used to get insights into the Local Explainer in use. To this end, it is possible to understand the reason behind, for example, the clause P_1 for Class₁ in Table 6 by the fact that each of the ~ 20 samples in this cluster corresponds to the external border of the grid-like motif, which are extracted for the vast majority in Class₁. This means that, in this specific case, PGExplainer had a bias in extracting solely the border of the grid motif in Class₁, ignoring the other motif present in the same sample (recall that every sample of Class₁ has two motifs). An example of such motif along with the overall sample-wise local explanations are reported in Figure 9.

378 A.7.2 Impact of the GNN Architecture on the Embedder h

379 In our previous experiments we implemented the Local Explanations Embedder h as a 2-layers GIN
 380 network. However, any compatible GNN architecture can be used instead. In Figure 11 we present
 381 an ablation study testing different GNN architectures, and comparing their respective performances
 382 in terms of Fidelity and Concept Purity. For every architecture under analysis we kept the number
 383 of layers equal to 2. Specifically, we tested GCN [27], GIN [28], SAGE [29], and GAT [30]. For
 384 BAMultiShapes, in which are present different and heterogeneous motifs, architectures with a sum
 385 local aggregator like SAGE and GIN seem to be preferable over GCN, where presumably the node-
 386 degree normalization in its propagation rule limits the richness of the learned representation. Given
 387 the absence of informative node features, GAT is not able to properly learn a useful embedding. For

¹<https://pypi.org/project/torch-explain/>

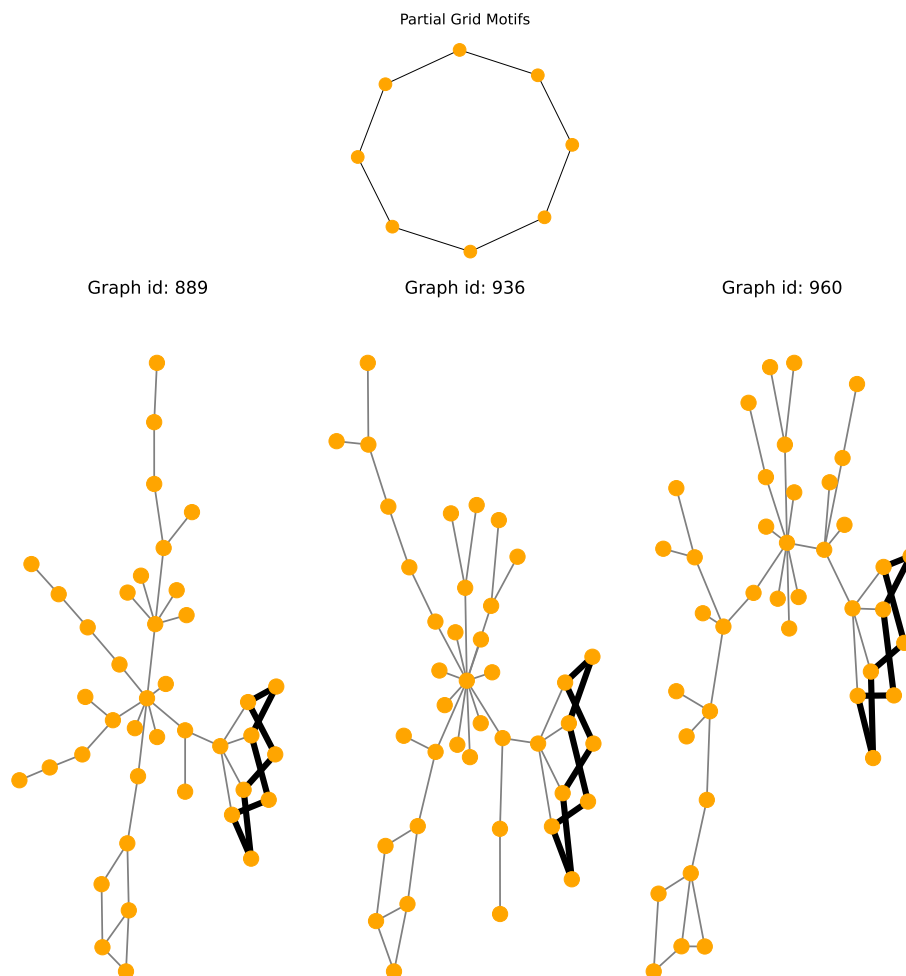


Figure 9: Illustration of the grid’s external border only local explanation extracted for some samples of Class_1 , along with three examples of instance-level local explanations with the selected explanation in bold. The expected local explanation is comprised by both the entire grid and the house.

388 Mutagenicity, instead, given the presence of informative node features, and given the absence of rich
 389 topological motifs, GAT and GCN perform comparably to GIN and SAGE.

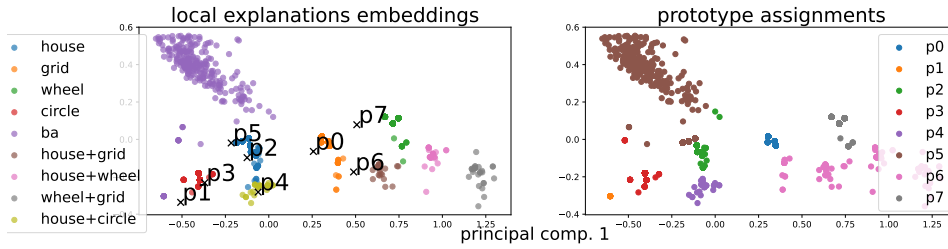


Figure 10: 2D view of the embedding space annotated with prototypes positions for the train split of BAMultiShapesMC.

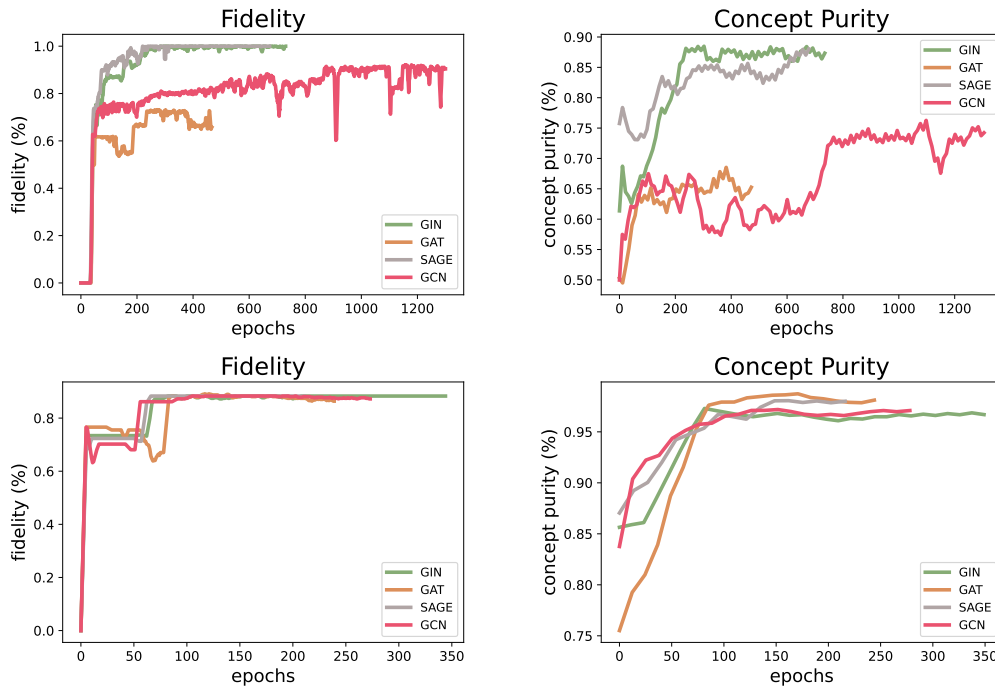


Figure 11: Ablation study on the architecture in use for the Local Explanations Embedder h . The first line is referred to BAMultiShapes, while the last to Mutagenicity