Explaining the Explainers in Graph Neural Networks: a Comparative Study

ANTONIO LONGA, Fondazione Bruno Kessler, Trento, Italy and DISI, University of Trento, Trento, Italy
STEVE AZZOLIN, DISI, University of Trento, Trento, Italy
GABRIELE SANTIN, Fondazione Bruno Kessler, Trento, Italy
GIULIA CENCETTI, Fondazione Bruno Kessler, Trento, Italy
PIETRO LIO, Cambridge University, Cambridge, United Kingdom of Great Britain and Northern Ireland
BRUNO LEPRI, Fondazione Bruno Kessler, Trento, Italy
ANDREA PASSERINI, DISI, University of Trento, Trento, Italy

Following a fast initial breakthrough in graph based learning, Graph Neural Networks (GNNs) have reached a widespread application in many science and engineering fields, prompting the need for methods to understand their decision process. GNN explainers have started to emerge in recent years, with a multitude of methods both novel or adapted from other domains. To sort out this plethora of alternative approaches, several studies have benchmarked the performance of different explainers in terms of various explainability metrics. However, these earlier works make no attempts at providing insights into why different GNN architectures are more or less explainable, or which explainer should be preferred in a given setting. In this survey we fill these gaps by devising a systematic experimental study, which tests twelve explainers on eight representative message-passing architectures trained on six carefully designed graph and node classification datasets. With our results we provide key insights on the choice and applicability of GNN explainers, we isolate key components that make them usable and successful and provide recommendations on how to avoid common interpretation pitfalls. We conclude by highlighting open questions and directions of possible future research.

CCS Concepts: • **Computing methodologies** → **Neural networks**.

Additional Key Words and Phrases: Explainability, Graph Neural Networks

1 Introduction

Graph Neural Networks (GNNs) have emerged as the de-facto standard for graph-based learning tasks. Regardless of their apparent simplicity, that allows most GNN architectures to be expressed as variants of Message Passing [31], i.e., exchanging messages between nodes, GNNs have proved extremely effective in preserving the natural symmetries present in many real-world physical systems [10, 25, 43, 75, 106]. The versatility of GNNs allowed them to be also applied to emulate classical algorithms [14], addressing tasks like bipartite matching [30], graph coloring [56] or the Traveling Salesperson Problem [74], and approximate symbolic reasoning tasks like propositional satisfiability [89, 90, 107] and probabilistic logic reasoning [127]. Despite recent works trying to adapt the Transformer architecture, made popular by a wide success first in language [76, 82, 102] and then

Authors' Contact Information: Antonio Longa, Fondazione Bruno Kessler, Trento, Italy and DISI, University of Trento, Trento, Italy; e-mail: longaantonio@gmail.com; Steve Azzolin, DISI, University of Trento, Trento, Italy; e-mail: steve.azzolin@studenti.unitn.it; Gabriele Santin, Fondazione Bruno Kessler, Trento, Italy; e-mail: gsantin@fbk.eu; Giulia Cencetti, Fondazione Bruno Kessler, Trento, Italy; e-mail: gcencetti@fbk.eu; Pietro Lio, Cambridge University, Cambridge, Cambridgeshire, United Kingdom of Great Britain and Northern Ireland; e-mail: pl219@cam.ac.uk; Bruno Lepri, Fondazione Bruno Kessler, Trento, Trentino-Alto Adige, Italy; e-mail: lepri@fbk.eu; Andrea Passerini, DISI, University of Trento, Trento, Italy; e-mail: andrea.passerini@unitn.it.

ACM 1557-7341/2024/9-ART
https://doi.org/10.1145/3696444

in vision applications [3, 61, 78], to the graph domain [22, 48, 79, 123, 129], the natural inductive bias of GNNs remains at the basis of the current success of GNNs. A major drawback of GNNs with respect to alternative graph processing approaches [34, 39, 72, 95] is the opacity of their predictive mechanism, which they share with most deep-learning based architectures. This severely limits the applicability of these technologies to safety-critical scenarios.

The need to provide insights into the decision process of the network, and the need to provide explanations for automatic decisions affecting human's life [18, 47], have stimulated research in techniques for shading light into the black box nature of deep architectures [35, 73, 81, 91, 98, 100, 101, 112, 124]. The approaches have also been adapted to generate explanations for GNN models [8, 73, 98, 100]. However, networked data have peculiarities that pose specific challenges that explainers developed for tensor data struggle to address. The main challenge comes from the lack of a regular structure, as nodes have a variable number of edges, which requires ad-hoc strategies to be properly addressed. Indeed, several approaches have been recently developed that are specifically tailored to explain GNN architectures, and Section 3 will summarize the main contributions.

It is often the case, however, that each work proposes a new set of benchmarks or metrics, making the comparison across works complicated. We thereby stress the need for a comprehensive evaluation that can fairly benchmark the explainers under a unified lens. One of the first attempts to provide such a comparative analysis is the up mentioned work by Yuan et al. [120], where a taxonomy of the available explainers was proposed. In addition to this, the authors reported a detailed overview of the most common datasets used to benchmark explainers, along with the adopted evaluation metrics.

However, despite the wide coverage of explainers, datasets, and evaluation metrics, only a single GNN architecture, namely a simple Graph Convolutional Network [49], was evaluated, so that nothing can be said about the impact of different architectures in the resulting explanations. A similar limitation affects the works of Zhao et al. [131] and Agarwal et al. [1, 2] that, despite presenting interesting insights in terms of consistency of explainers, desired properties of explanation metrics and even introducing a generator for synthetic graph benchmarks, focus their analysis to a single GNN architecture. Li et al. [55] conducted the first empirical study comparing different GNN architectures. However, their study is limited to node classification and the three explainers under analysis [64, 86, 115] are not well representative of the diversity of explanation strategies that have been proposed, as summarized in the aforementioned taxonomy [120]. The most comprehensive study to date is the recent work by Rathee at al. [80], that evaluated four GNN architectures over nine explainers for both node and graph classification. However, the main goal of this study is proposing a benchmarking suite to quantitatively evaluate explainers, with no attempts at providing insights into why different GNN architectures behave differently in terms of explainability, or which explainer should be preferred in a given setting.

In spite of the aforementioned recent studies benchmarking explainability methods for GNNs, no investigation has been done in characterizing the typical explanation patterns associated to the topological concepts learned by the network and how different architectures affect the explanation. In our work, we address these issues by answering the following research questions:

- **RQ1:** How does the architecture affect the explanations?
- **RQ2:** How do explainers affect the explanations?
- **RQ3:** How do different types of problems affect the explanations?

Overall, our work aims to go beyond a merely quantitative evaluation of the performance of explainer-GNN pairs and to make a significant step towards *explaining explainability*. We run an unprecedented number of experiments involving eight GNN architectures, twelve instance-based explainers, and six datasets (divided into node and graph classification) and enrich the quantitative results we obtain by providing a deep understanding of the reasons behind the observed behaviors, together with a set of recommendations on how to select and best use the most appropriate explainer for the task under investigation while avoiding common pitfalls, as well as a

Symbol	Description
$G := (V, E, \boldsymbol{X})$	A graph.
$V := \{1, \dots, n_V\}$	The set of $n_V \in \mathbb{N}$ nodes of a graph.
$E \subset V \times V$	The set of $n_E \in \mathbb{N}$ edges of a graph.
$\boldsymbol{X} \in \mathbb{R}^{n_V \times d}, \boldsymbol{X}_i \in \mathbb{R}^d$	The matrix of d-dimensional node features, and the feature vector of the node $i \in V$.
$\boldsymbol{I} \in \mathbb{R}^{n_V \times n_V}$	The identity matrix.
$\boldsymbol{A}, \tilde{\boldsymbol{A}} \in \mathbb{R}^{n_V \times n_V}$	The adjacency and normalized adjacency matrix of a graph.
$\boldsymbol{D}, \tilde{\boldsymbol{D}} \in \mathbb{R}^{n_V \times n_V}$	The degree and normalized degree matrix of a graph.
$\boldsymbol{L}, \tilde{\boldsymbol{L}} \in \mathbb{R}^{n_V \times n_V}$	The Laplacian and normalized Laplacian matrix of a graph.
$N(i) \subset V$	The first order neighborhood of the node $i \in V$.
$\boldsymbol{W} \in \mathbb{R}^{d \times d'}$	The trainable weights of a layer.

Table 1. List of the mathematical symbols used throughout the paper, and their meaning. A few symbols used only in specific settings are omitted and defined in the text.

number of open problems in GNN explainability that we believe deserve further investigation. Following previous analyses [1, 55, 58, 80, 121], our study investigates message-passing GNNs applied to static graphs. Exploring explainability in non-message passing architectures (e.g., Transformers) or different graph types (e.g., temporal graphs) is an interesting direction for future research, that may however necessitate distinct benchmarking approaches.

The remainder of the paper is structured as follows: Section 2 presents an overview of GNN architectures, with greater detail on the models that we adopted in our study. On the same vein, Section 3 introduces the explainers for graph models, while Section 4 describes the benchmark datasets. Section 5 presents the evaluation metrics employed to asses the explanation's quality. Section 6 summarizes how we trained the tested architectures, and Section 7 presents the results expressed with respect to the research questions defined above. In Section 8 we discuss the results. Finally, in Section 9 we propose future research directions and in Section 10 we draw some conclusions.

2 Graph Neural Networks

In this section we first introduce the notation to deal with the GNN formalism, then we review the GNN architectures explicitly used in our study. To facilitate the reading of the paper, recurring mathematical notation is summarized in Table 1.

We consider a graph $G := (V, E, \boldsymbol{X}^V, \boldsymbol{X}^E)$, with $n_V \in \mathbb{N}$ nodes $V := \{1, \dots, n_V\}$, $n_E \in \mathbb{N}$ edges $E \subset V \times V$, a matrix of d-dimensional node features $\boldsymbol{X}^V \in \mathbb{R}^{n_V \times d}$, where the i-th row of $\boldsymbol{X}$ is the vector of $d \in \mathbb{N}$ features of the i-th node, and similarly a matrix of d'-dimensional edge features $\boldsymbol{X}^E \in \mathbb{R}^{n_E \times d'}$. Most graphs considered in this paper do not have edge features, and we will simply write $G := (V, E, \boldsymbol{X})$ in order to denote a graph with node features only. We use the matrices $\boldsymbol{A}, \boldsymbol{L}, \boldsymbol{I}, \tilde{\boldsymbol{A}}, \tilde{\boldsymbol{D}}, \tilde{\boldsymbol{L}} \in \mathbb{R}^{n_V \times n_V}$, where $\boldsymbol{A}$ and $\boldsymbol{L}$ are the adjacency and Laplacian matrices of G, $\boldsymbol{I}$ is the n_V-dimensional identity matrix, $\tilde{\boldsymbol{A}} := \boldsymbol{A} + \boldsymbol{I}$, $\tilde{\boldsymbol{D}}$ is its diagonal matrix, and $\tilde{\boldsymbol{L}} := \frac{2}{\lambda_{\max}(\boldsymbol{L})}\boldsymbol{L} - \boldsymbol{I}$ is the scaled and normalized Laplacian, where $\lambda_{\max}(\boldsymbol{L})$ is the largest eigenvalue of $\boldsymbol{L}$. Furthermore, $N(i) := \{j \in V : (i, j) \in E\}$ is the first order neighborhood of the node $i \in V$.

Each GNN layer takes as input the graph G, and maps the node features $\boldsymbol{X} \in \mathbb{R}^{n_V \times d}$ to updated node features $\boldsymbol{X}' \in \mathbb{R}^{n_V \times d'}$ for a given $d' \in \mathbb{N}$. Some specific GNN layers, like Hierarchical Pooling layers [13, 52, 116, 133], instead of refining the embedding for each input node aggregate nodes in order to coarsen the graph in a similar way as done by pooling methods for vision models [16, 50, 100], thus resulting in a node feature matrix $\boldsymbol{X}' \in \mathbb{R}^{n' \times d'}$ where $n' < n_V$. Overall, this new feature matrix $\boldsymbol{X}'$ is the embedding or representation of the nodes after the application of one layer of the network. When needed, we denote as $\boldsymbol{X}_i, \boldsymbol{X}'_i$ the original and transformed feature vector of the i-node, i.e., the transpose of the i-th row of the matrices $\boldsymbol{X}, \boldsymbol{X}'$. Specifying the map $\boldsymbol{X} \to \boldsymbol{X}'$ is thus sufficient to provide a full definition of the different layers. These transformations are parametric, and they depend on trainable weights that are learned during the optimization of the network. We represent these weights as matrices $\boldsymbol{W}$. Additional terms specific to single layers are defined in the following.

After an arbitrary number t of GNN layers stacked in sequence, the node embedding matrix $\boldsymbol{X}^{(t)}$ is further processed in a way that depends on the task to perform. In node classification settings [41, 49], where the aim is predicting one or more node properties, a Multi-Layer Perceptron (MLP) [38] (with shared parameters across nodes) is applied to each node's embedding independently in order to output its predicted class. For graph classification settings [41] instead, where the goal is predicting a label for the entire graph, a permutation invariant aggregation function (like mean, max, or sum) is applied over nodes' embedding to compress $\boldsymbol{X}^{(t)}$ into a single vector which is then mapped to the final prediction via a standard MLP.

With this notation settled, we can now fully define the architectures that we are going to consider. In selecting the architectures to be included in our study, we relied on the comprehensive taxonomy of GNN methods published by Zhou et al. [133]. Since our goal is to provide an extensive overview of explainability methods for GNNs, we selected the models to benchmark aiming at covering as much as possible the different categories of the taxonomy. The specific methods are also selected depending on their popularity, their ease of training, their performance on our benchmark datasets, their code availability and their compatibility with the explainers being investigated. Overall, we analyzed the following categories: Convolutional whose computation can be roughly intended as a generalization of the convolution operation on the image domain. Such convolution can either be Spectral [19, 49], theoretically grounded in graph signal processing [97], or Spatial [32, 36, 103, 114], where the operations are usually defined in terms of graph topology; The Pooling category contains all approaches that aggregate node representations in order to perform graph-level tasks. They can be further differentiated into Direct [104, 125], where nodes can be aggregated with different aggregation strategies, often called readout functions, and Hierarchical [11, 13, 52, 116, 118], where nodes are progressively hierarchically aggregated based on their similarity. The latter methods often allow one to cluster nodes both based on their features and their topological neighborhood [11, 116]. Despite covering the major aspects of GNN architectures, the aforementioned taxonomy lacks some of the fundamental works that we will analyze in our study. Particularly, to compensate that, we decided to respectively include the Graph Isomorphism Network (GIN) [114] and the GraphConv Higher Order Network (GRAPHCONV) [69] as Spatial Convolution and Higher Order, the latter being a new category added to the taxonomy. A summary of such categorization end a comprehensive description of the GNN architectures employed in this study is provided in Supplementary Material A.

3 GNN Explainability

To analyze and understand the strengths and weakness of graph explanation algorithms, we selected instances of GNN explainers which are representative of the current state of the art. To this end, we follow the systematization proposed by Yuan et al. [120], and choose to investigate instance-based explainers [8, 26, 42, 57, 59, 63, 81, 87, 88, 91, 94, 98, 100, 101, 105, 115, 122, 128, 132], i.e., those which aim at identifying components of the input that are responsible for the model's output. This is in contrast with model-based explainers, which rather try to provide a global understanding of a trained model [5, 65, 66, 108, 119]. Since the available model-based explainers are very

heterogeneous (i.e., it is not available a unified evaluation setting), and since previous works on benchmarking graph explainers have focused on instance-based methods [1, 2, 55, 80, 120, 131], we thereby omit model-based explainers. In particular, Yuan et al. [120] identifies four macro categories of instance-based explainers, namely *gradient-*, *perturbation-*, *decomposition-* and *surrogate-based* models. Roughly speaking, gradient-based explainers exploit gradients of the input neural network [73, 100, 101], perturbation-based models perturb the input aiming to obtain explainable subgraphs [26, 64, 86, 115], decomposition-based models try to decompose the input identifying the explanations [8, 73, 87], while surrogate-based models use a simple interpretable surrogate to explain the original neural network [42, 105, 128]. Furthermore, in order to account for a number of new approaches based on modeling the underlying graph distribution via a generative process, and following the categorization proposed by Kakkad [45], a fifth category named *generation-based* is added [57, 59, 94].

3.1 Explanation masks

Independently from this categorization, a further fundamental distinction is among explainers providing explanations in terms of edge [64, 86, 115, 122] or node masks [8, 73, 87, 100, 101]. Given a graph $G := (V, E, \boldsymbol{X}^V, \boldsymbol{X}^E)$ (with possibly empty node or edge feature matrices - see Section 2), a node explanation mask is a graph $G_{\text{exp}} := (V, E, \boldsymbol{X}^V_{\text{exp}})$ where the node features $\boldsymbol{X}^V_{\text{exp}} \in \mathbb{R}^{n_V \times 1}$ are node explanation weights. Similarly, an edge explanation mask is a graph $G_{\text{exp}} := (V, E, \boldsymbol{X}^E_{\text{exp}})$, where now $\boldsymbol{X}^E_{\text{exp}} \in \mathbb{R}^{n_E \times 1}$ are edge explanation weights. For both nodes and edges, we have hard masks if the weights have binary values in $\{0, 1\}$, and soft masks if they have continuous values in $[0, 1]$. Any soft mask G_{exp} can be converted in an hard mask $G_{\text{exp}}(t)$ by thresholding its weights with a given threshold value $t \in (0, 1)$. Given an hard mask G_{exp}, its complement $G \setminus G_{\text{exp}}$ is another hard mask where the value of binary weights is flipped.

In addition to these two types of masks, a few explainers return also an explanation for the node features [98, 115]. However since single node features are not representative of the underlying topological structure which we are interested in, and in line with most previous works [1, 2, 55, 80, 120, 131], we do not consider single node features' explanations.

3.2 Selection of the explainers

Below we report a brief overview of our benchmark explainers. Despite the existence of other works proposing explainers, which occasionally fall outside the aforementioned categorization [42, 63, 87, 122, 128], we limited our analysis on a subset. More specifically, the criteria for selecting a given explainer can be roughly summarized by i) representativity of a specific category as outlined before; ii) code availability; and iii) feasibility of usage, i.e., whether the explainer is not too computationally heavy to be used.

Given a GNN g to be explained, let $g(e)^c = y^c = (w^c)^T e$ be the prediction of the model where e corresponds to the final graph-level or node-level embedding, and where the vector $w^c \in \mathbb{R}^{d'}$ contains instead the learned Fully Connected weights for class c to perform the final classification. Furthermore, we denote with $\mathrm{H}^c(i)$ the importance attributed to a given explainer to the node $i \in V$ for the prediction of class c, i.e., the collection of the values $\mathrm{H}^c(i)$ for $i = 1, \ldots, n_V$ provides the node explanation mask $\boldsymbol{X}^V_{\text{exp}}$ for class c. A detailed description of the methods can be found in Supplementary Material B. A concise summary of these methods is presented in Table 2.

4 Benchmark datasets

In this section we present the graph benchmark datasets employed in our work, the majority of which represent newly proposed datasets. In designing the new benchmarks, we took inspiration from Faber et al. [24] who analyzed frequent biases in evaluating GNN explainers and pointed out that explainers should be evaluated on controlled benchmarks where the ground-truth evidence for target labels is known, and that different benchmarks

Name	Category	Task	Mask type
GradExplNode	Gradient	Graph/Node	Node
GradExplEdge	Gradient	Graph/Node	Edge
GuidedBp	Gradient	Graph/Node	Node
IgEdge	Gradient	Graph/Node	Edge
IgNode	Gradient	Graph/Node	Node
GradCam	Gradient	Graph/Node	Node
GnnExpl	Perturbation	Graph/Node	Edge
PgExpl	Perturbation	Graph/Node	Edge
SubX	Perturbation	Graph/Node	Edge
PgmExpl	Surrogate	Graph/Node	Node
Cam	Decomposition	Graph/Node	Node
RgExpl	Generation	Graph/Node	Edge

Table 2. Summary of explainers analyzed in this work. The columns Task represents to which downstream task the explainer can be applied to, while Mask type represents whether the explainer returns explanations in terms of entire node importance, single node features importance, or edge importance.

should aim at testing different aspects of the GNN. In the following we describe in detail each dataset, both for node and graph classification.

4.1 Datasets for graph classification

Grid: Inspired by the benchmarks presented in Ying et al. [115], the Grid dataset is composed by 1000 Barabási-Albert (BA) graphs [9]. To half of these 1000 graphs we attach a 3×3 grid, and the resulting graphs are assigned to the positive class, while the ones without grid are the negative class. The number of nodes in the BA graph is a uniformly distributed random number between 15 and 30 (for the negative class) and between 6 and 21 (for the positive class). This guarantees that when adding the grid, the average number of nodes in the positive class matches the one in the negative class. It is worth mentioning that in the experiments done by Ying et al. [115], the total number of nodes is fixed. This benchmark evaluates the ability of the explainers to identify explanations consisting of a simple connected pattern.

Grid-House: Grid-House is characterized by two concepts. A 3×3 grid, as in the previous benchmark, and a house made of 5 nodes. The base structure, to which the concepts are attached, is a BA graph with a random number of nodes, and the final task corresponds to binary classification. The negative class consists of a BA graph connected to a grid or a house, while the positive class is composed by a BA graph connected to both a grid and a house. This benchmark aims at evaluating compositionality, as identifying simple patterns in isolation is insufficient to characterize the ground truth.

Stars: The Stars benchmark is characterized by a random graph connected to a variable number of star-shaped structures (from one to four). For the random graph generation, this time, we opted for the Erdős-Rényi (ER) random graph model [23] to avoid a possible interference of stars generated in a BA graph. We defined a three-class classification task, depending on the number of stars present in each sample: class 0 corresponds to 1 star, class 1 to 2 stars, and class 2 to 3 or 4 stars. Each star has a fixed size of 16 nodes, and the total number of nodes is uniformly distributed between 30 and 50. This benchmarks is aimed at evaluating how explainers deal with counting substructures.

House-Color: None of the previous benchmarks involves node features. Thus, in this benchmark we test how node features affect explanations. In particular, we have a random BA graph with (one-hot encoded) random colored nodes (blue, green, red). To the base BA graphs, we attached from one to three house-like structures

made of five nodes. One of these houses has a uniform color, which is blue for the negative class and green for the positive one. The other houses have random colors.

4.2 Datasets for node classification

Shapes: The BA-Shapes dataset (henceforth referred to as Shapes), introduced in Ying et al. [115], is a widely used dataset for benchmarking GNN explainers [64, 94, 115, 120, 122, 131]. It is composed by 300 nodes and a set of 80 five-node house-structured network motifs which are randomly attached to the base graph, generated following the BA model [9]. Nodes are assigned to four categories, namely they either do not belong to a house (class 0), or they are classified depending on their structural function in the house: they may be either on the middle below the roof (class 1), on the base (class 2), or on the top of the roof (class 3). The expected ground truth explanation is a house motif for all classes.

Infection: This benchmark graph has been introduced in Section 5.1 of Faber et al. [24], and we use it with minor modifications[1]. Starting from a directed ER graph [23] with 1000 nodes and an edge-generation probability $p = 0.004$, a set of 50 nodes is selected uniformly at random and identified as infected. The state of each node is mapped to a node feature by one-hot encoding, i.e., a node has feature $[0, 1]$ if it is healthy and $[1, 0]$ if it is infected. The label of a node is defined based on the length of a minimal directed path to reach this node from an infected one. Namely, if this distance is denoted as d, then the node has label 0 if $d = 0$ (i.e., the node itself is infected), label 1 if $d = 1$ or $d = 2$, and label 2 otherwise, i.e., $d \geq 3$. We remark that the original dataset included 5 classes ($d = 0, 1, 2, 3$, and $d \geq 4$), but we restrict to three to simplify the presentation of the results. Accordingly, we employ two-layer networks instead of the four-layer ones used in [24]. From the definition of the dataset we identify the expected ground truth explanations of a node v as follows. For label zero, the explanation is the node itself. For label 1, any directed path of length one or two from an infected node to v is a valid explanation. For label 2, the explanation is given by the union of all directed paths of length up to 2 from any node in the graph to v. In the last case, indeed, the network has to check the entire set of nodes from which v is reachable in at most two steps to exclude that any of them is infected.

5 Assessment of the explanation quality

Evaluating GNNs' explanations is a challenging task that requires to verify if and how the explainer is effective in capturing the behaviour of the model. There are two main strategies to evaluate explanation quality. The first is a supervised strategy [27, 85, 115], that measures the similarity of the extracted explanation with an existing ground-truth, which is assumed to be known. The second strategy measure in an unsupervised manner how much the prediction of a GNN on the full graph resembles the prediction computed on the extracted explanation only. Note that this does not require to have a ground-truth explanation available. We consider a metric for each of these two strategies, in order to capture different aspects of the quality of an explanation: the plausibility[80] of the explanation with respect to a ground-truth concept that an accurate GNN is expected to have learned, and the fidelity[20] of the explanation with respect to the prediction of the GNN to be explained. Specifically, with plausibility we quantify the consistency between the explainer mask and a human-level intuition of what a plausible explanation looks like. On the other hand, fidelity measures the consistency between the model prediction on the full graph and the prediction on the explanation subgraph, and thus it works with a sort of model-based instead of human-based ground truth.

In this work we will evaluate explainers according to both strategies, and study the trade-off between the two.

[1]The code to generate this benchmark can be found at https://github.com/m30m/gnn-explainability.

5.1 Single-instance metrics

In the following we detail the metrics we employ, namely plausibility[80] (P) and fidelity (F), the latter further divided in its comprehensiveness[20] (F_{com}) and sufficiency[20] (F_{suf}) components.

Each of the scores or metrics is computed for a specific instantiation of a dataset with $n_c \in \mathbb{N}$ classes, a class $c \in \{0, 1, \ldots, n_c - 1\}$, a model, and an explainer. We thus assume that these four are fixed in the following, and we stress that the same computation has to be repeated for each of these configurations. Moreover, we remark that the metrics are computed on the training set alone, as we need access to the labels of the graphs or nodes.

We assume to have a graph G (for graph classification tasks) or a node $v \in G$ (for node classification ones) of class $y = c$, and denote as g the trained GNN. We have GNNs which output a class probability prediction vector in form of a soft max, so that the predicted class probabilities sum to 1. Since we are considering one class at a time, in the following we assume to be working with only the output's entry corresponding to class c.

Only the graphs (or nodes) which are correctly classified by the trained GNN are considered further and run through the explainer, which returns a corresponding node or edge soft explanation mask G_{exp} (see Section 3.1).

Before computing the metrics, these soft-mask explanations are processed and filtered by means of three operations:

- Conversion: Edge masks are converted to node masks by assigning to each node the weight given by the average of the weights of its incident edges. This operation makes it easier to compare the scores of edge-based and node-based explainers, and we choose to use node masks since node-based explainers are more common in our taxonomy (see Section 2).
- Filtering: For each mask we check the difference between the largest and the smallest weight. If the difference is below a tolerance $\tau = 10^{-3}$, we discard the graph or node for the given combination of dataset, class, model, and explainer (the graph or node may still pass the filter for other settings). The goal of this filter is to discard poorly informative explanations.
- Normalization: The remaining explanation masks are normalized instance by instance, so that each explanation has weights in $[0, 1]$. This has the effect of making the computation of the metric uniform across the entire dataset, and comparing its values to those obtained with other settings.

After these operations have been applied, we compute the metrics as follows. We formalize each metric as it is computed on a single instance (a graph or a node), and remark that the overall values of plausibility or fidelity for the entire (dataset, class, model, explainer)-configuration is obtained by averaging over these single instances.

Plausibility. Let $\overline{G}_{\text{exp}}$ be the expected ground truth for class $c \in \{0, 1, \ldots, n_c - 1\}$, represented by a copy of the original graph G with an hard mask highlighting the ground truth nodes. Following [80], the plausibility P of the explanation is defined as

$$P = \text{AucROC}(G_{\text{exp}}, \overline{G}_{\text{exp}}),$$

i.e., the area under the ROC curve between the computed soft mask and the ground truth hard mask.

It is clear that this metric can only be computed on benchmarks in which the ground truth explanation can be defined, and it is completely dependent on this definition. For each dataset, the ground truths that we are using to compute P are defined in Section 4. Whenever multiple ground truths are possible (e.g., the shortest paths in Infection), we compute the plausibility of each candidate and consider only the highest one.

Sufficiency. The fidelity sufficiency F_{suf} [20] is the difference in the predicted probability when computed on the graph and on the explanation. Since the explanation is a soft mask, we fix a number of levels $N_t \in \mathbb{N}$ and apply an incremental thresholding with $N_t + 1$ threshold levels $t_k = k/N_t$, $k = 0, \ldots, N_t$, where we define $G_{\text{exp}}(t_k)$ to be the hard mask explanation derived from G_{exp} with threshold t_k.

Using $N_t = 100$, we define the metric by

$$F_{suf} = \frac{1}{N_t - 1} \sum_{k=1}^{N_t - 1} \left(g(G) - g(G_{\exp}(t_k)) \right),$$

i.e., the average change in prediction over all the possible hard masks.

This metric may possibly be negative, and a smaller value indicates a better result. This indeed may happen only if the explanation provides an higher probability for the correct class than the entire graph, and thus the explanation mask manages to filter unnecessary parts of the graph. For this reason this metric is harder to compare to other scores, so when used alone we transform it to a renormalized metric F'_{suf}, which has values in $[0, 1]$ and where $F'_{suf} = 1$ means a good quality of the explanation. The normalization takes into account the number of classes n_c, and it is defined for $p = \frac{n_c - 1}{n_c}$ as

$$F'_{suf} = 1 - \frac{F_{suf} + p}{1 + p} = \frac{n_c}{2n_c - 1}(1 - F_{suf}).$$

Comprehensiveness. The fidelity comprehensiveness F_{com} [20] is instead the difference in the predicted probability when computed on the graph and on the complement of the explanation. Proceeding as in the computation of the sufficiency, we define

$$F_{com} = \frac{1}{N_t - 1} \sum_{k=1}^{N_t - 1} \left(g(G) - g(G \setminus G_{\exp}(t_k)) \right),$$

where now $G \setminus G_{\exp}(t_k)$ is the complement of the hard mask $G_{\exp}(t_k)$. This metric may as well assume negative values, but good explanations have in this case F_{com} close to 1 (the complement of the explanation provides low probability).

We are not using this metric for node classification datasets, since its evaluation would require to compute the model prediction on $G \setminus G_{\exp}(t_k)$, which is a graph that may possibly not contain the node whose classification we are willing to explain.

Fidelity. To aggregate F_{com} and F_{suf} into a unique fidelity metric, for graph classification tasks we compute what we call $f1$-fidelity (F_{f1}), which is defined by

$$F_{f1} = 2\frac{(1 - F_{suf}) \cdot F_{com}}{(1 - F_{suf}) + F_{com}}.$$

This is indeed the $f1$ score [44] between F_{com} and $(1 - F_{suf})$. In graph classification tasks, we use this metric in place of F_{com} and F_{suf}.

5.2 Aggregation

After we evaluate any of these metrics on each (dataset, class, model, explainer)-configuration, we need an aggregation mechanism to assign a unique score to the models and the explainers over all classes and datasets. This permits to avoid visualizing the detailed metrics over the entire set of configurations, and make the results easier to be interpreted. However, we provide plausibility and fidelity values for each (dataset, class, model, explainer)-configuration in Supplementary Material C.

To define these aggregate metrics we proceed as follows, where the same procedures are repeated for both plausibility and fidelity: *(1)* For each (dataset, class, model, explainer)-configuration we keep only the class with the highest value of the metric, i.e., the best explained class. *(2)* For a given dataset, we rank the model-explainer pairs according to the values selected in point (1). The aggregated score of each pair is the ranking number $1, 2, \ldots$. *(3)* The dataset-level scoring of an architecture, of an explainer, or of a category of explainers (e.g., grad-based or

edge-based) is the average of the scores of point (2) over all the corresponding pairs. *(4)* To obtain global answers (over all the datasets), the scores of point (2) are averaged over the datasets, and the operations of point (3) are repeated using these values.

We mention in particular that the answer to the research questions (Section 7.1) are computed from these aggregations.

To assess instead the stability the explanations over an entire dataset, we propose a qualitative visualization of the masks which is discussed for each experimental setting.

6 Experimental setting

Any explainer provides an explanation of the prediction of a given instance of a model, as it is obtained after an optimization process on a specific dataset. It is thus of paramount importance to identify the choices made in the training of the networks that will be analyzed in the following.

Graph classification. We report in Table 3 the details of the networks used in each graph classification task, the parameters used for their optimization, and the resulting train and test accuracies.

For each dataset and each architecture, the table shows the dimensions of the hidden layers of GNN type (column *GNN*) and of fully connected type (column *Fully conn.*), and any additional parameter used for the definition of the architecture. For example, in the first row the numbers $30 - 30 - 30$ and $10 - 2$ mean that three Gcn layers are applied, each mapping to a target dimension of 30, followed by two fully connected layers with target dimensions 10 and 2. We remark that the final aggregation function is a mean for all datasets except for Stars, where we used a sum aggregation. The table additionally reports the learning rate (column *LR*) and number of epochs used in the training, where an ADAM optimizer has been used in each case. We remark that these configurations have been chosen with the guiding principle of obtaining the simplest configuration achieving a target 0.95 train accuracy. In particular, no validation set has been used. This choice is motivated by the fact that the explanations are computed on the training set. This is a common (sometimes implicit) choice in the explainability literature, with the rationale that explanations should identify what the model learned during training, possibly highlighting patterns that do not generalize to test examples. However, to avoid extracting spurious explanations for models that did not learn anything sensible, only the models with a training accuracy of at least 0.95 have been further analyzed, while all the others have been discarded. The last two columns of Table 3 show the resulting train and test accuracies obtained by the models trained according to these specifications, where an "X" indicates that it was not possible to achieve the desired target accuracy (in this case, the row reports the configuration of the largest architecture which has been tested). It is easy to see that with a 0.95 threshold on training accuracy, training and test accuracies end up being very similar.

Node classification. In the same way, we report in Table 4 the configurations and accuracies related to the node classification tasks.

7 Results

7.1 Research questions

The comparative analysis of the behavior of the explainers is developed along the following research questions, which will apply to both node and graph classification tasks.

- **RQ1: How does the architecture affect the explanations?** This research question can be naturally divided into the following three subquestions:
  - **RQ1.1:** Which is the architecture that has the best explanation? With this question we would like to understand which is the architecture that achieves the best score, either in terms of f1-fidelity or plausibility.

Dataset	Architecture	GNN	Fully conn.	HyperParams	LR	Epochs	Train Acc	Test Acc
Grid	Gcn	30-30-30	10-2	-	0.001	1500	0.994	0.998
	GraphSage	30-30-30	10-2	-	0.01	3000	X	X
	Gat	30-30-30	10-2	heads = 1	0.01	3000	X	X
	Gin	30-30	30-2	-	0.001	1000	1.0	1.0
	Cheb	30-30	30-2	-	0.001	1000	1.0	1.0
	MinCutPool	32-32-32	32-2	-	0.001	700	0.92	0.93
	Set2Set	30-30-30	10-2	-	0.001	1500	0.97	0.97
	GraphConv	30-30	30-2	-	0.001	500	1.0	1.0
Grid-House	Gcn	60-60-60-60	60-10-2	-	0.001	7000	0.97	0.97
	GraphSage	60-60-60-60	60-10-2	-	0.01	3000	X	X
	Gat	60-60-60-60	60-10-2	heads = 3	0.01	3000	X	X
	Gin	30-30	30-2	-	0.001	1000	0.99	1.0
	Cheb	30-30-30	30-2	-	0.001	1000	1.0	0.98
	MinCutPool	32-32-32	32-2	-	0.001	700	0.95	0.95
	Set2Set	60-60-60-60	60-10-2	-	0.001	1500	0.97	0.97
	GraphConv	30-30	30-2	-	0.001	500	1.0	1.0
Stars	Gcn	70-70-70	30-3	-	0.005	1000	0.99	1.0
	GraphSage	30-30-30	30-3	-	0.01	3000	X	X
	Gat	30-30-30	10-3	heads = 1	0.01	3000	X	X
	Gin	40-40	30-3	-	0.001	3000	0.99	1.0
	Cheb	30-30	30-3	-	0.001	1000	0.99	0.99
	MinCutPool	32-32-32	32-3	-	0.001	400	0.99	0.99
	Set2Set	70-70-70	30-3	-	0.001	1500	0.99	0.99
	GraphConv	30-30	30-3	-	0.001	500	0.99	0.99
House-Color	Gcn	30-30	15-2	-	0.001	4000	0.99	0.99
	GraphSage	30-30-30	30-2	-	0.001	1000	1.0	0.99
	Gat	10-20-40	10-2	heads = 2	0.001	500	0.99	0.99
	Gin	30-30	30-2	-	0.001	1000	1.0	0.99
	Cheb	30-30	30-2	-	0.001	500	1.0	1.0
	MinCutPool	32-32	32-2	-	0.001	400	0.96	0.97
	Set2Set	30-30	15-2	-	0.001	1500	1.0	0.99
	GraphConv	30-30	30-2	-	0.001	500	1.0	1.0

Table 3. Configuration of the graph classification models, and corresponding accuracies. The table reports for each dataset and each architecture the dimension, number, and hyperparameters defining the hidden layers, together with the optimization parameters, and the obtained train and test accuracies. The configuration-dataset pairs which did not reach the target 95% train accuracy are marked with an "X", and are not further analyzed in this work.

  - **RQ1.2:** Which is the easiest architecture to explain? This question aims to find what is the architecture that is well explained by the greatest number of explainers.
  - **RQ1.3:** Which is the hardest architecture to explain? In this case we want to search for an architecture that achieves the lowest score.
- **RQ2: How do explainers affect the explanations?** Even this question can be divided into subquestions, which try to cover different open problems related to state-of-the-art GNN explainers. We identify them as follows:
  - **RQ2.1**: Which is the explainer that explains in the best way? Here we are interested in finding the explainer able to obtain the highest plausibility or fidelity.

Dataset	Architecture	GNN	Fully conn.	LR	Epochs	Train Acc	Test Acc
Shapes	Gcn	30-30-30	10-4	0.0005	2000	0.98	0.98
	GraphSAGE	30-30	4	0.005	2000	1.0	1.0
	Gat	30-30-30	10-4	0.0005	2000	X	X
	Gin	70-70-70	4	0.0005	5000	0.96	0.95
	Cheb	30-30	4	0.0005	300	1.0	1.0
Infection	Gcn	30-30	30	0.005	500	0.951	0.950
	GraphSAGE	30-30	30	0.005	500	1.000	0.995
	Gat	30-30	30	0.005	500	0.977	0.975
	Gin	30-30	30	0.005	500	0.952	0.950
	Cheb	30-30	30	0.0005	600	0.993	0.950

Table 4. Configuration of the node classification models, and corresponding accuracies. The table reports for each dataset and each architecture the dimensions and number of hidden layers, together with the optimization parameters, and the obtained train and test accuracies. The configuration-dataset pairs which did not reach the target 95% train accuracy are marked with an "X", and are not further analyzed in this work.

  - **RQ2.2**: Which is the explainer that explains the maximum number of architectures? This aspect is particularly important because we need explainers which are robust with respect to different GNN architectures.
  - **RQ2.3**: Which is the category of explainers that provides the best explanations? The subquestion searches for the best category of explainers. As defined by [120], we consider three macro-categories, namely gradient-based (Grad), perturbation-based (Pert), and decomposition-based (Dec).
  - **RQ2.4**: Which is the best mask type between node and edge? By answering this question we investigate if there is an advantage for explainers based on node or edge importance.

  We would like to remark that **RQ2.3** and **RQ2.4** are particularly relevant for future research in GNN explainability, since they may provide actionable guidelines for the development of new explainers.
- **RQ3: How do different types of problems affect the explanations?** To address this question we analyze a series of different problems such as: single versus multiple concepts, counting substructures and others. Each problem has been investigated throw different datasets, that will be discussed in relation to this question.

7.2 Graph classification

7.2.1 RQ1: How does the architecture affect the explanations? Table 5 visualizes in a compact form the answer to research questions **RQ1**, where the aggregation mechanism described in Section 5.2 has been used to identify a ranking of the architectures for each dataset across all explainers, both in terms of plausibility and fidelity and the highest ranking architecture is reported for each research question and dataset. Each column in the table refers to a specific dataset (Grid to House-Color, see Section 4), while the first column shows the overall answer obtained by considering all the datasets at once. Also in this case we refer to Section 5.2 for the details of the computation of this ranking.

Although the architecture with the best explanation (**RQ1.1**) varies depending on the dataset, the overall best performer is GraphConv both for plausibility (paired with GradExplEdge) and fidelity (paired with IgEdge). This indicates a sort of consistency in the performances of GraphConv: It may not be the single best one for any dataset, but it is always among the best performing ones, in a way that makes it to be the best in terms of aggregated scores.

The overall easiest architecture to explain on average (**RQ1.2**) is Gcn for both plausibility and fidelity, and even at the single dataset level it prevails in three out of five datasets for plausibility, in four out of five for fidelity. Moreover, the two metrics agree to identify Gcn as the easiest to explain in three out of five datasets. A possible motivation behind this result may come from the fact that Gcn are among the simplest models based on message passing, even if other architectures (e.g., GraphSage and Gin) are equally simple. A difference so large may thus be related to other aspects that we are unable to identify. Moreover, we remark that Gcn and Set2Set work with the same underlying GNN layers (Gcn), and they differ only in the final aggregation operation (a sum in Gcn, and an LSTM in Set2Set). The better performances of Gcn are thus hinting to the fact that a different global aggregation alone is responsible for changing a network's explainability, and that linear aggregations (Gcn) are, perhaps unsurprisingly, easier to explain than nonlinear ones (Set2Set). In general terms, the role of the global aggregation function and its stability across different tasks is yet to be fully understood and has not received great attention in the explanation literature. Thus, we believe that a systematic study in this direction may be an interesting future topic of research.

Finally, from the table it is easy to see that Gin is the most difficult network to explain (**RQ1.3**), for both plausibility and fidelity in each dataset except for Stars. This stark difficulty in explaining Gin is not directly understandable, especially because it is implemented with a single-layer MLP, which does not introduce stronger non-linearity than a simpler Gcn. A possible explanation for this behaviour is the fact that most explainers have been developed and tested on Gcn, potentially introducing a bias that affects the performance of that architecture. This aspect warrants further investigation in future studies.

Moreover, the difficulty in explaining MinCutPool on Stars may be due to the fact that in this case the task is to count the number of occurrences of a concept (the stars) in a dataset. Although the concept itself is the easiest possible to identify for message passing based GNNs, it seems that concept-counting is hard to explain for a pooling mechanism.

Plausibility

	All	Grid	Grid-House	Stars	House-Color
RQ1.1	**GraphConv**	GraphConv	Cheb	Set2Set	Gcn
RQ1.2	**Gcn**	Gcn	Gcn	Set2Set	MinCutPool
RQ1.3	**Gin**	Gin	Gin	MinCutPool	Gin

Fidelity

	All	Grid	Grid-House	Stars	House-Color
RQ1.1	**GraphConv**	GraphConv	Set2Set	GraphConv	Set2Set
RQ1.2	**Gcn**	Gcn	Gcn	GraphConv	Gcn
RQ1.3	**Gin**	Gin	Gin	MinCutPool	Gin

Table 5. Experimental answer to **RQ1** for graph classification. The table shows the top-ranking architecture with respect to each subquestion **RQ1.1**, **RQ1.2**, **RQ1.3**, for each dataset Grid to House-Color, and overall. The rankings are computed with respect to the plausibility and the fidelity metrics, and the colors identify different architectures.

To offer additional insight into this fine-grained behavior of Gcn, which is the easiest architecture to explain according to RQ1.2 in terms of both metrics, we pick a random element from each dataset and analyze the mask provided by each explainer. Figure 1 reports these masks, where the rows show the representative graph from each dataset, and each column corresponds to a different explainer. The first five explainers return node importance, while the last four are edge-based. In both cases, the node or edge importance is rendered by a different color intensity. We stress once again that only one graph per dataset is shown in the figure, and thus the following discussion is of a rather qualitative nature, to be complemented with the metrics discussed in the first part of this section.

Gcn manages to achieve good results in terms of both plausibility and fidelity, meaning that its explanations are both close to the expected ground truth and to the actual one used by the models to realize their prediction. This duality can be observed across the examples of Figure 1. Indeed, there are cases where the masks identify clearly the grids in Grid (Cam, GradCam, IgNode, GradExplEdge, PgExpl), the grid, the house and the path connecting them in Grid-House (most node-based explainers, and PgExpl), the stars in Stars (GuidedBp and GradExplNode), and the colored house in House-Color (GradCam, GuidedBp, GradExplNode). On the other hand, for many other dataset-explainer combinations the mask is less localized and interpretable by a human eye, but the overall high scores of this explainer suggest these explanations could still be good, at least in terms of fidelity and thus from a model perspective, even if they deviate from the expected ground truth. A more in-depth analysis of the actual behavior on each dataset is presented in Section 7.2.3.

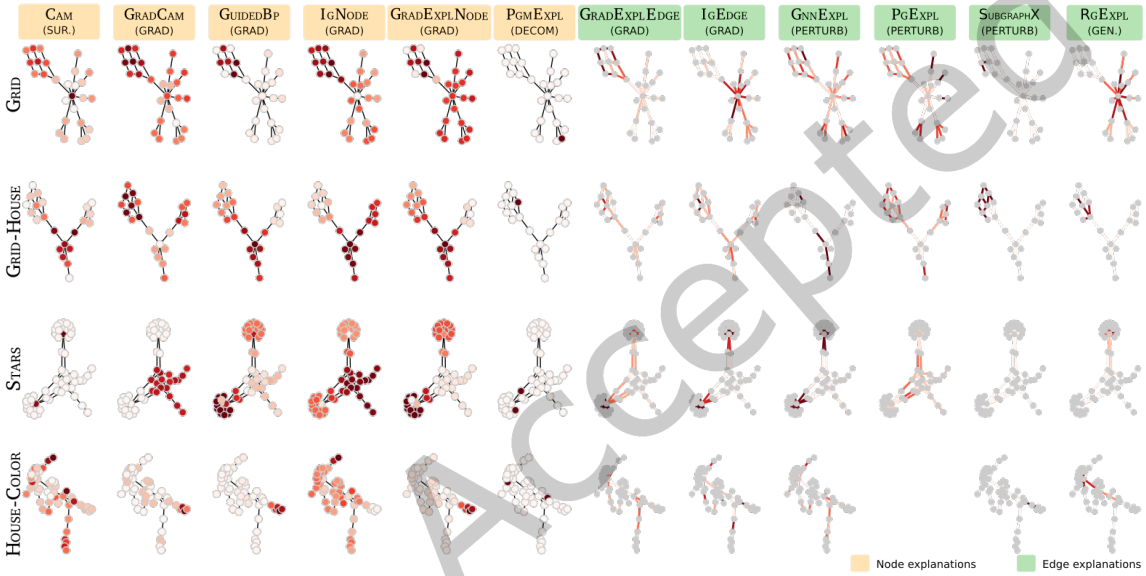

Fig. 1. Explanation masks (node- or edge-based) computed by the different explainers on the predictions of Gcn. Each row visualizes the mask computed for a given random graph from each dataset.

7.2.2 RQ2: How do explainers affect the explanations? The answers to this question are summarized in Table 6, where we used again the aggregation strategies defined in Section 5.2 to establish a ranking of the explainers and select the best ones, both for each dataset and overall.

Interestingly, the overall best explainer is different for plausibility and fidelity in absolute terms (**RQ2.1**), but it is the same (SubX) when looking at the best performing one on average across all architectures (**RQ2.2**). They are however all edge-based.

At the dataset level, it is worth remarking that House-Color is the only one whose absolute best explainer (RQ2.1, GuidedBp) is node-based, and this may clearly be due to the fact that this dataset is the only one with meaningful node features.

In terms of average performances (**RQ2.3**), perturbation-based explainers are those that best explain all the models for plausibility, while generation-based explainers prevail with respect to fidelity, both at the aggregate and single-dataset levels. In the case of plausibility, the single best explainer is instead gradient-based (RQ2.1), but this discrepancy is similar to what happens in node classification (Section 7.3.2). However, while for node

tasks the local gradient-based explainers worked better, here perturbation mechanisms are more effective, and this is understandable since graph classification may benefit from these more global types of explanation.

When looking at the average over the entire groups (**RQ2.4**), edge-mask based explainers are clearly over-performing node-based ones, in accordance with RQ2.1 and RQ2.2. We argue that this may be due to the fact that edge-based explainers have been developed specifically for graph-explanation tasks, while node-based ones are all adaptations of existing explainers, introduced for other settings. We remark once again that this is the case only for graph classification, while for node-based tasks (Section 7.3.2) node-based explainers appear to be superior.

			Plausibility		
	All	Grid	Grid-House	Stars	House-Color
RQ2.1	**GradExplEdge**	RgExpl	PgExpl	IgEdge	GuidedBp
RQ2.2	**SubX**	SubX	PgExpl	GradExplEdge	SubX
RQ2.3	**Pert**	Gen	Pert	Grad	Pert
RQ2.4	**Edge**	Edge	Edge	Edge	Edge

			Fidelity		
	All	Grid	Grid-House	Stars	House-Color
RQ2.1	**IgEdge**	SubX	IgEdge	GradExplEdge	PgExpl
RQ2.2	**SubX**	RgExpl	SubX	GnnExpl	SubX
RQ2.3	**Gen**	Gen	Gen	Gen	Pert
RQ2.4	**Edge**	Edge	Edge	Edge	Edge

Table 6. Experimental answer to **RQ2** for graph classification. The table reports the top-ranking explainer with respect to each subquestion **RQ2.1**-**RQ2.4**, for each dataset Grid to House-Color, and overall. The rankings are computed with respect to the plausibility and the fidelity metrics, and the colors identify different explainers.

Similarly to the previous section, Figure 2 zooms into SubX, which is the best-ranking explainer according to RQ2.1 and with respect to plausibility. The high plausibility of the explainer means that it is effective in identifying the human-expected explanations in the graphs, and this is clearly visible in the examples of Figure 2: with a few exceptions, the dark red edges identify the grid in Grid, a path connecting the grid and the house in Grid-House, the stars in Stars, and the colored house in House-Color.

7.2.3 RQ3: How do different types of problems affect the explanations? To address the third question we analyze the datasets separately. We remark that each dataset has been chosen to represent different types of challenges, which will be discussed in each of the following paragraphs.

Grid. In the first dataset, the concept is a grid attached to a random Barabási–Albert (BA) network. Since the BA component is identical in the positive and negative classes, the only discriminative subgraph is the grid (or part of it). In fact, the minimal discriminant subgraph for this dataset is a square, because the BA component does not contain it.

The left panel of Figure 3 visualizes the performance of each model-explanation pair when applied to this dataset. Each pair is located according to the two-dimensional coordinate given by the resulting fidelity (horizontal axis) and plausibility (vertical axis), and it is identified by the model name and by a color representing the explainer. We use warm colors for node-based explainers, and cold colors for edge-based ones.

This visualization permits to identify those model-explanation pairs that strike the best balance between the two scores, namely, models that maximize both plausibility and fidelity are in the top right corner of the figure. It is first relevant to observe that a clear positive correlation emerges for the top-performing pairs, in the sense that there are no cases where high fidelity is achieved without a correspondingly high plausibility, and

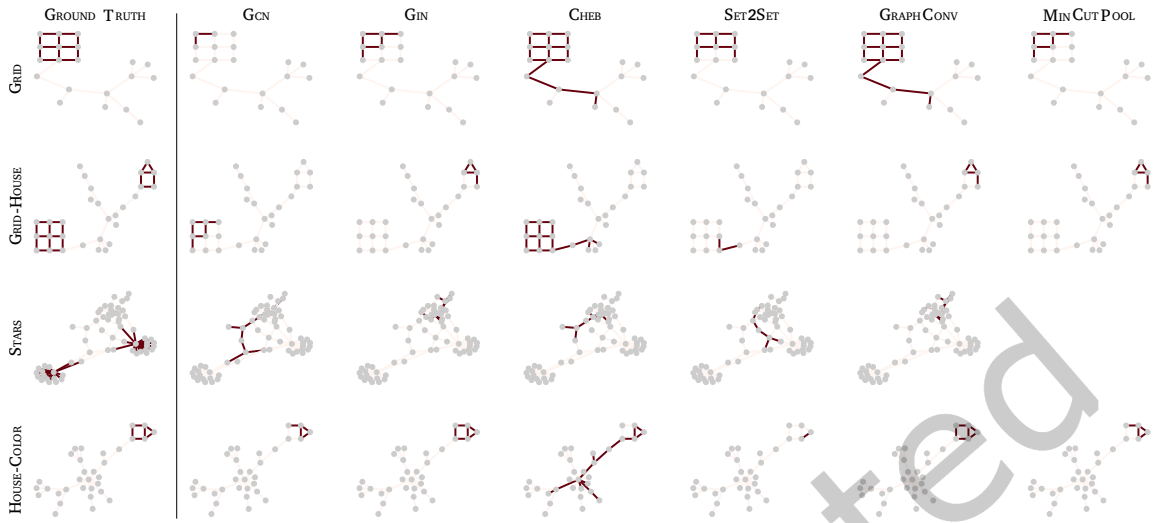

Fig. 2. Explanation masks computed by SubX on the predictions of the different models. Each row visualizes the mask computed for a given random graph from each dataset.

vice-versa. Moreover, the highest plausibility is achieved by GraphConv with RgExpl, while GraphConv with SubX obtains the highest fidelity. These are also the two Pareto-optimal pairs, i.e., any other pair reaches either a smaller fidelity or a smaller accuracy. In this sense, they are the best ones according to this evaluation.

Moreover, it is remarkable to observe that the best performing pairs (thus including also the pairs Gcn-RgExpl, GraphConv-IgEdge, Gin-RgExpl, Cheb-PgExpl) have all edge-based explainers. This fact is in perfect accordance to the answer to RQ2.4 (Section 7.2.2), which identifies this type of explainer as superior to node-based ones. Observe however that Grid has no node features, and this may bias this aspect.

For these two top-performing pairs (GraphConv-RgExpl and GraphConv-SubX) we further investigate the quality of the explanations by quantifying their stability. Namely, we are interested in understanding how the different instances of graphs in the dataset are explained, and if there is any recurring pattern in these explanations.

This stability is shown in the right panel of Figure 3, where each edge in the grid motif is colored according to its importance averaged over all the networks in Grid, with a color scale ranging from white (for importance 0) to dark red (for importance 1). The width of each edge is instead proportional to the standard deviation of the explanation across the dataset, such that thicker edges describe a larger deviation, hence a smaller stability, and vice-versa.

While GraphConv-SubX provides significant explanations (dark red edges), we can observe a very high variability across the dataset (thick edges). On the other hand, GraphConv-RgExpl provides consistent explanations over the entire dataset (thin edges), but with a less pronounced emphasis on the significant edges (light red edges).

To conclude the analysis on Grid, Figure 4 shows a prototypical explanation for each GNN, paired with its best explainer as identified by the highest combination of the two metrics. Overall, we can assert that each GNN can be explained fairly well if the concept is a simple subgraph of the network. As anticipated in subsection 7.2.1, Gin is the hardest to explain, while the best explanations are obtained with gradient, perturbation, and generation-based explanations producing edge masks.

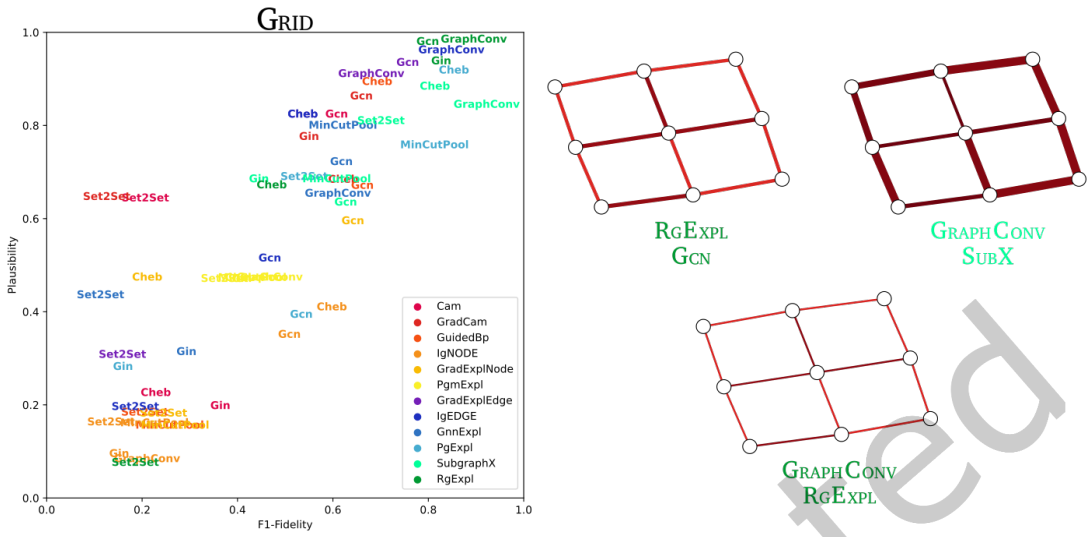

Fig. 3. Left: fidelity and plausibility achieved by all the model-explainer pairs when applied to GRID. In each pair, the name refers to the model while the color identifies the explainer. Right: stability of the explanations for the three top-performing model-explanation pairs. The colors identify important edges (dark red), and the edge thickness the variability of the importance in the dataset.

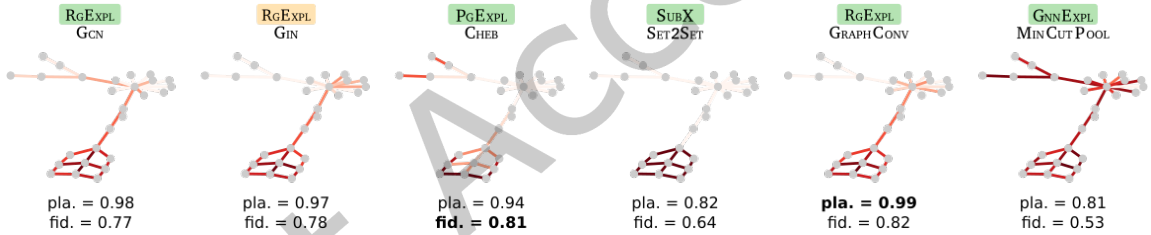

Fig. 4. Examples of explanations provided for each model and its highest plausibility explainer, when applied to a random sample from GRID. The plausibility and fidelity values are those of the entire dataset, as reported in Figure 3

GRID-HOUSE. In this dataset class 0 contains either a house or a grid, while class 1 contains both of them. Thus just identifying the presence of a simple pattern (a grid or a house) is insufficient for discrimination, and the GNN needs to learn how to combine them. In addition to investigating compositionality, this dataset can also help investigating an aspect that we name laziness. Namely, a network can address a binary classification problem by learning patterns characterizing only one of the two classes and predicting the other one when these patterns are absent.

As with GRID, we start with the comparative analysis of plausibility and fidelity for each GNN-explainer pair in the left of Figure 5, where in this case the results are reported for both class 0 (left panel) and class 1 (right panel). Here a remarkable difference can be observed between the two classes since it is clear that for class 1 a linear correlation is present between the two metrics for each model-explainer pair, while for class 0 a high plausibility is associated with a low fidelity, and vice-versa, the only partial exception being the pair GRAPHCONV-RGEXPL, which achieves a good balance between the two metrics. The best explanations are clearly

those for class 1, confirming the laziness phenomenon explained above. This finding suggests that care should be taken in evaluating instance-based explanations, as the reason for predicting a certain class could lie in the absence of evidence in favor of the alternative one.

In more general terms, this result shows a substantial discrepancy between plausibility and fidelity, and it indicates that it is crucial to jointly consider both metrics to properly evaluate GNN explainability.

Moreover, for class 1 it is easy to see that MinCutPool, Cheb and Gcn stay on the Pareto front when paired with PgExpl, which is thus the best explainer in this case. In the top right of the figure, the majority of the colors are cold, corresponding to edge-based explainers. In the case of class 0, instead, it can be verified that the high-plausibility model-explanation pairs (top left corner in the figure) all capture either the house or the grid, but no other structure in the graph. On the other hand, the high-fidelity ones (bottom right, i.e., Set2Set plus IgEdge, MinCutPool plus Cam, and Gcn plus Cam) have explainers which capture both part of the motif (grid or house), and part of the BA graph. This confirms the unreliability of the explanations extracted from the "default" class.

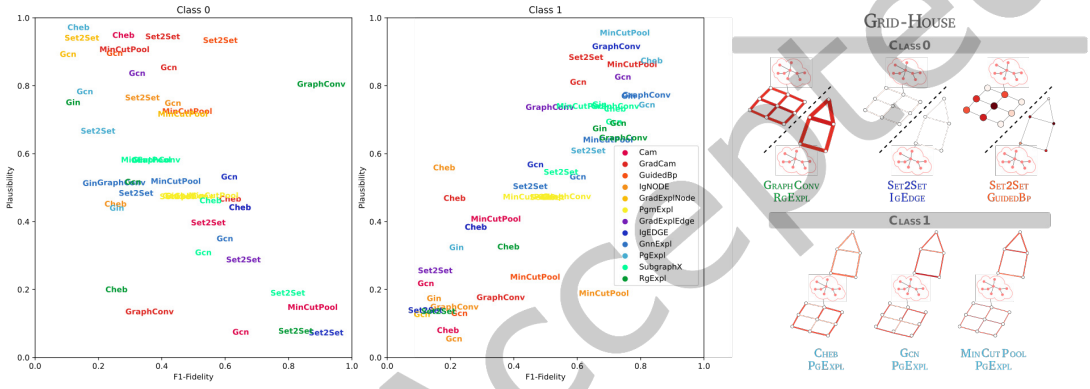

Fig. 5. Left: Fidelity and plausibility achieved by all the model-explainer pairs when applied to Grid-House, for class 0 (left) and class 1 (right). In each pair the name refers to the model, while the color identifies the explainer. Right: Stability of the explanations for the three top-performing model-explanation pairs, for each of the two classes. The colors identify important edges (dark red), and the edge thickness the variability of the importance in the dataset.

In terms of stability, right of Figure 5 shows the average explanations for the three top-performing explainers for each of the two classes. Regarding class 0, we can see that both GraphConv-RgExpl and Set2Set-GuidedBp identify both the grid and the house, and this reflects the high plausibility shown in the left panel in Figure 5. On the other hand, IgEdge does not capture neither the grid nor the house, but it captures part of the BA component, and this explains the low plausibility and high fidelity.

For class 1 the situation is more uniform, since the three optimal pairs all achieve a rather similar stability. Indeed, the two motifs are colored in dark red, and the edges are rather thin. A partial exception is the case of Cheb-GuidedBp, where the the house motif has a lighter color (and thus a smaller average importance across the dataset), and the grid has some variability over the edges' thickness, indicating a larger standard deviation in their importance.

Finally, in Figure 6 we show an example of the explanation provided by the best explainer associated to each model, for both class 0 and class 1. These explanations are visualized with the same color scale used for the previous dataset, and computed for a graph randomly selected from each class in Grid-House. The high plausibility pairs are clearly visible since they identify the house for class 0 (Cheb-PgExpl and MinCutPool-PgExpl) and both structures and their connection for class 1 (MinCutPool-PgExpl). Interestingly, even the high

fidelity ones are easy to spot (GradCam-Set2Set for class 0, Cheb-PgExpl for class 1), and this indicates a good agreement between the expected and learned concepts.

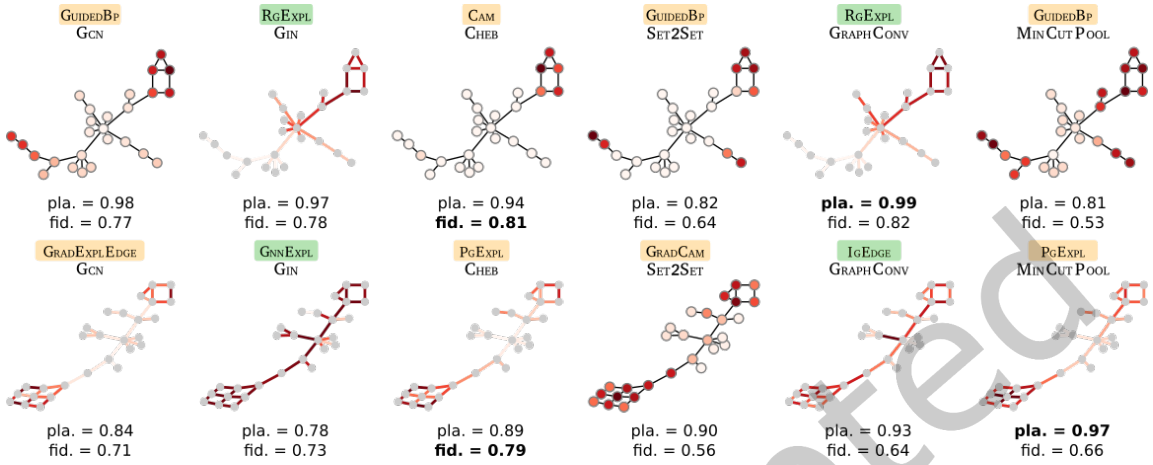

Fig. 6. Examples of explanations provided for each model and its highest plausibility explainer, when applied to a random sample from Grid-House. Each row shows the results for one of the two classes. The plausibility and fidelity values are those of the entire dataset, as reported in Figure 5.

Stars. This dataset has three classes: each graph is obtained starting from an Erdős–Rényi (ER) graph, to which we attach one (class 0), two (class 1), three or four (class 2) stars. This dataset thus evaluates GNN explainers on concepts involving counting.

In order to enable this motif-counting capability, we used for all architectures a sum-based global aggregation in place of a mean-based one, which would prevent the networks from being able to count occurrences of motifs. Indeed, mean-based versions of all models were trained as well, yet without reaching the thresholded train accuracy of 95% (see Section 6).

The behaviors of each model-explainer pair is visualized in Figure 7, with a panel for each class.

Class 0 is explained well by explainers producing node masks, and especially by GradExplNode on GraphConv. However, all the model-explanation pairs have a fairly limited plausibility when compared with the other datasets, and none of them has a plausibility larger than 0.8.

For class 1 the best combined performances are achieved by Gin explained by IgNode and Cam, even if also in this case the fidelity is almost always below 0.8, with instead a quite high plausibility. This indicates that these explainers may be good in identifying the expected ground truth (the stars), thus obtaining high plausibility, but that the presence of these motifs alone may be insufficient for predicting the class, when not complemented by a non-negligible part of the ER graph.

The right panel of Figure 7 shows the results for class 2. Also in this case the plausibility is very limited across the entire set of models and explanations. Looking at the fidelity alone, it is evident that the best performing explainers are those that produce an edge mask. In this case, a direct inspection of the explanations shows that these explainers not only capture the stars, but also paths connecting them. This makes the explanations farther away from the expected ground truth (thus obtaining low plausibility), but apparently provide better masks for the model to correctly identify the class.

From these results, it seems clear that there may be some clash between the identification of the motifs (high plausibility) and the fact that these motifs are sufficient to predict the class (low fidelity). To try to explain this

behavior, we evaluated all the trained networks on the exact expected ground truth explanations, i.e., four graphs obtained by one, two, three or four disconnected stars. For all these graphs, all the models predict class 2 (i.e., that with three or four stars attached to an ER graph), thus completely missing the correct classification for class 0 and class 1. These stars are thus clearly insufficient to characterize what the networks use for prediction. This result reveals the difficulty in defining the plausibility of an explanation, even in the presence of explicitly defined ground truths.

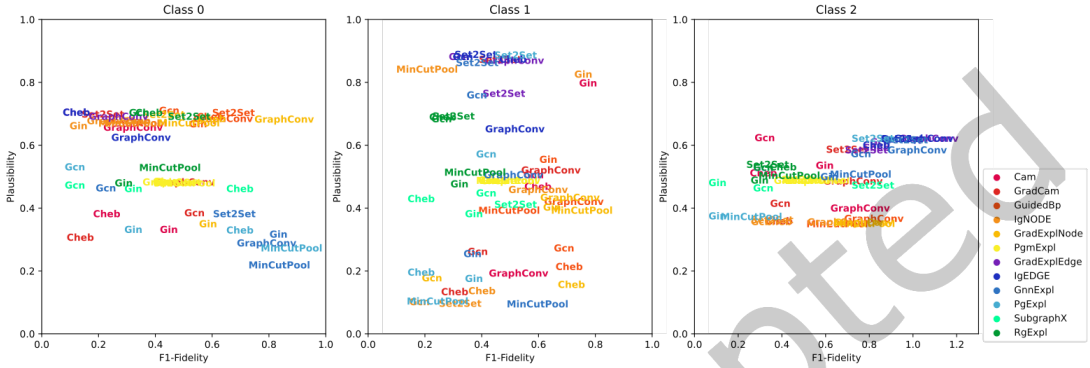

Fig. 7. Fidelity and plausibility achieved by all the model-explainer pairs when applied to Stars, for class 0 (left) and class 1 (right). In each pair the name refers to the model, while the color identifies the explainer.

Figure 8 shows the stability of the explanations when restricted to the stars which identify the three classes. We remark that both node- and edge-based explainers are visualized, and thus the coloring and thickness may apply to either the nodes or the edges.

In class 0, explainers that produce a node mask (IgNode on Set2Set, and GradExplNode on GraphConv) capture really well the star (intensity of the color) with a low standard deviation (size of the nodes), i.e., they are extremely stable. Remarkably, in the case of GraphConv the explainer gives no importance to the center of the star, while for Set2Set it does. On the other hand, GnnExpl on Gin identifies the star but with a low intensity, and this explains the corresponding low plausibility observed in Figure 7.

For class 1, all the three best model-explainer pairs capture the stars, even if Cam has a higher standard deviation. Surprisingly, Cam gives to the central node of the star always the highest importance (dark color and small size), while IgNode does the opposite (light color and large size). It is interesting to observe that both masks are reasonable ways to identify the two stars, and it may be difficult to recognize one or the other as the actual correct explanation. This difference is made even more interesting by the fact that the two explanations apply to the same model (Gin), and that the resulting fidelity and plausibility are essentially the same (see Figure 9, central panel).

Finally, for class 2 all explainers capture the stars with high importance and low standard deviation, without significant differences.

Figure 9 reports, for each model and each class, an example explanation provided by the corresponding best explainer. All masks are computed on a graph randomly drawn from the dataset for each class.

House-Color. In this dataset we would like to explore how the node features affect the explanations. In particular, both classes have a BA base graph with attached one, two, or three houses, and each node has a random color, except for one house that has all nodes colored blue (class 0) or green (class 1). In particular, the two classes represent essentially the same type of pattern, and thus we have no reason to expect that a lazy

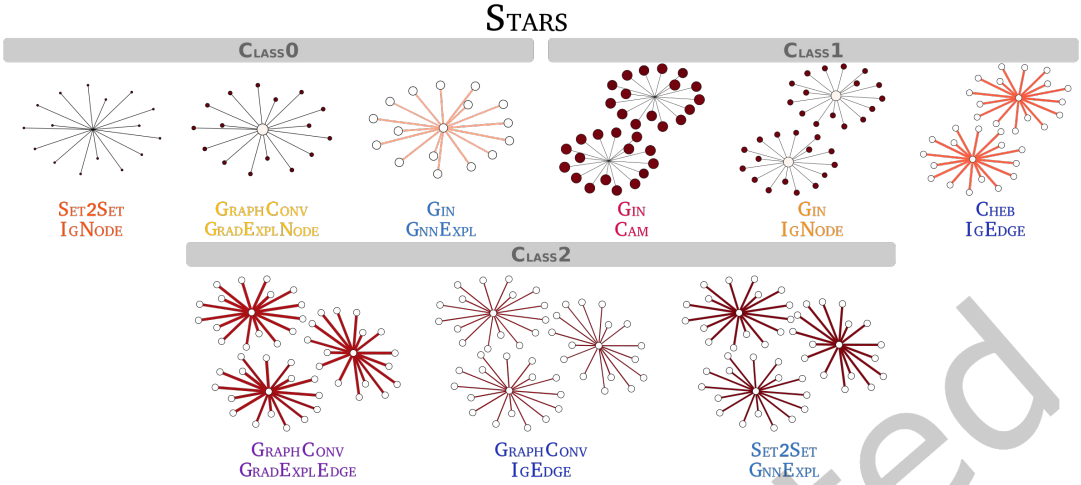

Fig. 8. Stability of the explanations for the three top-performing model-explanation pairs, for each of the three classes. The colors identify important edges (dark red), and the edge thickness the variability of the importance in the dataset.

GNN should prefer one or the other, and consequently no way to anticipate a difference in their explainability. By laziness, specifically, we mean the GNN focusing on learning the discriminant features of class 0 (1) while predicting the remaining class 1 (0) by the absence of such discriminant feature, without properly modeling the underlying discriminant feature of the latter class.

The comparison of fidelity and plausibility in shown in the left panel of Figure 10. To avoid visualizing too many results, the figure is limited to those pairs with both plausibility and fidelity greater than 0.5. Also in this case a linear correlation between the two metrics is clearly present, even if there is a group of outliers with high plausibility and low fidelity (the MINCUTPOOL models with explainers GRADEXPLNODE and GUIDEDBP, for the two classes). In this case, a closer inspection of the single explanations reveals that the explanation masks are able to capture well the colored house, thus achieving a large plausibility, but they contain also a significant amount of noise that spoils the fidelity. An example of this behavior can be observed in the examples in Figure 11. Moreover, for this dataset the optimal pairs are based on GRAPHCONV and SET2SET, with explanations GRADEXPLEDGE, IGEDGE, PGEXPL. In particular, edge-based explainers are the top performing ones also in this case. Another remarkable aspect is the fact that for each model-explainer pair, only one between class 0 and class 1 achieves high scores (i.e., both high fidelity and high accuracy). This suggests again a lazy behavior for the GNN even in the case in which both classes are equally easy to explain (Figure 10), where some GNNs are learning to characterize class 0 and others class 1, but none modelling both.

In the right panel of Figure 10 we report the average explanation for the three top-performing pairs, which all comprise edge-based explainers. Differently from the previous datasets which had no node feature, the figure visualizes also the color of the node feature close to each node. In all cases the explanation is very strong across the dataset (dark red color), with a minimal standard deviation for GRAPHCONV-GRADEXPLEDGE, and a maximal one for GRAPHCONV-IGEDGE.

Examples of explanation masks computed on random samples from HOUSE-COLOR are reported in Figure 11, where again we show for each class and for each model only the corresponding best performing explainer. Once again, explainers that produces edge features are more effective in terms of plausibility and fidelity. It is worth mentioning that there are no explainers having high fidelity in both classes, arguing the thesis that GNNs are lazy and learn only one class. On the other hand, at least one class for each GNN can be explained by an explainer. As

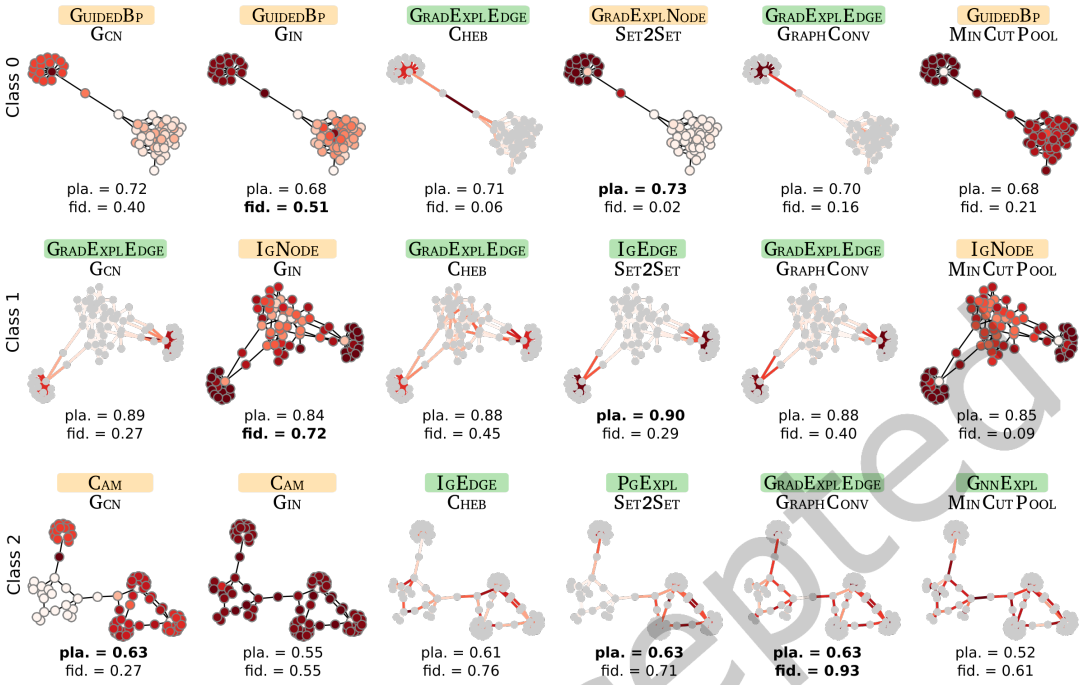

Fig. 9. Examples of explanations provided for each model and its highest plausibility explainer, when applied to a random sample from Grid-House. Each row shows the results for one of the two classes .The plausibility and fidelity values are those of the entire dataset, as reported in Figure 7.

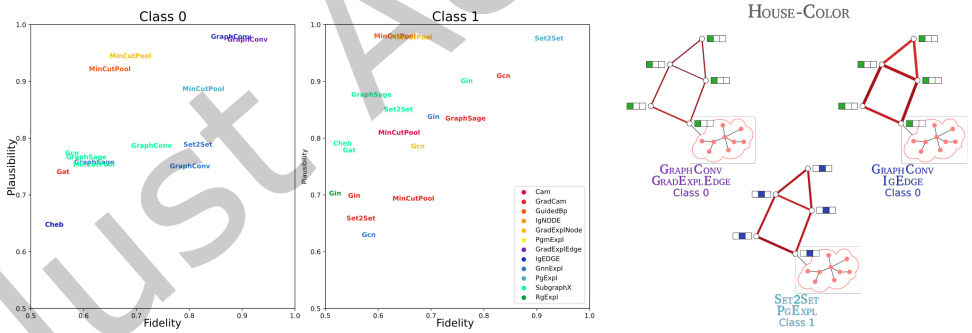

Fig. 10. Left: fidelity and plausibility achieved by all the model-explainer pairs when applied to House-Color, where the two classes (0 and 1) are represented in the left and right panel. In each pair the name refers to the model, while the color identifies the explainer. The plot is limited to metrics larger than 0.5 to simplify the visualization. Right: stability of the explanations for the three top-performing model-explanation pairs. The colors identify important edges (dark red), and the edge thickness the variability of the importance in the dataset.

one may expect, the only exception is the Gat architecture, because the attention mechanism strongly operate on node features, and none of the studied explainers operate on node features.

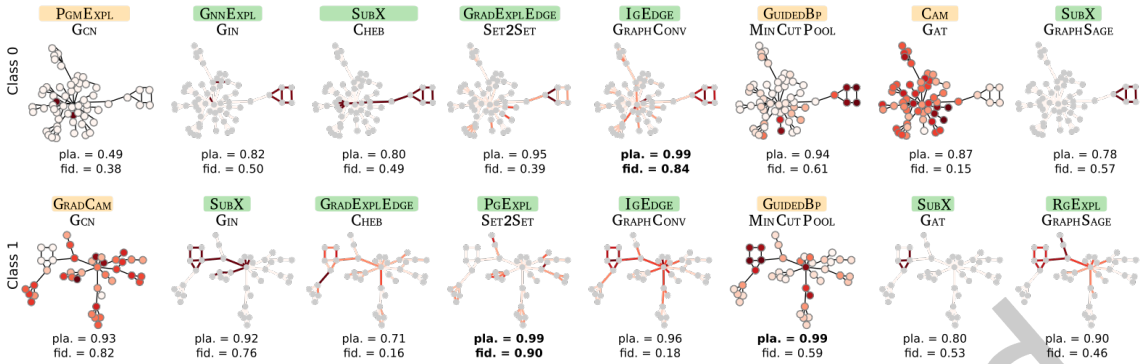

Fig. 11. Examples of explanations provided for each model and its highest plausibility explainer, when applied to a random sample from GRID-HOUSE. Each row shows the results for one of the two classes. The plausibility and fidelity values are those of the entire dataset, as reported in the right of Figure 10.

7.3 Node classification

7.3.1 RQ1: How does the architecture affect the explanations? To answer this question, we summarize the performance of the different architectures across all explainers by using the aggregation mechanism described in Section 5.2. The plausibility and fidelity results for each of the two datasets SHAPES and INFECTION, and the overall results are reported in Table 7.

The most explainable architectures are GRAPHSAGE for the plausibility metric (explained by GRADCAM), and GIN for fidelity (explained by GNNEXPL). They are clearly identified as such by **RQ1.1** and **RQ1.2**, i.e., both as single best performing architectures and mean best performing ones, and both at the aggregate and single-dataset levels. The difference between the two metrics indicates that for GRAPHSAGE one gets explanations that better resemble the expected ground truth, and are thus closer to a human-level explanation. On the other hand, for GIN one obtains explanations with a higher fidelity, meaning that they better capture the actual patterns that the trained GIN uses in building its decisions.

The hardest architecture to explain on average (**RQ1.3**) is identified to be CHEB, for both metrics and both at the aggregate and at dataset levels. The only exception is INFECTION with GRAPHSAGE.

	Plausibility			Fidelity		
	All	SHAPES	INFECTION	**All**	SHAPES	INFECTION
RQ1.1	**GRAPHSAGE**	GRAPHSAGE	GCN	**GIN**	GCN	GIN
RQ1.2	**GRAPHSAGE**	GRAPHSAGE	GRAPHSAGE	**GIN**	GCN/GIN	GIN
RQ1.3	**CHEB**	CHEB	CHEB	**CHEB**	GRAPHSAGE	CHEB

Table 7. Experimental answer to **RQ1** for node classification. The table shows the top-ranking architecture with respect to each subquestion **RQ1.1**, **RQ1.2**, **RQ1.3**, both for the single datasets SHAPES and GRID-HOUSE, and overall. The rankings are computed with respect to the plausibility and the fidelity metrics.

Figure 12 shows examples of explanation masks computed by the different explainers on GRAPHSAGE, which is the architecture with the highest average plausibility (RQ1.2), over all explainers which passed the filtering procedure (Section 5.2). For each dataset we visualize a sample node and its 2-hop neighborhood (directed in the case of INFECTION, i.e., the set of nodes from which the ego node is reachable following two edges). For both datasets we focused on nodes with label 1, i.e., the base of the roof in SHAPES, and a node at distance one or two from an infected node in INFECTION. This means that their associated ground truths are the entire house in

Shapes, and a path of length one or two in Infection (in this case multiple possible ground truth may exist, and only the one with the highest plausibility is considered for the computation of the metric). Despite GraphSage being the model with the highest plausibility, it is clear from these examples that there is a high variability across nodes and explainers. Moreover, for Shapes the house is well highlighted by GradCam, PgmExpl, GnnExpl, and RgExpl, but the explanation additionally includes spurious nodes and edges with equally large importance. A similar situation can be observed for Infection. In both cases, a finer inspection (see Section 7.3.3) reveals that the plausibility of GraphSage, despite being the highest on average over classes and explainers, is fairly limited for class 1: the values are below 0.6 for class 1 in Shapes (Figure 14), and with high spread and mean around 0.6 for class 1 in Infection (Figure 16). We remark that for both datasets, other classes achieve an higher plausibility.

In any case, this variability across the same dataset has to be taken into account, and a word of caution is in order when using instance-based explanations, since the single masks are those which are actually utilized to inspect the model and try to extract information on its decisions. Their usefulness may vary largely from node to node.

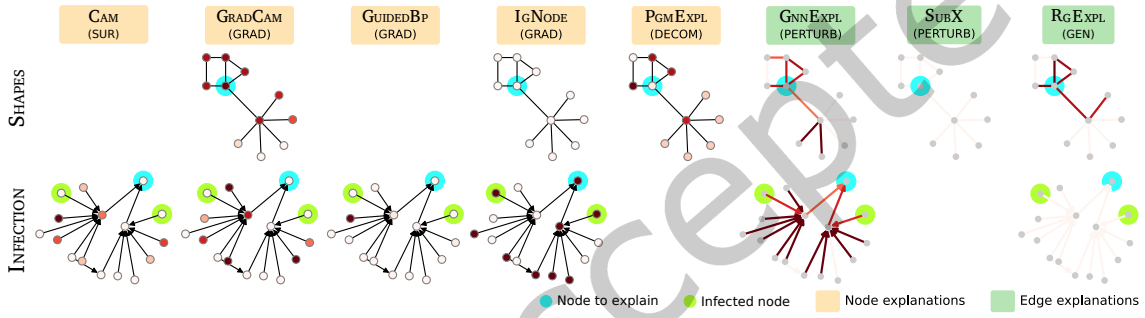

Fig. 12. Explanation masks (node- or edge-based) computed by the different explainers on the predictions of GraphSage on Shapes and Infection. Each row visualizes the mask computed for a given random graph from each dataset. For each dataset, only the explainers which passed the filtering procedure are shown.

7.3.2 RQ2: How do explainers affect the explanations? Table 8 shows that, while GradCam provides the single best explanation in terms of plausibility, GnnExpl is the best one both in terms of mean performances (**RQ2.2**) for both metrics, and maximal performances (**RQ2.1**) for plausibility, and can be thus considered as the overall best performing explainer in this setting. It is worth remarking that this is a perturbation- and edge-based explainer.

When looking at average performances for the entire category the situation is instead different, and it turns out that gradient- (**RQ2.3**) and node-based (**RQ2.4**) explainers are to be preferred, with a uniform consensus across metrics and aggregation levels. In particular, PgExpl (perturbation- and edge-based) has very poor performances and this lower the aggregates scores of the corresponding categories. This fact is possibly justifiable since we are dealing with node classification tasks, where local explanations like those provided by gradient based explainers may be more effective. Moreover, since meaningful explanation should be limited to a node's k-hop neighborhood, it is reasonable to expect that highlighting single nodes instead of edges is sufficient to explain the decision. In particular, the node itself is already defining the relevant neighborhood, and we may thus expect that the additional missing information required to elaborate a decision is given by the nodes' labels.

In the same setting of Figure 12, we report in Figure 13 examples of explanation masks obtained by GnnExpl, which has the highest average plausibility over all architectures and classes - RQ2.2. Also here, architectures not passing the filtering step have been removed (Section 5.2). In this case a better localization of the explanations

	Plausibility			Fidelity		
	All	Shapes	Infection	**All**	Shapes	Infection
RQ2.1	**GradCam**	GradCam	GradCam	**GnnExpl**	PgExpl	GradCam
RQ2.2	**GnnExpl**	GradCam	GnnExpl	**GnnExpl**	GnnExpl	GnnExpl
RQ2.3	**Grad**	Grad	Grad	**Grad**	Grad	Grad
RQ2.4	**Node**	Node	Node	**Node**	Node	Node

Table 8. Experimental answer to **RQ2** for node classification. The table shows the top-ranking explainer with respect to each subquestion **RQ2.1**, **RQ2.2**, **RQ2.3**, **RQ2.4**, both for the single datasets Shapes and Grid-House, and overall. The rankings are computed with respect to the plausibility and the fidelity metrics.

may be observed for Shapes for all models. Indeed, all architectures have a plausibility above 0.8 in class 1 in Shapes (Figure 14).

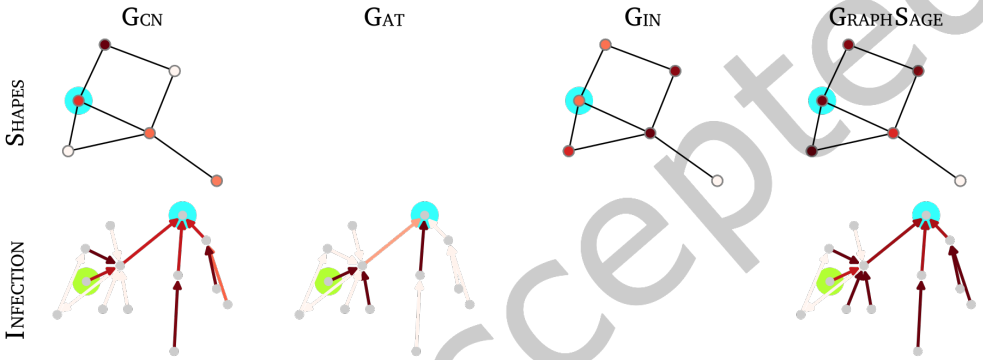

Fig. 13. Explanation masks (node- or edge-based) computed on the predictions of GnnExpl by the different explainers, both on Shapes and Infection. Each row visualizes the mask computed for a given random graph from each dataset. For each dataset, only the explainers which passed the filtering procedure are shown.

7.3.3 RQ3: How do different types of problems affect the explanations? We analyze the two datasets separately, in order to highlight the different aspects and problems that they represent.

Shapes. The comparative visualization of the plausibility and fidelity of each architecture-explainer pair is reported in Figure 14, where we visualize only the classes 1 to 3 (structural elements of a house, see Section 4.2), while we omit class 0, which having no structure is less interesting from the point of view of explanations.

No clear correlation emerges between the two metrics, except partially for class 2, which is the only one with architecture-explainer pairs with both high fidelity and high plausibility. Several pairs achieve very high plausibility (even close to 1, and for all three classes), but always with low fidelity (values bounded by $0.6 - 0.7$ depending on the class). The only outlier is the Gcn-PgExpl pair for class 2 and 3, which has a plausibility of about 0.9 and fidelity of 0.8 (class 2), and 0.7 (class 3).

The very high plausibility indicates that the explainers agree with the human definition of a ground truth, which is thus correctly defined to be the same for all classes (the house structure). On the other hand, this in turn results in an observed low fidelity. This seems to suggest that it is enough to have a few house elements missing from the explanation (the plausibility is never equal to 1) to spoil the model predictions, and that indeed the entire motif has to be observed to compute a prediction. It is possible moreover that an high plausibility is a proxy for a low sparsity of the explanation, which may thus miss some crucial edges. On the other hand, explanations

with lower plausibility may be more spread, thus covering the entire house structure, even if with some spurious additional edge. This result highlights again that the sufficiency alone may be non very informative in some settings.

Regarding the well-performing Gcn-PgExpl pair, a sort of two-class laziness can be observed: the explanations are good for class 2 and 3, while they have fairly limited plausibility and fidelity (around 0.5) for class 1. This may suggest that the nodes in class 1 (the middle of the house) are classified by Gcn as not being in any other class. Examples in Figure 15 shows explanation masks returned for each model by its explainer which reaches the highest plausibility. A significant overlapping between the explanation mask and the house structure (the ground truth) can be observed, even if in most cases the mask is at least partially spread also over other nodes.

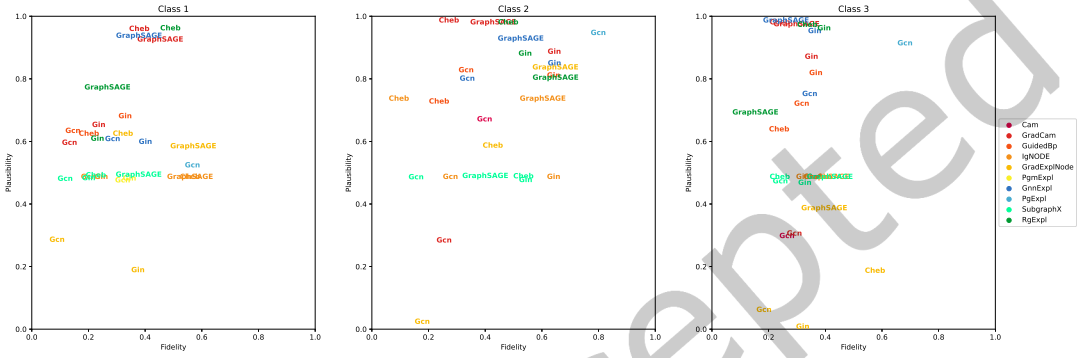

Fig. 14. Fidelity and plausibility achieved by all the model-explainer pairs when applied to Shapes. In each pair the name refers to the model, while the color identifies the explainer. Class 0 is omitted from the visualization since it is less relevant for the discussion of the explainers.

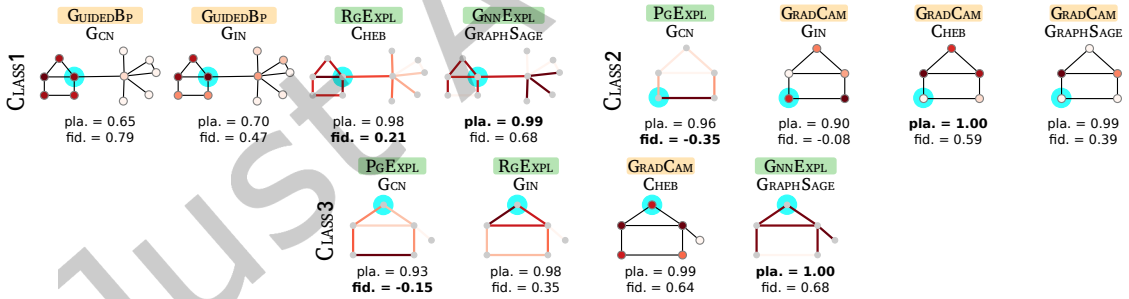

Fig. 15. Examples of explanations provided for each model by its highest plausibility explainer, when applied to a random node from Shapes. Each row shows the results for one of the three house-structure classes. The plausibility and fidelity values are those of the entire dataset, as reported in Figure 14.

Infection. The pairs of models and explanations are shown in Figure 16 according to their plausibility and fidelity on the three classes. We first remark that, out of the two edge-based explainers considered here, only GnnExpl passes the filtering step (Section 5.2), more specifically GnnExpl with Gcn (for class 1,2) and with Gat and GraphSage (for all classes). Also in this case no correlation appears among the two metrics, while there are several pairs that reach a high value of only one of the two metrics. It is again possible that even relatively

high values of the plausibility comprise explanations where some relevant part of the ground truth is missing, thus achieving low fidelity. For this dataset the necessity of covering the entire motif is even more clear than for Shapes, since here single nodes missing from an explanation may break the minimal paths connecting the node to an infected one. Exceptions are GraphSage-GradCam and GraphSage-IgNode for class 1, and especially Gcn-GradExplNode for class 2, which have both metrics above values of 0.8.

With the exception of Gcn, class 2 exhibits a clear lack of explainability, indicating a preference of GNN architectures in focusing on class 0 and 1. This laziness is perfectly sensible, since it is much easier to model class 2 in terms of a lack of paths from nearby infected nodes (i.e., the negation of the other classes) than by trying to characterize all possible longer paths from an infected node. The exception to this pattern is Gcn-GradExplNode, which has very high scores for class 2, but is instead significantly unexplainable for class 1 and class 0, where it is below the threshold for visualization. This is reasonable since Gcn has a simple message passing - aggregation mechanism, which may facilitate the identification of lack of infected nodes in the 2-hop neighborhood (since the infected status is the node feature itself, see Section 4.2), even if GraphSage has a completely analogous functioning mechanism, and it achieves nevertheless poorer results. At this stage, we are not able to identify the actual mechanism that make their performances so different.

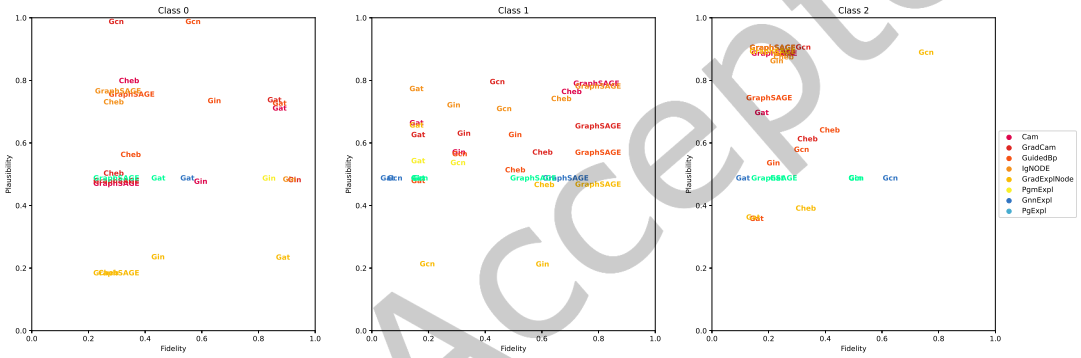

Fig. 16. Fidelity and plausibility achieved by all the model-explainer pairs when applied to Infection. In each pair the name refers to the model, while the color identifies the explainer.

Example of explanations are shown in Figure 17 for each model paired with the explainer which explains it with the highest plausibility. We show only class 1, since class 0 is trivial to visualize (the explanation is the actual node), while for class 2 the local networks happen to have too many edges and their visualization is not clear enough. We thus omit both since their visualization does not provide significant insights. In this case, it is difficult to observe a good accordance between the mask and the ground truth (a path connecting an infected node with the node to be explained). This happens despite the relatively high plausibilities (Figure 16), and demonstrates again that care should be taken in considering single-instance explanations for the interpretation of a model's prediction.

In Table 9, we present a comprehensive summary of the principal outcomes derived from our research. Each row within the table is dedicated to addressing a specific research question.

8 Discussion

The main objective of this work is to experimentally study the effectiveness of explainers on different GNNs and types of data, identifying current pitfalls and formulating possible future directions in the field of GNN explainability. Given that GNNs may learn different concepts, possibly less intuitive, than those expected by

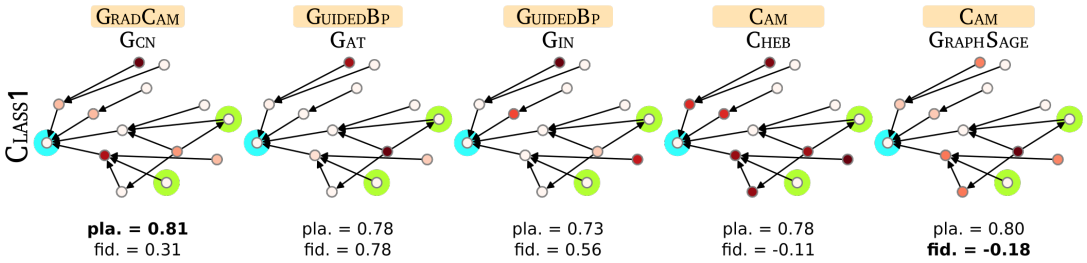

Fig. 17. Examples of explanations provided for each model by its highest plausibility explainer, when applied to a random node from Infection. Class 0 and class 2 are omitted since their visualization does not provide significant insights. The plausibility and fidelity values are those of the entire dataset, as reported in Figure 16.

humans, defining ground truths is not a trivial task and may be prone to human biases. A first simple remark is that GNNs, as neural networks in general, tend to be lazy and learn a "default" option for one of the classes. For this reason, a high accuracy does not necessarily imply having learned the ground-truth concept for the class. A useful insight that emerged from our analysis is that these human biases can often be detected by comparing plausibility and fidelity. Indeed, an explainer with high fidelity and low plausibility (or vice-versa) clearly indicates a discrepancy between what is considered to be the ground truth and the concept learned by the GNN. It is important to remark, however, that fidelity alone may not be the optimal choice for both graph and node classification. In the case of graph classification, the aggregation used to convert nodes embedding into a graph embedding directly influences the fidelity. In particular, sum aggregation, which is needed to allow GNN to count substructures, often negatively affects fidelity, with the mere ground-truth structure achieving relatively low fidelity because of the reduced number of nodes which in turn reduces the norm of the overall embedding.

The aggregation mechanism is a crucial component of GNNs and its decision directly affects the quality of the explanation. Nonetheless, there is a lack of works studying the impact of the aggregation mechanism on explainability. This is an interesting direction for further research. Reliably measuring fidelity can be tricky for node classification too. On the one hand, comprehensiveness is poorly defined when explaining node predictions (see Section 5). On the other hand, measuring fidelity only in terms of sufficiency introduces a bias that favours larger explanations. Indeed, finding the optimal metric for evaluating explainers is still an open problem that deserves further investigation.

Identifying a general category of explainers working consistently better than others is challenging. However, our results suggest that gradient-based explainers are more suited in explaining node classification networks. We conjecture that this is due to the fact that in node classification, the gradient is computed only on the neighborhood of the node under investigation, limiting the receptive field of the network. On the other hand, the category of explainers that best explain GNNs for graph classification are those that focus on edges, be it by perturbation or gradient. In general, edge-based explainers outperform node-based ones whenever node features are not available.

Concerning GNN architectures, there is a substantial difference in their explainability, regardless of the explainer that best suits each of them. We believe that this result is surprising yet not fully understood, given that explainers are usually aimed to be model agnostic. Given the importance of explaining predictions, it would be advisable to include explainability as a metric to be optimized when designing novel GNN architectures. Finally, we would like to highlight that this work did not consider the most recent and complex architectures, like diffusion-based [4, 29, 92] and spatio-temporal models [62, 68, 84], which are typically omitted from the evaluation of explainers in the literature. We leave an investigation on the usability of existing explainers for

	Graph classification	Node classification
RQ1: How does the GNN architecture affect the explanations?		
RQ1.1	GraphConv excels in both plausibility and fidelity, achieving the highest aggregated scores.	GraphSage provides explanations closer to the expected ground truth, akin to human-level explanations, while Gin attains elevated fidelity by effectively encapsulating the patterns acquired by the Gin model during its training.
RQ1.2	Gcn is the easiest architecture to explain, possibly due to its straightforward message-passing model.	
RQ1.3	Gin is the hardest to explain, which is surprising given its rather simple aggregation strategy.	Cheb is the hardest to explain due to its broader receptive field, aggregating information from more distant nodes compared to other networks.
RQ2: How do explainers affect the explanations?		
RQ2.1	The best explainer differs if evaluated in terms of plausibility or fidelity, but the two best performing ones are both edge- and gradient-based.	The best explainer differs if evaluated in terms of plausibility (GradCam) or fidelity (GnnExpl).
RQ2.2	SubX explains the maximal number of models.	GnnExpl explains the maximal number of models.
RQ2.3	Perturbation methods excel in plausibility, while generative methods perform best in fidelity, but may not capture the expected ground truth.	Gradient methods excel in both plausibility and fidelity.
RQ2.4	Edge-based explainers outperform node-based ones, possibly because they are designed for graph-explanation tasks, while node-based explainers are adapted from other contexts.	Node-based explainers outperform edge-based. The motivation remains unclear, necessitating future investigation.
RQ3: How do different types of problems affect the explanations?		
Motif-based explanations	In learning problems where the underlying ground truth is a motif, care should be taken in designing the dataset. In some popular scenarios, the ground-truth motif contains a smaller minimal discriminant subgraph (MDS). For example, in Grid, the expected explanation is a 3x3 grid, while the MDS is a simple square. This clearly undermines a correct evaluation of the explanations being extracted. The problem can also affect the evaluation of real-world explanations by domain experts.	
Class-specific explanations	When considering explanations for both positive and negative predictions, the phenomenon of laziness severely affects the quality of the explanation being extracted. GNNs typically tend to learn features for only one of the two classes, and predict the other as a "default" option. Extracting explanations for the "default" class is useless and potentially misleading.	
Aggregation-dependent explanations	The sum aggregator, used in tasks like counting substructures (e.g., Stars), makes the graph-level embedding dependent on the number of nodes in the graph. However, in fidelity evaluation, explanations involve smaller subgraphs, potentially introducing a distribution shift that harms model fidelity, as also previously noted in [131].	

Table 9. Summary of the main lessons learned, divided by research question.

these architectures, and on the need for ad-hoc explainers specifically designed for their characteristics, to future research.

9 Future directions

Several explainers have been proposed and tested for the classical GCN architecture [49], but we argue that significant work has yet to be developed to fully grasp the complex interaction of models and explanations and their interpretability. In the following we outline some promising research directions we believe deserve further investigation.

9.1 Going beyond instance-level explanations

While many studies focus on instance-level explanations, model-level explanations are less explored [5, 70, 71, 109, 117, 119]. A deeper understanding of global explanations could provide a more comprehensive insight into the inner workings of GNNs. This approach can reveal how features are aggregated and utilized across the model, offering a broader perspective that complements instance-level insights. Apart from extracting explanations for providing human-aware guidance on the behavior of a trained GNN, several works also use explanations during the training of the model as a regularization term. It has been shown, in fact, that explanation-aware training of GNNs can result in better generalization, faster convergence, and intrinsically higher explainability [33, 96, 99]. Taking this design choice further, a recent trend is to have the GNN make predictions based solely on its explanations. In this framework, the GNN is typically divided into two components: first an explanation extractor takes as input the entire graph and outputs the explanation, then a classifier takes as input the explanation and makes the final prediction. Explanations can come in different forms, like subgraphs [67, 93], activation values for semantic concepts or prototypes similarity [77, 130]. Following this schema, if the model is faithful to the extracted explanation then the explanation unambiguously depicts the underlying reason for the model's predictions. Unfortunately, recent studies have shown that despite this training-aware conditioning the resulting models are often unfaithful to the explanations, meaning that it does not fully depict the intended model behavior [15], an issue shared with classic post-hoc explainers. Therefore, further research is needed in the study of trustworthy and effective explainers, whether they are used at training or evaluation time.

9.2 What is an explanation

A direction that warrants further investigation is the definition of explanations in GNNs. Current explainability-related studies typically consider a subgraph as an explanation. However, we argue that in some real-world applications, this subgraph may not provide a sufficiently faithful explanation. For example, in a recommendation system, a user might purchase an item not because of its inherent features, but because it is currently trending. In this case, the explanation should highlight the high degree of the item's node, rather than its enclosing subgraph. An additional example is when local predictions are influenced by a global property of the network. In traffic flow prediction, for instance, the congestion of a particular street may be due to its role as the sole connection between two parts of the city. This situation can be explained by the high betweenness centrality of that street, indicating its critical position in the network. Another intriguing direction involves investigating the causality of explanations [59, 60, 110]. This entails understanding the minimal modifications required to the computational graph of a node to alter its prediction. Such causal explanations can provide deeper insights into what the model has learned to make predictions and how node features are aggregated and utilized by the model.

9.3 The role of the dataset

We used only synthetic datasets in our experiments because they provide the ground truth explanations necessary for our evaluations. However, despite the predefined ground truth, some well-known benchmark datasets have design issues [24]. Therefore, it is crucial to reconsider these synthetic benchmarks, paying close attention to potential shortcuts and ensuring that minimal explanations are provided. Furthermore, as discussed in Subsection 9.2, it is essential to define synthetic datasets that match the expected ground truth in real-world scenarios, where the explanation may differ from a local structure.

10 Conclusion

In this survey we proposed an extensive experimental study to quantify the effectiveness of the existing explainers and to obtain actionable recommendations to select the optimal method for a given task. For this comparison we evaluated ten explainers on eight different GNN architectures, all chosen to represent the most commonly utilized

instances in a vast taxonomy of existing solutions. These methods have been tested on six different datasets for both graph and node classification, carefully designed or adapted to model interesting and challenging aspects of real-world datasets. As a result of our experimental study, we were able to describe significant criticalities in the common explainer evaluation methods and to identify recurring patterns that make some categories of explainers preferable in certain situations. In particular, we proposed insights on which explainer to use, and how to use it, depending on the available data. Our findings naturally point to promising future research directions, and especially highlight once more that much has yet to be understood to achieve a satisfactory GNN explainability.

11 Acknowledgments

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor the granting authority can be held responsible for them. AL, BL and AP acknowledge the support of the MUR PNRR project FAIR - Future AI Research (PE00000013) funded by the NextGenerationEU. BL and AP also acknowledge the support of the Horizon Europe Programme, grant number 101120237 - ELIAS and grant number 101120763 - TANGO. GS acknowledges that this work took place within the framework of the DoE 2023-2027 (MUR, AIS.DIP.ECCELLENZA2023_27.FF project). GC acknowledges the support of the European Union's Horizon research and innovation program under the Marie Skłodowska-Curie grant agreement No 101103026.

References

[1] Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. 2022. Evaluating Explainability for Graph Neural Networks. arXiv preprint arXiv:2208.09339 (2022).

[2] Chirag Agarwal, Marinka Zitnik, and Himabindu Lakkaraju. 2022. Probing gnn explainers: A rigorous theoretical and empirical analysis of gnn explanation methods. In International Conference on Artificial Intelligence and Statistics. PMLR, 8969–8996.

[3] Paolo Andreini, Giorgio Ciano, Simone Bonechi, Caterina Graziani, Veronica Lachi, Alessandro Mecocci, Andrea Sodi, Franco Scarselli, and Monica Bianchini. 2021. A two-stage GAN for high-resolution retinal image generation and segmentation. Electronics 11, 1 (2021), 60.

[4] James Atwood and Don Towsley. 2016. Diffusion-Convolutional Neural Networks. In Advances in Neural Information Processing Systems, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2016/file/390e982518a50e280d8e2b535462ec1f-Paper.pdf

[5] Steve Azzolin, Antonio Longa, Pietro Barbiero, Pietro Lio, and Andrea Passerini. 2022. Global Explainability of GNNs via Logic Combination of Learned Concepts. In The Eleventh International Conference on Learning Representations.

[6] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. The Journal of Machine Learning Research 11 (2010), 1803–1831.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014).

[8] Federico Baldassarre and Hossein Azizpour. 2019. Explainability techniques for graph convolutional networks. arXiv preprint arXiv:1905.13686 (2019).

[9] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. science 286, 5439 (1999), 509–512.

[10] Sourya Basu, Jose Gallego-Posada, Francesco Viganò, James Rowbottom, and Taco Cohen. 2022. Equivariant Mesh Attention Networks. arXiv preprint arXiv:2205.10662 (2022).

[11] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. 2020. Spectral clustering with graph neural networks for graph pooling. In International Conference on Machine Learning. PMLR, 874–883.

[12] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. 2005. Protein function prediction via graph kernels. Bioinformatics 21, suppl_1 (2005), i47–i56.

[13] Cătălina Cangea, Petar Veličković, Nikola Jovanović, Thomas Kipf, and Pietro Liò. 2018. Towards sparse hierarchical graph classifiers. arXiv preprint arXiv:1811.01287 (2018).

[14] Quentin Cappart, Didier Chételat, Elias B. Khalil, Andrea Lodi, Christopher Morris, and Petar Veličković. 2021. Combinatorial Optimization and Reasoning with Graph Neural Networks. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4348–4355. https://doi.org/10.24963/ijcai.2021/595 Survey Track.

[15] Marc Christiansen, Lea Villadsen, Zhiqiang Zhong, Stefano Teso, and Davide Mottin. 2023. How Faithful are Self-Explainable GNNs?. In The Second Learning on Graphs Conference.
[16] Dan C Cireşan, Ueli Meier, Jonathan Masci, Luca M Gambardella, and Jürgen Schmidhuber. 2011. High-performance neural networks for visual object classification. arXiv preprint arXiv:1102.0183 (2011).
[17] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020).
[18] Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Jun Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. 2022. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. ArXiv abs/2204.08570 (2022).
[19] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems 29 (2016).
[20] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. ERASER: A benchmark to evaluate rationalized NLP models. arXiv preprint arXiv:1911.03429 (2019).
[21] Alexandre Duval and Fragkiskos Malliaros. 2022. Higher-order Clustering and Pooling for Graph Neural Networks. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 426–435.
[22] Vijay Prakash Dwivedi and Xavier Bresson. 2020. A generalization of transformer networks to graphs. arXiv preprint arXiv:2012.09699 (2020).
[23] Paul Erdős and Alfréd Rényi. 1959. On random graphs I. Publicationes mathematicae 6, 1 (1959), 290–297.
[24] Lukas Faber, Amin K. Moghaddam, and Roger Wattenhofer. 2021. When comparing to ground truth is wrong: On evaluating gnn explanation methods. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 332–341.
[25] Xiang Fu, Tian Xie, Nathan J Rebello, Bradley D Olsen, and Tommi Jaakkola. 2022. Simulate Time-integrated Coarse-grained Molecular Dynamics with Geometric Machine Learning. arXiv preprint arXiv:2204.10348 (2022).
[26] Thorben Funke, Megha Khosla, and Avishek Anand. 2021. Hard Masking for Explaining Graph Neural Networks. https://openreview.net/forum?id=uDN8pRAdsoC
[27] Thorben Funke, Megha Khosla, Mandeep Rathee, and Avishek Anand. 2022. Z ORRO: Valid, Sparse, and Stable Explanations in Graph Neural Networks. IEEE Transactions on Knowledge and Data Engineering (2022).
[28] Jianliang Gao, Jun Gao, Xiaoting Ying, Mingming Lu, and Jianxin Wang. 2021. Higher-order interaction goes neural: a substructure assembling graph attention network for graph classification. IEEE Transactions on Knowledge and Data Engineering (2021).
[29] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. 2022. Diffusion Improves Graph Learning. arXiv:1911.05485 [cs.SI]
[30] Dobrik Georgiev and Pietro Liò. 2020. Neural Bipartite Matching. https://doi.org/10.48550/ARXIV.2005.11304
[31] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In Proceedings of the 34th International Conference on Machine Learning - Volume 70. 1263–1272.
[32] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In International conference on machine learning. PMLR, 1263–1272.
[33] Valentina Giunchiglia, Chirag Varun Shukla, Guadalupe Gonzalez, and Chirag Agarwal. 2022. Towards training GNNs using explanation directed message passing. In Learning on Graphs Conference. PMLR, 28–1.
[34] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 855–864.
[35] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. ACM computing surveys (CSUR) 51, 5 (2018), 1–42.
[36] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. Advances in neural information processing systems 30 (2017).
[37] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. 2011. Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis 30, 2 (2011), 129–150.
[38] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. Neural Networks. Springer New York, New York, NY, 389–416. https://doi.org/10.1007/978-0-387-84858-7_11
[39] David Haussler et al. 1999. Convolution kernels on discrete structures. Technical Report. Technical report, Department of Computer Science, University of California ….
[40] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural Comput. 9, 8 (nov 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
[41] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. Advances in neural information processing systems 33 (2020), 22118–22133.
[42] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. 2022. Graphlime: Local interpretable model explanations for graph neural networks. IEEE Transactions on Knowledge and Data Engineering (2022).
[43] Priyank Jaini, Lars Holdijk, and Max Welling. 2021. Learning Equivariant Energy Based Models with Equivariant Stein Variational Gradient Descent. Advances in Neural Information Processing Systems 34 (2021), 16727–16737.

[44] K Sparck Jones and Cornelis Joost Van Rijsbergen. 1976. Information retrieval test collections. Journal of documentation (1976).
[45] Jaykumar Kakkad, Jaspal Jannu, Kartik Sharma, Charu Aggarwal, and Sourav Medya. 2023. A Survey on Explainability of Graph Neural Networks. arXiv preprint arXiv:2306.01958 (2023).
[46] Arash Keshavarzi Arshadi, Milad Salem, Arash Firouzbakht, and Jiann Yuan. 2022. MolData, a molecular benchmark for disease and target based machine learning. Journal of Cheminformatics 14 (03 2022). https://doi.org/10.1186/s13321-022-00590-y
[47] Megha Khosla. 2022. Privacy and Transparency in Graph Machine Learning: A Unified Perspective. arXiv preprint arXiv:2207.10896 (2022).
[48] Jinwoo Kim, Tien Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. 2022. Pure transformers are powerful graph learners. arXiv preprint arXiv:2207.02505 (2022).
[49] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016).
[50] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. Commun. ACM 60, 6 (2017), 84–90.
[51] Harold William Kuhn and Albert William Tucker (Eds.). 1953. Contributions to the Theory of Games (AM-28), Volume II. Princeton University Press, Princeton. https://doi.org/doi:10.1515/9781400881970
[52] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. In International conference on machine learning. PMLR, 3734–3743.
[53] Bin Li, Yunlong Fan, Yikemaiti Sataer, Zhiqiang Gao, and Yaocheng Gui. 2022. Improving Semantic Dependency Parsing with Higher-Order Information Encoded by Graph Neural Networks. Applied Sciences 12, 8 (2022), 4089.
[54] Jianxin Li, Hao Peng, Yuwei Cao, Yingtong Dou, Hekai Zhang, Philip Yu, and Lifang He. 2021. Higher-order attribute-enhancing heterogeneous graph neural networks. IEEE Transactions on Knowledge and Data Engineering (2021).
[55] Peibo Li, Yixing Yang, Maurice Pagnucco, and Yang Song. 2022. Explainability in Graph Neural Networks: An Experimental Survey. arXiv preprint arXiv:2203.09258 (2022).
[56] Wei Li, Ruxuan Li, Yuzhe Ma, Siu On Chan, David Pan, and Bei Yu. 2022. Rethinking Graph Neural Networks for the Graph Coloring Problem. https://doi.org/10.48550/ARXIV.2208.06975
[57] Wenqian Li, Yinchuan Li, Zhigang Li, Jianye Hao, and Yan Pang. 2023. DAG Matters! GFlowNets Enhanced Explainer For Graph Neural Networks. arXiv:2303.02448 [cs.LG]
[58] Yiqiao Li, Jianlong Zhou, Sunny Verma, and Fang Chen. 2022. A survey of explainable graph neural networks: Taxonomy and evaluation metrics. arXiv preprint arXiv:2207.12599 (2022).
[59] Wanyu Lin, Hao Lan, and Baochun Li. 2021. Generative causal explanations for graph neural networks. In International Conference on Machine Learning. PMLR, 6666–6679.
[60] Wanyu Lin, Hao Lan, Hao Wang, and Baochun Li. 2022. Orphicx: A causality-inspired latent variable model for interpreting graph neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 13729–13738.
[61] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. 2021. A survey of visual transformers. arXiv preprint arXiv:2111.06091 (2021).
[62] Antonio Longa, Veronica Lachi, Gabriele Santin, Monica Bianchini, Bruno Lepri, Pietro Lio, franco scarselli, and Andrea Passerini. 2023. Graph Neural Networks for Temporal Graphs: State of the Art, Open Challenges, and Opportunities. Transactions on Machine Learning Research (2023). https://openreview.net/forum?id=pHCdMat0gI
[63] Ana Lucic, Maartje A. Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. 2022. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. In Proceedings of The 25th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 151), Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (Eds.). PMLR, 4499–4511. https://proceedings.mlr.press/v151/lucic22a.html
[64] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. Advances in neural information processing systems 33 (2020), 19620–19631.
[65] Ge Lv and Lei Chen. 2023. On Data-Aware Global Explainability of Graph Neural Networks. Proc. VLDB Endow. 16, 11 (aug 2023), 3447–3460. https://doi.org/10.14778/3611479.3611538
[66] Ge Lv, Lei Chen, and Caleb Chen Cao. 2022. On Glocal Explainability of Graph Neural Networks. In Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part I. Springer-Verlag, Berlin, Heidelberg, 648–664. https://doi.org/10.1007/978-3-031-00123-9_52
[67] Siqi Miao, Mia Liu, and Pan Li. 2022. Interpretable and generalizable graph learning via stochastic attention mechanism. In International Conference on Machine Learning. PMLR, 15524–15543.
[68] Alessio Micheli and Domenico Tortorella. 2022. Discrete-time dynamic graph echo state networks. Neurocomputing 496 (2022), 85–95. https://doi.org/10.1016/j.neucom.2022.05.001
[69] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In Proceedings of the AAAI conference on artificial intelligence,

Vol. 33. 4602–4609. Issue 01.
[70] Peter Müller, Lukas Faber, Karolis Martinkus, and Roger Wattenhofer. 2024. GraphChef: Decision-Tree Recipes to Explain Graph Neural Networks. In The Twelfth International Conference on Learning Representations. https://openreview.net/forum?id=IjMUGuUmBI
[71] Yi Nian, Yurui Chang, Wei Jin, and Lu Lin. 2024. Globally Interpretable Graph Learning via Distribution Matching. In Proceedings of the ACM on Web Conference 2024. 992–1002.
[72] Sabyasachi Patra and Anjali Mohapatra. 2020. Review of tools and algorithms for network motif discovery in biological networks. IET systems biology 14, 4 (2020), 171–189.
[73] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. 2019. Explainability methods for graph convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 10772–10781.
[74] Marcelo Prates, Pedro H. C. Avelar, Henrique Lemos, Luis C. Lamb, and Moshe Y. Vardi. 2019. Learning to Solve NP-Complete Problems: A Graph Neural Network for Decision TSP. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. Article 581, 8 pages.
[75] Omri Puny, Matan Atzmon, Heli Ben-Hamu, Edward J Smith, Ishan Misra, Aditya Grover, and Yaron Lipman. 2021. Frame averaging for invariant and equivariant network design. arXiv preprint arXiv:2110.03336 (2021).
[76] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. 21, 140 (2020), 1–67.
[77] Alessio Ragno, Biagio La Rosa, and Roberto Capobianco. 2022. Prototype-based interpretable graph neural networks. IEEE Transactions on Artificial Intelligence 5, 4 (2022), 1486–1495.
[78] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022).
[79] Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. 2022. Recipe for a General, Powerful, Scalable Graph Transformer. arXiv preprint arXiv:2205.12454 (2022).
[80] Mandeep Rathee, Thorben Funke, Avishek Anand, and Megha Khosla. 2022. BAGEL: A Benchmark for Assessing Graph Neural Network Explanations. arXiv preprint arXiv:2206.13983 (2022).
[81] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 1135–1144.
[82] Gabriel Roccabruna, Steve Azzolin, and Giuseppe Riccardi. 2022. Multi-source Multi-domain Sentiment Analysis with BERT-based Models. In Proceedings of the Thirteenth Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, 581–589. https://aclanthology.org/2022.lrec-1.62
[83] T Mitchell Roddenberry and Santiago Segarra. 2019. HodgeNet: Graph neural networks for edge data. In 2019 53rd Asilomar Conference on Signals, Systems, and Computers. IEEE, 220–224.
[84] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal Graph Networks for Deep Learning on Dynamic Graphs. In ICML 2020 Workshop on Graph Representation Learning.
[85] Benjamin Sanchez-Lengeling, Jennifer Wei, Brian Lee, Emily Reif, Peter Wang, Wesley Qian, Kevin McCloskey, Lucy Colwell, and Alexander Wiltschko. 2020. Evaluating attribution for graph neural networks. Advances in neural information processing systems 33 (2020), 5898–5910.
[86] Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. 2020. Interpreting graph neural networks for nlp with differentiable edge masking. arXiv preprint arXiv:2010.00577 (2020).
[87] Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T Schutt, Klaus-Robert Müller, and Grégoire Montavon. 2021. Higher-order explanations of graph neural networks via relevant walks. IEEE transactions on pattern analysis and machine intelligence (2021).
[88] Robert Schwarzenberg, Marc Hübner, David Harbecke, Christoph Alt, and Leonhard Hennig. 2019. Layerwise relevance visualization in convolutional text graph classifiers. arXiv preprint arXiv:1909.10911 (2019).
[89] Daniel Selsam and Nikolaj Bjørner. 2019. Guiding High-Performance SAT Solvers with Unsat-Core Predictions. In Theory and Applications of Satisfiability Testing – SAT 2019, Mikoláš Janota and Inês Lynce (Eds.). Springer International Publishing, Cham, 336–353.
[90] Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, and David L. Dill. 2019. Learning a SAT Solver from Single-Bit Supervision. In International Conference on Learning Representations. https://openreview.net/forum?id=HJMC_iA5tm
[91] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision. 618–626.
[92] K Tamil Selvi, R Thamilselvan, and S Mohana Saranya. 2021. Diffusion convolution recurrent neural network – a comprehensive survey. IOP Conference Series: Materials Science and Engineering 1055, 1 (feb 2021), 012119. https://doi.org/10.1088/1757-899X/1055/1/012119

[93] Giuseppe Serra and Mathias Niepert. 2024. L2XGNN: Learning to Explain Graph Neural Networks. arXiv:2209.14402 [cs.LG] https://arxiv.org/abs/2209.14402
[94] Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. 2021. Reinforcement Learning Enhanced Explainer for Graph Neural Networks. In Advances in Neural Information Processing Systems, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 22523–22533. https://proceedings.neurips.cc/paper/2021/file/be26abe76fb5c8a4921cf9d3e865b454-Paper.pdf
[95] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. 2011. Weisfeiler-lehman graph kernels. Journal of Machine Learning Research 12, 9 (2011).
[96] Yucheng Shi, Kaixiong Zhou, and Ninghao Liu. 2023. Engage: Explanation guided data augmentation for graph representation learning. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 104–121.
[97] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE signal processing magazine 30, 3 (2013), 83–98.
[98] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013).
[99] Indro Spinelli, Simone Scardapane, and Aurelio Uncini. 2022. A meta-learning approach for training explainable graph neural networks. IEEE Transactions on Neural Networks and Learning Systems (2022).
[100] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014).
[101] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In International conference on machine learning. PMLR, 3319–3328.
[102] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems 30 (2017).
[103] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. arXiv preprint arXiv:1710.10903 (2017).
[104] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2015. Order matters: Sequence to sequence for sets. arXiv preprint arXiv:1511.06391 (2015).
[105] Minh Vu and My T Thai. 2020. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. Advances in neural information processing systems 33 (2020), 12225–12235.
[106] Rui Wang, Robin Walters, and Rose Yu. 2022. Approximately Equivariant Networks for Imperfectly Symmetric Dynamics. arXiv preprint arXiv:2201.11969 (2022).
[107] Wenxi Wang, Yang Hu, Mohit Tiwari, Sarfraz Khurshid, Kenneth McMillan, and Risto Miikkulainen. 2021. NeuroComb: Improving SAT Solving with Graph Neural Networks. arXiv:2110.14053 (2021). http://www.cs.utexas.edu/users/ai-lab?wang:arxiv21
[108] Xiaoqi Wang and Han Wei Shen. 2022. GNNInterpreter: A Probabilistic Generative Model-Level Explanation for Graph Neural Networks. In The Eleventh International Conference on Learning Representations.
[109] Xiaoqi Wang and Han Wei Shen. 2024. GNNBoundary: Towards Explaining Graph Neural Networks through the Lens of Decision Boundaries. In The Twelfth International Conference on Learning Representations. https://openreview.net/forum?id=WIzzXCVYiH
[110] Xiang Wang, Yingxin Wu, An Zhang, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2022. Reinforced causal explainer for graph neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 45, 2 (2022), 2297–2309.
[111] Boris Weisfeiler and Andrei Leman. 1968. The reduction of a graph to canonical form and the algebra which appears therein. NTI, Series 2, 9 (1968), 12–16.
[112] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R. Lyu, and Yu-Wing Tai. 2020. Towards Global Explanations of Convolutional Neural Networks With Concept Attribution. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 8649–8658. https://doi.org/10.1109/CVPR42600.2020.00868
[113] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning. PMLR, 2048–2057.
[114] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018).
[115] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems 32 (2019).
[116] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. Advances in neural information processing systems 31 (2018).
[117] Zhaoning Yu and Hongyang Gao. 2024. MAGE: Model-Level Graph Neural Networks Explanations via Motif-based Graph Generation. arXiv preprint arXiv:2405.12519 (2024).

[118] Hao Yuan and Shuiwang Ji. 2020. StructPool: Structured Graph Pooling via Conditional Random Fields. In International Conference on Learning Representations. https://openreview.net/forum?id=BJxg_hVtwH
[119] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. Xgnn: Towards model-level explanations of graph neural networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 430–438.
[120] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2020. Explainability in graph neural networks: A taxonomic survey. arXiv preprint arXiv:2012.15445 (2020).
[121] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2022. Explainability in graph neural networks: A taxonomic survey. IEEE transactions on pattern analysis and machine intelligence 45, 5 (2022), 5782–5799.
[122] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. In International Conference on Machine Learning. PMLR, 12241–12252.
[123] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. Advances in neural information processing systems 32 (2019).
[124] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In European conference on computer vision. Springer, 818–833.
[125] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An End-to-End Deep Learning Architecture for Graph Classification. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence (New Orleans, Louisiana, USA) (AAAI'18/IAAI'18/EAAI'18). AAAI Press, Article 544, 8 pages.
[126] Ruochi Zhang, Yuesong Zou, and Jian Ma. 2019. Hyper-SAGNN: a self-attention based graph neural network for hypergraphs. arXiv preprint arXiv:1911.02613 (2019).
[127] Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. 2020. Efficient Probabilistic Logic Reasoning with Graph Neural Networks. In International Conference on Learning Representations. https://openreview.net/forum?id=rJg76kStwH
[128] Yue Zhang, David Defazio, and Arti Ramesh. 2021. Relex: A model-agnostic relational model explainer. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society. 1042–1049.
[129] Zaixin Zhang, Qi Liu, Qingyong Hu, and Cheekong Lee. 2022. Hierarchical Graph Transformer with Adaptive Node Sampling. ArXiv abs/2210.03930 (2022).
[130] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. 2022. Protgnn: Towards self-explaining graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36. 9127–9135.
[131] Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. 2022. On Consistency in Graph Neural Network Interpretation. arXiv preprint arXiv:2205.13733 (2022).
[132] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2921–2929.
[133] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. AI Open 1 (2020), 57–81.

Received 24 October 2022; revised 8 July 2024; accepted 25 August 2024