A Background and foundational concepts

A.1 Interpretability, explainability and transparency

There is a general misunderstanding of the terms explainability and interpretability. While interpretability is the common term in the philosophical literature, the scientific community prefers the term explainability. For this reason, we will only make use of terms that come from the same etymology as “explain”. An explanation is the process (and its product) aiming at making something intelligible through the provision of structured information. Thus, the word explanation can be misleading as it refers to both the method and the result. Note that, for practical reasons, we explicitly use the term "method" to designate the method ("explainability method" or "explanation method") and the term "explanation" to describe the result of this method. As opposed to general explanations, scientific explanations answer only why-questions, where premises are always followed by a deduction. This does not mean that the explanation is unique: we often observe the existence of a large space of alternatives for the same question. Therefore, explanations need to take into consideration the social aspect of the process. Explainability of machine learning models has recently become a top-priority in AI, where it is often abbreviated as explainable Artificial Intelligence (xAI) or interpretable Machine Learning (iML). We adopt the first initialism here to stay as general as possible.

A.2 GNN models and explanation quality

There are several variants of GNNs (graph convolutional networks (GCNs) [38], graph attention networks (GATs) [39], graph isomorphism networks (GINs) [40]), and they differ in their aggregation strategy. In this paper, we restrict our evaluation framework to methods that explain GCNs. We tested our framework on the simple GCN architecture proposed by [38]. Some papers [16, 21, 22, 24, 26, 45, 46] have tested their method for different GNN models and report their results for each one. To rigorously measure the robustness of explainers to the change of GNN model, the authors of [47] define the *consistency* metric. It measures how accuracy varies across different hyperparameters of a model or model architectures. When comparing explanations for different GNNs, those papers tackle the question: does the performance of an explainability method depend on our initial choice of the GNN architecture? In the scope of this paper, we only want to raise awareness on the potential importance of the GNN model on the generated explanations.

A.3 Taxonomy of explainability methods for GNNs

Even if close in meaning, the definitions presented in this section are not to be confused with the ones introduced in [1] and [48].

Input-level/Local vs Model-level/Global explanations. An *input-level* or *example-level* or even *local* explanation identifies features in a given input that are important for its prediction. In contrast, *model-level* or *global* explanations are input-independent: they investigate what input graph patterns can lead to a certain GNN prediction without respect to any specific input example. They explain the general behavior of the model.

Intrinsic explanations vs Post-hoc explanations. *Intrinsic* explanations are produced for models that are self-understandable like linear regression and decision trees. No external method is required to explain their outcomes. *Post-hoc* explanations are brought up for models with higher complexity like neural networks, including GNNs, that do not presume any knowledge of the inner-workings or type of model at hand. In this case, an external method called explainability method is required to bring some clarity.

Model-aware vs model-agnostic explanations. Among post-hoc explanations, we have *model-aware* explanations and *model-agnostic* explanations. *Model-aware* methods look inside the model to extract information. They directly study the model parameters to reveal the relationships between the features in the input space and the output predictions. *Model-agnostic* explanations consider the model as a black-box. To infer what elements are important in the input, they perturb the input and study the changes in the output.

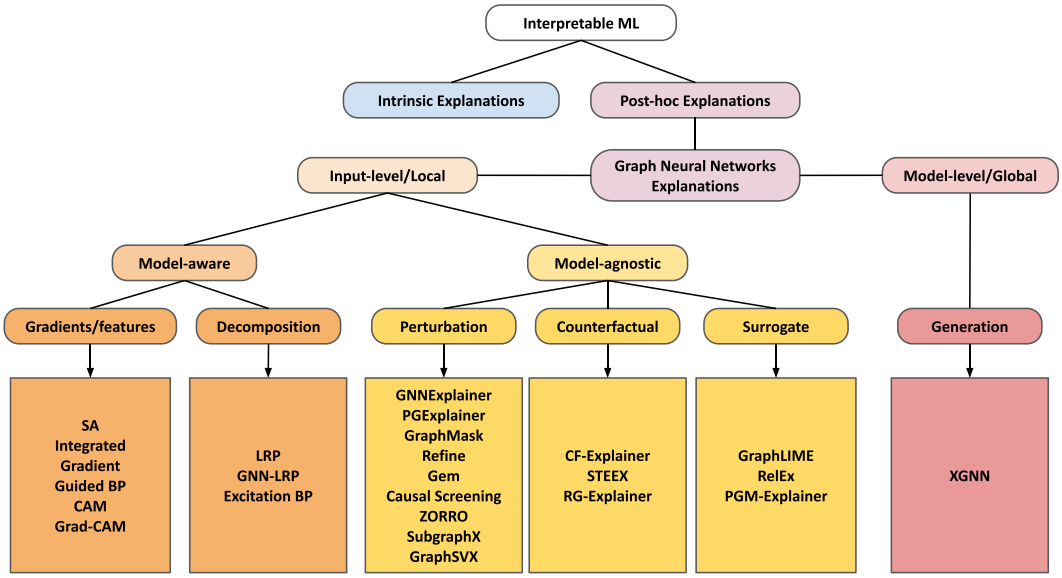

Figure 10: Explanation Taxonomy

A.4 Fidelity measure

Phenomenon	**Model**
$fid+^{acc} = \frac{1}{N}\sum_{i=1}^{N}\left\lvert \mathbb{1}(\hat{y}_i = y_i) - \mathbb{1}(\hat{y}_i^{G_{C\setminus S}} = y_i)\right\rvert$	$fid+^{acc} = \frac{1}{N}\sum_{i=1}^{N}\left\lvert \mathbb{1}(\hat{y}_i = \hat{y}_i) - \mathbb{1}(\hat{y}_i^{G_{C\setminus S}} = \hat{y}_i)\right\rvert = 1 - \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}(\hat{y}_i^{G_{C\setminus S}} = \hat{y}_i)$
$fid-^{acc} = \frac{1}{N}\sum_{i=1}^{N}\left\lvert \mathbb{1}(\hat{y}_i = y_i) - \mathbb{1}(\hat{y}_i^{G_S} = y_i)\right\rvert$	$fid-^{acc} = \frac{1}{N}\sum_{i=1}^{N}\left\lvert \mathbb{1}(\hat{y}_i = \hat{y}_i) - \mathbb{1}(\hat{y}_i^{G_S} = \hat{y}_i)\right\rvert = 1 - \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}(\hat{y}_i^{G_S} = \hat{y}_i)$
$fid+^{prob} = \frac{1}{N}\sum_{i=1}^{N}(f(G_C)_{y_i} - f(G_{C\setminus S})_{y_i})$	$fid+^{prob} = \frac{1}{N}\sum_{i=1}^{N}\lvert f(G_C)_{\hat{y}_i} - f(G_{C\setminus S})_{\hat{y}_i}\rvert$
$fid-^{prob} = \frac{1}{N}\sum_{i=1}^{N}(f(G_C)_{y_i} - f(G_S)_{y_i})$	$fid-^{prob} = \frac{1}{N}\sum_{i=1}^{N}\lvert f(G_C)_{\hat{y}_i} - f(G_S)_{\hat{y}_i}\rvert$

We use the fidelity measures introduced in [26] to evaluate the contribution of the produced explanatory subgraph to the initial prediction, either by giving only the subgraph as input to the model (fidelity-) or by removing it from the entire graph and re-run the model (fidelity+). The fidelity scores capture how good an explainable model reproduces the natural phenomenon or the GNN model logic. The fidelity is measured with respect to the ground truth label or the predicted label according to

the focus choice. Equations A.4 detail the mathematical expressions of the different fidelity scores. The fidelity scores (+/-) can be expressed either with probabilities ($fid^{prob}_{+/-}$) or indicator functions ($fid^{acc}_{+/-}$). While $fid^{prob}_{+/-}$ metrics are more appropriate for evaluating explanations in the context of regression tasks because they are only based on the predicted probabilities, $fid^{acc}_{+/-}$ metrics use the indicator function and are more suitable for classification problems. In this paper, we convey our results with the fidelity metrics that use the indicator function and are more suitable for classification problems.

A.5 Accuracy measure and the concept of groundtruth

The accuracy metric is based on the assumption that we actually know the groundtruth explanation. In current synthetic datasets, node labels are defined based on their position in the graph. Therefore, the groundtruth explanations are artificially built and interpreted as the motifs which the nodes belong to. We are critical towards this method of assigning explanations as it is an a posteriori assignment and is only based on the labeling procedure. How we, humans, synthetically build and explain the node labels is not necessary the right explanation of the GNN model logic. The GNN might put its attention on different graph entities than the ones of the human-intelligible substructures. For this reason, we claim here that accuracy is not the right evaluation metric as it is limited to datasets where we have ground-truth explanations and in these very rare cases, we strongly question their "ground-truth" quality.

A.6 Classification of synthetic datasets

The term synthetic is widely used but its definition is not always clear. Synthetic refers here to data for which we have groudtruth explanations available. But, the procedure to generate the synthetic data and its groundtruth explanations differ. We have identified three origins of groudtruth:

- **Type 1 synthetic data** The true explanation is artificially defined by humans while they construct the graphs and can be identified as the nodes in the k-hop neighborhood of the target node. Such simple explanations can be easily discovered with nearest neighbor search or personalized PageRank. For instance, in the BA-house dataset, the motif house is the expected explanation. These synthetic datasets have been introduced in [10] and are now widely used as benchmarks to evaluate new explainability methods.
- **Type 2 synthetic data** The true explanation is also defined during the construction of the datasets. But, this time, it is more complex than the simple target node neighbourhood. Type 2 synthetic datasets correspond to the three benchmarks introduced in [25]. They have been created to overcome the 5 pitfalls encountered in type 1 synthetic datasets.
- **Type 3 synthetic data** The true explanation finds its origin in scientific experiments, human observations or human intuitions. Type 3 synthetic data often reflect biological and chemical problems, where particular substructures can predict properties for molecules, as in the MUTAG [29] or the MoleculeNet [30] datasets (HIV, BACE, BBBP, Tox21, QM7), or predict properties of proteins, as in the Enzymes dataset [29].

In this paper, we tested explainability methods on type 1 synthetic datasets to highlight their limitation in a rigorous evaluation of explainers. In addition, type 1 and type 3 are the most common families of synthetic data in recent papers [9–14, 16, 16–19, 26]. We have not tested the methods on type 3 synthetic datasets since they are made for graph classification and regression tasks.

A.7 Mask transformation strategies

Sparsity. Sparsity is defined as the minimum percentage X of edges to remove from the initial graph. The sparsity strategy consists in keeping only edges which belong to the (100-X)% highest values in the mask. A sparsity of 70% or 0.7 means that we keep at least 30% of the edges in the mask. Some very sparse explainability methods might return sparser explanations with even less edges. But, we have the assurance that explanations cannot be bigger. Note that the size of the explanation is dependent on the size of the graph: if we change the dataset, the number of edges contained in the transformed masks will be different. Thus, for the sparsity strategy, the size of the explanation depends on the dataset.

Datasets		BA-House	BA-Grid	Tree-Cycle	Tree-Grid	BA-Bottle
Base	Type	BA graph	BA graph	Tree	Tree	BA graph
	Size	300 nodes	300 nodes	height 8	height 8	300 nodes
Motif	Type	house	grid	cycle	grid	bottle
	Size	5 nodes	9 nodes	6 nodes	9 nodes	5 nodes
	Number	80	80	60	80	80
# Features		constant	constant	constant	constant	constant
# Classes		4	2	2	2	4

Table 2: Synthetic datasets statistics

Datasets	Cora	CiteSeer	PubMed	Chameleon	Squirrel	Actor	Facebook	Cornell	Texas	Wisconsin
# Nodes	2708	3327	19717	2277	5201	7600	22470	183	183	251
# Edges	5429	4732	44338	36101	217073	33544	171002	295	309	499
# Features	1433	3703	500	2325	2089	931	4714	1703	1703	1703
# Classes	7	6	3	5	5	5	4	5	5	5

Table 3: Real datasets statistics

Threshold. Threshold is a value between 0 and 1 that defines the lowest value for edge importance.The threshold strategy consists in keeping the edges whose value in the mask is greater than the threshold. For a threshold $\tau \in [0, 1]$, we keep only values in the mask greater than τ. This leads to explanations of different sizes among the explainability methods, since some methods might value edges high while other methods give to their most important edges values below 0.5. Thus, for the threshold strategy, the size of the explanation depends on the method.

Topk. Topk is the number of edges in the explanatory subgraph. The topk strategy only keeps the top k highest values in the mask. This strategy always returns explanations with a similar absolute size whatever the dataset and the method. We also define the directed topk strategy and the undirected topk strategy. While the first one keeps the top k directed edges, the second one avoids double counting of node-to-node connections and returns explanations with k connections, i.e. the explanation is an undirected subgraph of k edges.

B Experimental details

B.1 Datasets

Details on how the synthetic datasets were constructed can be found in Table 2. Table 3 presents the structural properties of the real datasets. eBay graph characteristics are detailed in Table 4.

Synthetic datasets. We use type 1 synthetic datasets introduced in [10] (see Appendix A.6), which are widely used in the xAI litterature [9–14, 16, 16–19, 26]. We follow the code[2] of Vu et al. [13] to create the synthetic datasets. In these datasets, each input graph is a combination of a base graph and a set of motifs. Diverse motifs (house, cycle, grid, bottle) are plugged in on a base graph (Barabasi graph or tree). Nodes are labeled based on their position in the graph: they receive a label 0 if they are in the base graph and a non-zero label if they belong to a motif. For house and bottle, the position in the motif is also important. For grid and cycle, we only look if the node belongs to the shape. The ground-truth label of each node on a motif is determined based on its role in the motif. As the labels are determined based on the motif's structure, the explanation for the role's prediction of a node are the nodes in the same motif. Thus, the ground-truth explanation in these datasets are the nodes in the same motif as the target.

Citation datasets. We consider three citation network datasets: Citeseer, Cora and Pubmed[49]. The datasets contain sparse bag-of-words feature vectors for each document and a list of citation links

[2] https://github.com/vunhatminh/PGMExplainer/tree/master/PGM_Node/Generate_XA_Data

Dataset	# Nodes	# Txn Nodes	# Edges	# Features	# Classes	# Positive label	Train/Val/Test split
eBay	288853	207749	1225808	114	2	3081 (1.48% of *txns*)	0.75/0.15/0.1

Table 4: eBay graph statistics

between documents. Citation links are treated as (undirected) edges. Each document has a class label. For training, we only use 20 labels per class, but all feature vectors.

Facebook. This dataset is a page-page graph of verified Facebook sites. Nodes correspond to official Facebook pages, links to mutual edges between sites. Node features are extracted from the site descriptions. The task is multi-class classification of the site category.

Wikipedia network. Chameleon and squirrel are two page-page networks on specific topics in Wikipedia. In those datasets, nodes represent web pages and edges are mutual links between pages. And node features correspond to several informative nouns in the Wikipedia pages. We classify the nodes into five categories in terms of the number of the average monthly traffic of the web page.

Actor co-occurrence network. This dataset is the actor-only induced subgraph of the film-director-actor-writer network. Each node corresponds to an actor, and the edge between two nodes denotes co-occurrence on the same Wikipedia page. Node features correspond to some keywords in the Wikipedia pages. Nodes are classified into five categories in terms of words on the actor's Wikipedia.

WebKB. WebKB1 is a web page dataset collected from computer science departments of various universities by Carnegie Mellon University. We use the three subdatasets of it, Cornell, Texas, and Wisconsin, where nodes represent web pages, and edges are hyperlinks between them. Node features are the bag-of-words representation of web pages. The web pages are manually classified into the five categories, student, project, course, staff, and faculty.

eBay. We conducted a case study on a real-world dataset with collaboration with the eBay Risk Team. We construct a bipartite graph with 2 different kinds of nodes: transaction nodes (*txn*), which are what we want to predict as targets, and entity nodes, which are unique assets including buyer account, payment tokens, email, IP address, and shipping address, acting like a linkage medium to connect *txns* together. If a *txn* has relation with an entity, we put an edge between these two nodes. Two different *txns* will be linked to the same entity node if they are sharing the same entity, *e.g.* the same shipping address is used in the two *txns*. Each *txn* is labeled as legit or fraudulent, and carries features provided by eBay risk system. These features include the information of transaction itself and expert-designed features extracted from its neighbors such as user and email information. For the entity nodes, the feature vectors are filled with zero value. Our source data is sampled from e-commerce history transaction logs. To ensure the connectivity of the graph, we first sample some seed *txns* within certain period of time, and then expand 3 hop neighbors from these seeds, and at each hop, no more than 32 neighbors are picked. Then we collect all involved nodes. The final graph has a size of 288,853 nodes (includes 207,749 *txn* nodes) and 1,225,808 edges. Among the *txn* nodes, 3,081 are labeled as fraudulent. Each *txn* node has 114 features. The graph we are using is the same with *eBay-small* graph in paper xFraud [50]. The desensitization version data is available for legitimate, non-commercial usage after submitting the application [3]. According to our experience, user based features usually contribute more, and payment tokens are usually a stronger evidence of fraud propagation among other entities. For example, a transaction with large user behavior change may be caused by account takeover attack, and a transaction using a payment token which has been used in other proved fraudulent purchases are more likely to be malicious.

B.2 Explainability methods

Model-aware. Gradient-based methods compute the gradients of target prediction with respect to input features by back-propagation. Features-based methods map the hidden features to the input space via interpolation to measure important scores. Decomposition methods measure the importance of input features by distributing the prediction scores to the input space in a back-propagation manner.

[3] https://github.com/eBay/xFraud

Model-agnostic. Perturbation-based methods use masking strategy in the input space to perturb the input. Surrogate models use node/edge dropping, BFS sampling and node feature perturbation. Counterfactual methods generate counterfactual explanations by searching for a close possible world using adversarial perturbation techniques [51].

Explainer	Model-aware/agnostic	Target	Type	Flow
SA	Model-aware	N/E	Gradient	Backward
IG	Model-aware	N/E	Gradient	Backward
Grad-CAM	Model-aware	N	Gradient	Backward
Occlusion	Model-agnostic	N/E	Perturbation	Forward
GNNExplainer	Model-agnostic	N/E/NF	Perturbation	Forward
PGExplainer	Model-agnostic	N/E	Perturbation	Forward
PGM-Explainer	Model-agnostic	N/E	Perturbation	Forward
SubgraphX	Model-agnostic	N/E	Perturbation	Forward
PageRank	Model-agnostic	N	Baseline	-
Distance	Model-agnostic	N	Baseline	-

Table 5: Explainability methods tested in the context of our evaluation framework.

B.3 GNN training

For all datasets, we use Adam optimizer [52]. The graph convolution network (GCN) has 2 or 3 layers with 16, 20 or 32 units. We eventually apply regularization on the weights with a weight decay factor of 0.05 or 0.005. We also apply dropout for some datasets. We indicate all parameters for each family of datasets. For synthetic datasets and for Facebook dataset, we use a 0.8/0.15/0.1 train/val/test split. For the Planetoid datasets, we use the default split: 140/500/1000 for Cora, 120/500/1000 for CiteSeer and 60/500/1000 for PubMed. We use the default train/val/test split for all other real datasets, namely 0.48/0.32/0.2. We further describe the model accuracy, F1-score, precision and recall for synthetic and real datasets.

B.4 Protocol

For each dataset, we first train a graph convolution network (GCN) as introduced by Kipf and Welling [38]. For synthetic datasets, we use the version implemented by Rex Ying [4] [10]. For real datasets, we use the original GCN implementation from Kipf [5]. We use the trained model to do predictions of node targets of a testing set. We test twelve explainability methods on the synthetic and real datasets. We select 100 testing nodes which label we want to explain. We run each experiment on 5 different seeds and present the average results. All computations were run on ETH Zurich internal clusters:

[4] https://github.com/RexYing/gnn-model-explainer
[5] https://github.com/tkipf/gcn

Datasets	Syn	WebKB	Citat., Wiki Faceb., Actor	eBay
layers	3	2	2	2
hidden dim	20	32	16	32
epochs	1000	400	200	500
learning rate	0.001	0.001	0.01	0.001
weight decay	$5 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
dropout	0	0.2	0.5	0.5

Table 6: GNN model and training parameters

Datasets	BA House	BA Grid	Tree Cycle	Tree Grid	BA Bottle
accuracy	0.986	1	1	0.895	1
F1-score	0.976	1	1	0.897	1
recall	0.979	1	1	0.87	1
precision	0.972	1	1	0.925	1

Table 7: GNN testing accuracy on synthetic datasets

Datasets	Cora	CiteSeer	PubMed	Chameleon	Squirrel	Actor	Facebook	Cornell	Texas	Wisconsin	eBay
accuracy	0.803	0.676	0.779	0.632	0.376	0.286	0.926	0.532	0.511	0.535	0.953
F1-score	0.799	0.652	0.777	0.64	0.35	0.254	0.92	0.344	0.285	0.397	0.594
recall	0.817	0.652	0.782	0.634	0.373	0.249	0.918	0.339	0.277	0.406	0.7
precision	0.781	0.651	0.772	0.647	0.333	0.261	0.923	0.355	0.297	0.398	0.566

Table 8: GNN model accuracy on real datasets and production data (eBay graph)

the HPC clusters of ID SIS HPC (Euler Clusters). We use 2 GPUs. For the synthetic datasets, we only explain the nodes that belong to a noteworthy motif (house, grid, cycle or bottle). For eBay, we only explain fraudulent nodes. The returned explanations are in the form of a mask over the nodes/edges and/or the node features. For the methods that return node masks, we convert those into edge masks: we assign to an edge the average importance of the nodes it connects. We apply the topk mask transformation strategy to reduce the size of the masks to k edges. This allows a fair comparison of the explainability methods by enforcing a similar size to all explanations. We compare our methods for different explanations with 1, 5, 10, 15, 20, 25, 50 and 100 edges. We also consider $k = 10$ edges as being a decent size for an explanation as it is still large to have interesting insight but sparse enough to stay human-intelligible. We compare methods in four different settings defined as the combinations of the two possible focuses, *phenomenon* and *model*, and mask natures, *hard* or *soft* masks.

C Additional results

C.1 Variance analysis

We report here the average and standard deviation of our experiments run over 5 random seeds. Figure 11 shows the characterization score of the explainability methods on the real datasets. We observe that the variance stays small for every method, thus proving the robustness of our evaluation.

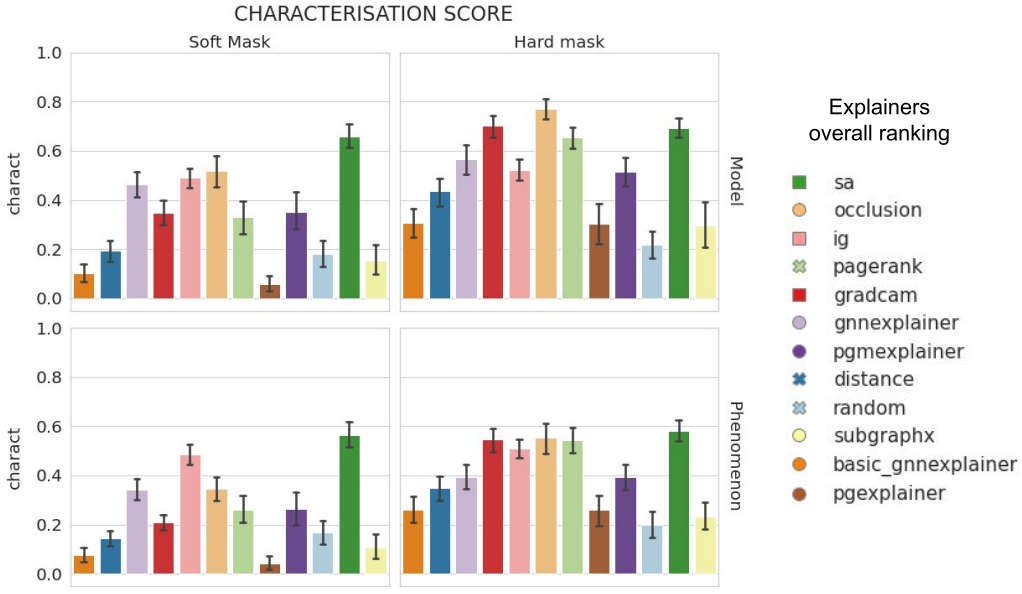

Figure 11: Characterization score of each explainability method in the four settings. Results are averaged over all real datasets and 5 different seeds and the standard deviation is indicated as the black segment.

C.2 Mask properties and method performance

To understand the reasons behind the performance of an explainabiality method, we can further study the properties of its explanations. We look at four properties of the mask:

- **Mask size**: the number of edges selected by the method before any mask transformation. A large mask has non-zero values on most of the edges. A sparse mask has only a few non-zero edges.
- **Mask entropy**: the entropy of the value distribution in the mask. A high entropy means that the distribution of values in the mask is close to the uniform distribution: edges have different importance levels. A low entropy means that most of the values are concentrated around a few significant scalars: all edges have similar importance.
- **Mask maximum value**: the maximum importance level in the mask. A high maximum value means that the explainability method rates high important nodes. Note that if you have a large maximum value, i.e. close to 1, and the mask entropy is also large, the explanation covers the whole spectrum of importance levels.
- **Mask connectivity**: the fraction of connected components in the explanatory subgraph. The mask connectivity represents the ratio of connected components. Some explanations are composed of distinct edges without any connection, while others consist of one unique connected subgraph. The mask connectivity is computed as the fraction of number of connected components over the total number of edges in the explanatory subgraph. A ratio close to 0 indicates high connectivity in the explanation, while a ratio of 1 indicates that all edges are disconnected. We favor explanations with low ratios because highly connected explanations are more human-intelligible.

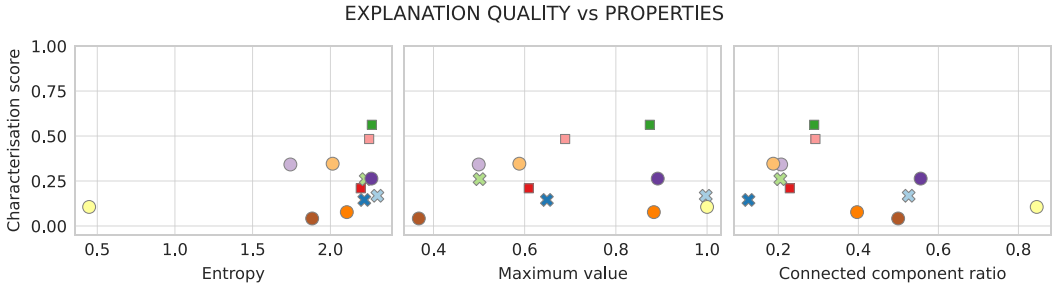

Figure 12: Explanation quality estimated with the characterization score vs. mask properties, *i.e.* entropy, maximum value and connected component ratio in a typical explanation of default size 10 edges. Results are averaged over all real datasets and 5 different seeds.

The mask analysis aims at finding correlations between methods' internal characteristics and their performance. We observe for instance that the best performing method Saliency generates masks with high entropy, i.e. the 10 edges have almost the same importance, high maximum value, i.e. all edges are considered as very important, and low connected component ratio, i.e. the explanation i almost one unique subgraph. When those three conditions are not met, it seems that the explanation cannot be a good characterization of the GNN predictions. This conclusion has been drawn from initial observations on some real datasets. Our study is still in its early stages. We have not yet found general rules that relate the mask structure to the method performance. In the future, we will further investigate if there exists a real correlation between the mask properties and the characterization score.

C.3 Different GNN models

We have tested three GNN models: graph convolutional networks (GCN) [38], graph attention networks (GAT) [39] and graph isomorphism networks (GIN) [40]. GAT uses a neural network to learn the best weighting factors for aggregation and GIN's aggregator follows the Weisfeiler-Lehman test to better discriminate graphs. We report in Figure 13 the Fidelity+ and Fidelity- scores for the dataset Cora in the specific setting of explanations generated as hard masks for a phenomenon focus. We observe the strongest performance differences for the Fidelity- measure: Saliency's performance drops with GCN models, Integrated Gradients performs much better with GIN models, GNNexplainer seems to perfom well only with GAT models. The results are more consistent for Fidelity+. Based on these findings, we would like to further study the relationship between GNN models and epxlainability.

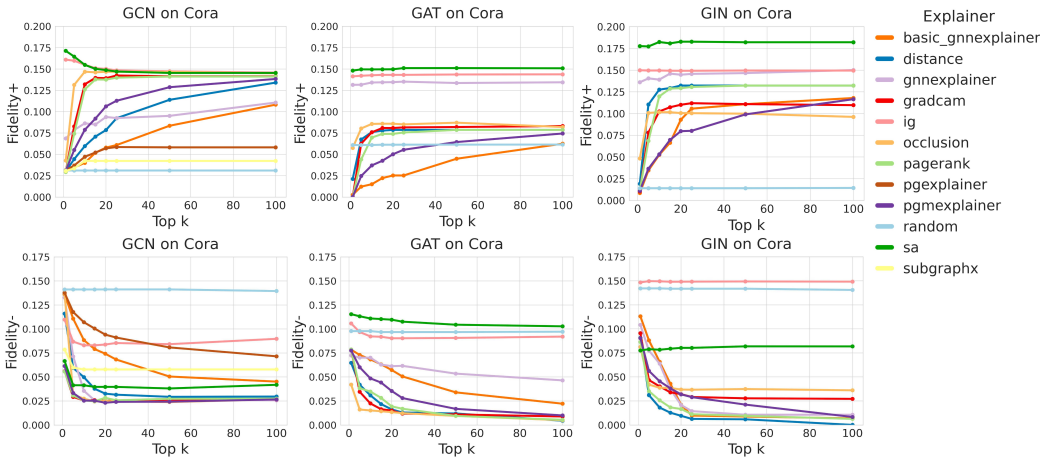

Figure 13: The Fidelity+ (top) and Fidelity- (bottom) comparisons between different GNN explanation techniques under different Topk levels for Cora dataset. Here, explanations are generated as hard masks (aspect 2) with a phenomenon focus (aspect 1).

D Evaluation framework of explainability methods for graph classification

In this paper, we highlight the limitations of evaluation protocol of the explainability methods in the context of node classification. Here, we demonstrate to the readers that similar limitations exist for graph classification. Table 9 shows that existing works which seek to explain graph classification tasks only use small and different sets of metrics. In contrast with node classification, more papers test their methods on real graphs. But, they still do not consider the aspect of time and the influence of GNN accuracy on its explainability. Taking these observations into consideration, we wish to extend our evaluation framework to graph classification tasks in order to offer a universal evaluation protocol for any type of GNN tasks.

Table 9: XAI LITERATURE FOR GNN GRAPH CLASSIFICATION. Acc defines the accuracy (AUC, F1-score) measured with respect to the groundtruth, Fid+ and Fid- refer to the fidelity metrics as defined in [26]. "Time" indicates if the paper has run a comparative analysis of the computation time of the explainability methods. The final column "GNN accuracy" shows if the authors have reported the testing accuracy of their model.

Paper Type	Year	Explainer	Target	Synthetic			Real			Time	GNN Accuracy
				Acc	Fid-	Fid+	Acc	Fid-	Fid+		
Method[9]	2019	LRP	E				✓		✓		
Method[11]	2020	PGExplainer	E				✓			✓	0.92 - 1.00
Method[12]	2020	RelEx	E				✓				
Method[13]	2020	PGM-Explainer	E				✓				0.85-1.00
Method[20]	2020	XGNN	E		✓*			✓*			
Method[21]	2021	GNN-LRP	E		✓*	✓*		✓*	✓*		0.77-0.95
Method[22]	2021	Causal Screening	E					✓*		✓	0.64 - 0.98
Method[16]	2021	SubgraphX	E			✓			✓	✓	0.86-0.99
Method[23]	2021	Refine	E	✓	✓*		✓	✓*		✓	0.60-1.00
Method[14]	2021	RG-Explainer	E	✓			✓				
Method[19]	2021	Gem	E					✓*		✓	
Taxonomy[24]	2019	CG,EB,c-EB CAM,Grad-CAM	E						✓		0.88-0.99
Taxonomy[26]	2020	GNNExplainer,PGExplainer SubgraphX,DeepLift GNN-LRP,Grad-CAM,XGNN	E	✓	✓	✓	✓	✓	✓		0.44-0.91
Taxonomy [28]	2022	VanillaGrad,IntergratedGrad GNNExplainer,PGMExplainer	E					✓*			

* Different denomination in the paper, but the same evaluation mechanism as ours.