APPENDIX

A EXPERIMENTAL SETUP

All experiments were conducted using the Ubuntu 18.04 operating system on an NVIDIA DGX Station equipped with four V100 GPU cards, each having 128GB of GPU memory. The system also included 256GB of RAM and a 20-core Intel Xeon E5-2698 v4 2.2 GHz CPU.

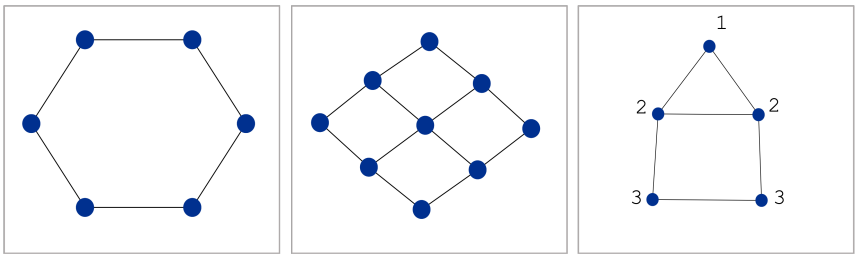
The datasets for factual and counterfactual explainers follow an 80:10:10 split for training, validation and testing. We explain some of our design choices below.

- For factual explainers, the inductive explainers are trained on the training data and the reported results are computed on the entire dataset. We also report results only on test data (please see Sec. A.5) comparing only inductive methods. Transductive methods are run on the entire dataset.
- For counterfactual explainers, the inductive explainers are trained on the training data, and the reported results are computed on the test data. Since transductive methods do not have the notion of training and testing separately, they are run only on the test data.

A.1 BENCHMARK DATASETS

Datasets for Node classification: The following datasets have node labels and are used for the node classification task.

- **TREE-CYCLES Ying et al. (2019b):** The base graph used in this dataset is a binary tree, and the motifs consist of **6-node cycles** (Figure D(a)). The motifs are connected to random nodes in the tree. Non-motif nodes are labeled 0, while the motif nodes are labeled 1.
- **TREE-GRID Ying et al. (2019b):** The base graph used in this dataset is a binary tree, and the motif is a $\mathbf{3 \times 3}$ **grid** connected to random nodes in the tree (Figure D(b)). Similar to the tree-cycles dataset, the nodes are labeled with binary classes (0 for the non-motif nodes and 1 for the motif nodes).
- **BA-SHAPES Ying et al. (2019b):** The base graph in this dataset is a Barabasi-Albert (BA) graph. The dataset includes **house-shaped** structures composed of 5 nodes (Figure D (c)). Non-motif nodes are assigned class 0, while nodes at the top, middle, and bottom of the motif are assigned classes 1, 2, and 3, respectively.



(a) **(b)** (c)

Figure D: Motifs used in (a) Tree-Cycles, (b) Tree-Grid and (c) BA-Shapes datasets for the node classification task. Please note the following. (i) Tree-Cycles and Tree-Grid have labels 0 and 1 for the non-motif and the motif nodes, respectively. Hence, all nodes in (a) and (b) have label 1. (ii) BA-Shapes dataset has 4 classes. Non-motif nodes have labels 0; motif nodes have integral labels depending on the position in the house motif. The other labels are 1 (top node), 2 (middle nodes) and 3 (bottom nodes). They are represented in (c).

Datasets for Graph Classification: The following datasets are used for the graph classification task and contain labeled graphs.

- **MUTAG Ivanov et al. (2019) and Mutagenicity Riesen & Bunke (2008); Kazius et al. (2005):** These are graph datasets containing chemical compounds. The nodes represent atoms, and the edges represent chemical bonds. The binary labels depend on the mutagenic effect of the compound

on a bacterium, namely mutagenic or non-mutagenic. MUTAG and Mutagenicity datasets contain 188 and 4337 graphs, respectively.

- **AIDS:** Ivanov et al. (2019) This dataset contains small molecules. The nodes and edges are atoms and chemical bonds, respectively. The molecules are classified by whether they are active against the HIV virus or not.
- **Proteins Borgwardt et al. (2005); Dobson & Doig (2003) and DD Dobson & Doig (2003):** These datasets are comprised of proteins categorized into enzymes and non-enzymes. The nodes represent amino acids, and an edge exists between two nodes if their distance is less than 6 Angstroms.
- **NCI1 Wale et al. (2008):** This dataset is derived from cheminformatics and represents chemical compounds as input graphs. Vertices in the graph correspond to atoms, while edges represent bonds between atoms. This dataset focuses on anti-cancer screenings for cell lung cancer, with chemicals labeled as positive or negative. Each vertex is assigned an input label indicating the atom type, encoded using a one-hot-encoding scheme.
- **IMDB-B Yanardag & Vishwanathan (2015):** The IMDB-BINARY dataset is a collection of movie collaboration networks, encompassing the ego-networks of 1,000 actors and actresses who have portrayed roles in films listed on IMDB. Each network is represented as a graph, where the nodes correspond to the actors/actresses, and an edge is present between two nodes if they have shared the screen in the same movie. These graphs have been constructed specifically from movies in the Action and Romance genres, which are the class labels.
- **REDDIT-B Yanardag & Vishwanathan (2015):** REDDIT-BINARY dataset encompasses graphs representing online discussions on Reddit. Each graph has nodes representing users, connected by edges when either user responds to the other's comment. The four prominent subreddits within this dataset are IAmA, AskReddit, TrollXChromosomes, and atheism. IAmA and AskReddit are question/answer-based communities, while TrollXChromosomes and atheism are discussion-based communities. Graphs are labeled based on their affiliation with either a question/answer-based or discussion-based community.
- **GRAPH-SST2 Yuan et al. (2022):** The Graph-SST2 dataset is a graph-based dataset derived from the SST2 dataset Socher et al. (2013), which contains movie review sentences labeled with positive or negative sentiment. Each sentence in the Graph-SST2 dataset is transformed into a graph representation, with words as nodes and edges representing syntactic relationships capturing the sentence's grammatical structure. The sentiment labels from the original SST2 dataset are preserved, allowing for sentiment analysis tasks using the graph representations of the sentences.
- **ogbg-molhiv** Allamanis et al. (2018): ogbg-molhiv is a molecule dataset with nodes representing atoms and edges representing chemical bonds. The node features represent various properties of the atoms like chirality, atomic number, formal charge etc. Edge attributes represent the bond type. We study binary classification task on this dataset. The task is to achieve the most accurate predictions of specific molecular properties. These properties are framed as binary labels, indicating attributes like whether a molecule demonstrates inhibition of HIV virus replication or not.

A.2 Details of GNN model Φ used for Node Classification

We use the same GNN model used in CF-GNNEXPLAINER and CF^2. Specifically, it is a Graph Convolutional Networks Kipf & Welling (2016) trained on each of the datasets. Each model has 3 graph convolutional layers with 20 hidden dimensions for the benchmark datasets. The non-linearity used is *relu* for the first two layers and *log* $\mathrm{softmax}$ after the last layer of GCN. The learning rate is 0.01. The train and test data are divided in the ratio 80:20. The accuracy of the GNN model Φ for each dataset is mentioned in Table I.

Table I: Accuracy of black-box GNN Φ on the datasets used for node classification, for evaluation of counterfactual explainers. Φ is a GCN Kipf & Welling (2016) for this task

Dataset	Train accuracy	Test Accuracy
Tree-Cycles	0.9123	0.9086
Tree-Grid	0.8434	0.8744
BA-Shapes	0.9661	0.9857

A.3 Details of Base GNN Model Φ for the Graph Classification Task

Our GNN models have an optional parameter for continuous edge weights, which in our case represents explanations. Each model consists of 3 layers with 20 hidden dimensions specifically designed for benchmark datasets. The models provide node embeddings, graph embeddings, and direct outputs from the model (without any softmax function). The output is obtained through a one-layer MLP applied to the graph embedding. We utilize the max pooling operator to calculate the graph embedding. The dropout rate, learning rate, and batch size are set to 0, 0.001, and 128, respectively. The train, validation, and test datasets are divided into an 80:10:10 ratio. The algorithms run for 1000 epochs with early stopping after 200 patience steps on the validation set. The performance analysis of the base GNN models (Kipf & Welling (2016); Veličković et al. (2018); Xu et al. (2019); Hamilton et al. (2017)) for each graph classification dataset is presented in Table J.

Table J: Test accuracy of black-box GNN Φ trained for the graph classification task, averaged over 10 runs with random seeds. We train multiple GNNS for this task to test explainers for stability against GNN architectures.

Dataset	GCN	GAT	GIN	GraphSAGE
Mutagenicity	0.8724 ± 0.0092	0.8685 ± 0.0111	0.8914 ± 0.0101	0.8749 ± 0.0059
Mutag	0.925 ± 0.0414	0.8365 ± 0.0264	0.9542 ± 0.0149	0.8323 ± 0.0445
Proteins	0.8418 ± 0.0144	0.8362 ± 0.0269	0.8352 ± 0.0165	0.8408 ± 0.0124
IMDB-B	0.8318 ± 0.0197	0.8292 ± 0.015	0.8554 ± 0.027	0.8373 ± 0.0093
AIDS	0.999 ± 0.0005	0.9971 ± 0.0068	0.9797 ± 0.0099	0.9903 ± 0.0088
NCI1	0.8243 ± 0.028	0.8096 ± 0.015	0.8365 ± 0.0201	0.8303 ± 0.0137
Graph-SST2	0.957 ± 0.001	0.9603 ± 0.0009	0.9552 ± 0.0014	0.9611 ± 0.0011
DD	0.736 ± 0.0377	0.7312 ± 0.048	0.7693 ± 0.0238	0.7541 ± 0.0415
REDDIT-B	0.8984 ± 0.0247	0.8444 ± 0.0266	0.6886 ± 0.1231	0.8733 ± 0.0196
ogbg-molhiv	0.9729 ± 0.0002	0.9722 ± 0.0010	0.9726 ± 0.0003	0.9725 ± 0.0005

A.4 Details of Factual Explainers for the Graph Classification Task

In many cases, explainers generate continuous explanations that can be used with graph neural network (GNN) models, which can handle edge weights. To be able to use explanations in our GNN models, we map them into $[0, 1]$ using a sigmoid function if not mapped. While generating performance results, we calculate top-k edges based on their scores instead of assigning a threshold value (e.g., 0.5). However, there are some approaches, such as GEM and SubgraphX, that do not rely on continuous edge explanations.

GEM employs a variational auto-encoder to reconstruct ground truth explanations. As a result, the generated explanations can include negative values. While our experiments primarily focus on the order of explanations and do not require invoking the base GNN in the second stage of GEM, we can still use negative explanation edges.

On the other hand, SubgraphX ranks different subgraph explanations based on their scores. We select the top 20 explanations and, for each explanation, compute the subgraph. Then, we enhance the importance of each edge of a particular subgraph by incrementing its score by 1. Finally, we normalize the weights of the edges. This process allows us to obtain continuous explanations as well. Moreover, since SubgraphX employs tree search, its scalability is limited when dealing with large graphs. For instance, in the Mutagenicity dataset, obtaining explanations for 435 graphs requires approximately 26.5 hours. To address this challenge, we restricted our analysis to test graphs when calculating explanations using SubgraphX. It is important to include this disclaimer, working on only subset of graphs may introduce potential biases or noises in the results.

A.5 Factual Explainers: Inductive Methods on Test Set

The inductive factual explainers are run only on the test data and the results are reported in Figure E. The results are similar to the ones where the methods are run on the entire dataset (Figure 2). Consistent with the earlier results, PGEXPLAINER consistently delivers inferior results compared to other

baseline methods, and no single technique dominates across all datasets. Overall, RCEXPLAINER could be recommended as one of the preferred choices.

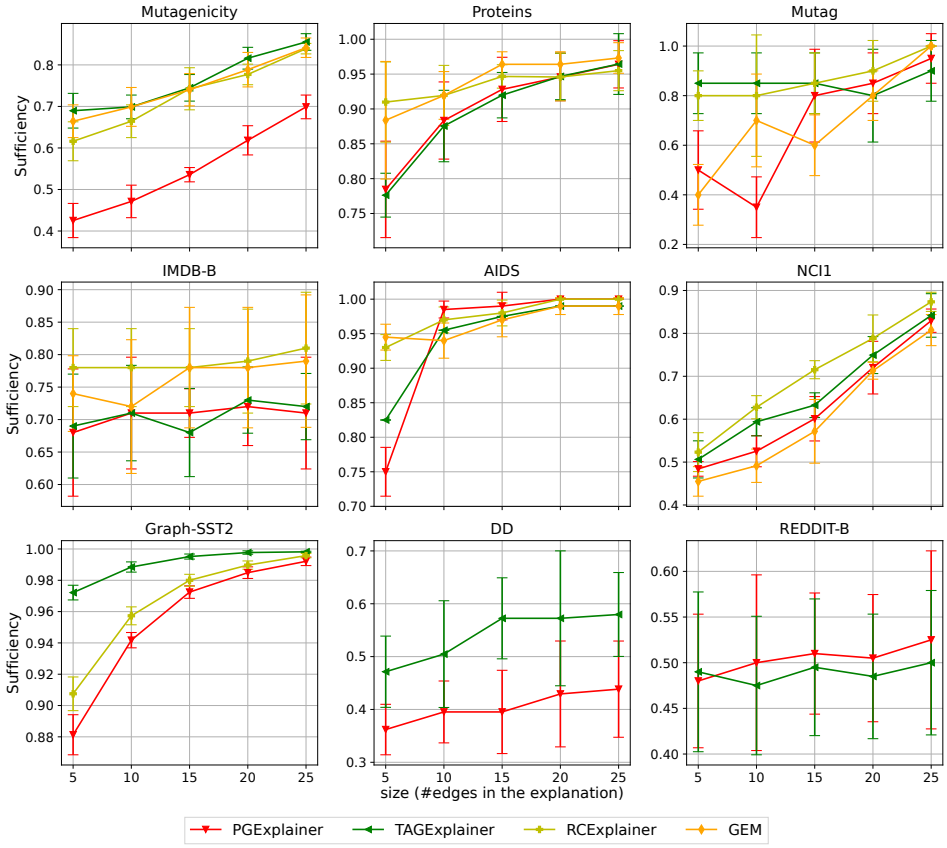


Figure E: Sufficiency of the inductive factual explainers against the explanation size on only test data. For factual explanations, higher is better. We omit those methods for a dataset that throw an out-of-memory (OOM) error and are not scalable.

A.6 CODES AND IMPLEMENTATION

Table K shows the code bases we have used for the explainers. We have adapted the codes based on our base GNN models. Our repository, https://github.com/idea-iitd/gnn-x-bench/, includes the adaptations of the methods to our base models.

Table K: Reference of code repositories.

Method	Repository
PGExplainer Luo et al. (2020)	https://github.com/LarsHoldijk/RE-ParameterizedExplainerForGraphNeuralNetworks/
TAGExplainer Xie et al. (2022)	https://github.com/divelab/DIG/tree/main/dig/xgraph/TAGE/
CF^2 Tan et al. (2022)	https://github.com/chrisjtan/gnn_cff
RCExplainer Bajaj et al. (2021)	https://developer.huaweicloud.com/develop/aigallery/notebook/detail?id=e41f63d3-e346-4891-bf6a-40e64b4a3278
GNNExplainer Ying et al. (2019b)	https://github.com/LarsHoldijk/RE-ParameterizedExplainerForGraphNeuralNetworks/
GEM Lin et al. (2021b)	https://github.com/wanyu-lin/ICML2021-Gem/
SubgraphX Yuan et al. (2021)	https://github.com/divelab/DIG/tree/main/dig/xgraph/SubgraphX

A.7 FEASIBILITY

Counterfactual explanations: As shown in Table L, we observe statistically significant deviations from the expected values in two out of three molecular datasets. This suggests a heightened probability of predicting counterfactuals that do not correspond to feasible molecules. This finding underscores

Table L: Assessing the statistical significance of deviations in the number of connected graphs between the test set and their corresponding counterfactual explanations on molecular datasets. Statistically significant deviations with p-value< 0.05 are highlighted.

Dataset	RCExplainer			CF^2		
	Expected Count	Observed Count	p-value	Expected Count	Observed Count	p-value
Mutagenicity	233.05	70	< 0.00001	206.65	0	< 0.00001
Mutag	11	9	0.55	4	1	0.13
AIDS	17.6	8	< 0.00001	1.76	0	0.0001

a limitation of counterfactual explainers, which has received limited attention within the research community.

Factual explanations: The feasibility metric is commonly used in the context of counterfactual graph explainers because it measures how feasible it is to achieve a specific counterfactual outcome. In other words, it assesses the likelihood of a counterfactual scenario being realized given the constraints and assumptions of the underlying base model. On the other hand, factual explainers aim to explain why a model makes a certain prediction based on the actual input data. They do not involve any counterfactual scenarios, so the feasibility metric is not relevant in this context. Instead, factual explainers may use other metrics such as sufficiency and reproducibility to provide insights into how the model is making its predictions. Therefore, we have not used feasibility metrics for factual explanations.

B Sufficiency of Factual Explanations under Topological Noise

We check the sufficiency of the factual explanations under noise for four different datasets. Figure F demonstrates the results, including when there is no noise (i.e., when X = 0). We set the explanation size (i.e., the number of edges) to 10 units. We observe that, in most cases, increasing noise results in a decrease in the sufficiency metric for the Mutagenicity and AIDS datasets, which is expected. However, for the Proteins and IMDB-B datasets, even though there are still drops for some methods, others remain stable in sufficiency across different noise levels. This demonstrates that, despite the changes in explanations caused by noise, GNN may still predict the same class under noisy conditions.

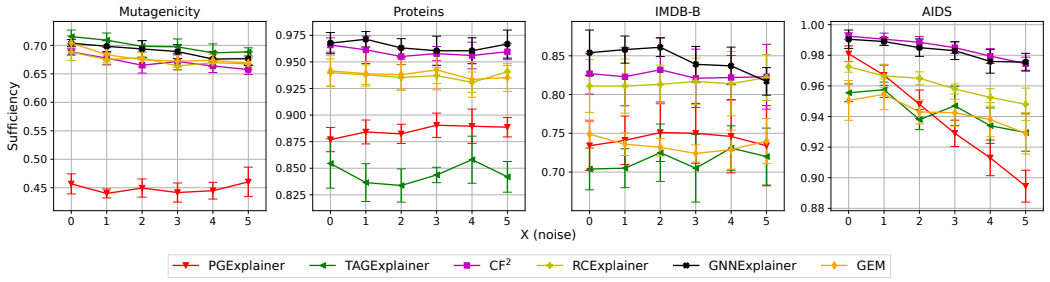


Figure F: Sufficiency of factual explainers under topological noise.

C Stability

In addition to stability against topological noise, different seeds, and different GNN architecture, we also analyze *stability against feature pertubation* and *stability against topological adversarial attack*.

For feature perturbation, we first select the percentage of nodes to be perturbed ($X\% \in \{10, 20, 30, 40, 50\}$). Then, perturbation operation varies depending on the nature of node features (continuous or discrete). For the Proteins dataset (continuous features); for each feature f, we compute its standard deviation σ_f. Then we sample a value uniformly at random from $\Delta \sim [-0.1, 0.1]$. The feature value f_x is perturbed to $f_x + \Delta \times \sigma_f$. For other datasets (discrete features), for each selected node, we flip its feature to a randomly sampled feature.

For topology adversarial attack, we follow the flip edge method from Wan et al. (2021) with a query size of one and vary the number flip count across datasets.

C.1 FACTUAL EXPLAINERS

Stability against feature perturbation:

Figure G illustrates the outcomes, demonstrating a continuation of the previously observed trends. Among these trends, we see that there is one clear winner for both datasets. However, PGEXPLAINER performs better than most datasets in both datasets (with discrete and continuous features). On the other hand, the transductive method GNNEXPLAINER performs very poorly in both datasets compared to other methods (i.e., inductive), which further provides evidence that transductive methods are poor in stability for factual explanations.

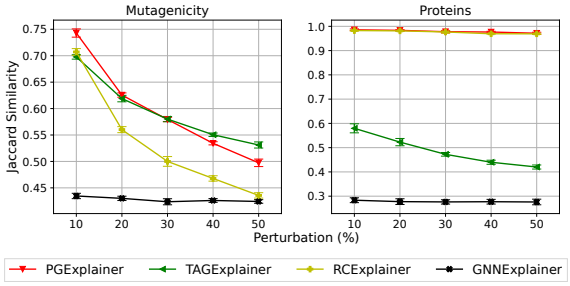


Figure G: Stability of factual explainers against feature perturbation in Jaccard similarity. The stability of explanations drops when the perturbation percentage increases. GNNEXPLAINER (transductive) is the worst method for these two datasets.

Adversarial attack on topology: Figure H demonstrates the performance of four factual methods on these evasion attacks for four datasets. The behavior of the factual methods is similar to the topological noise attack explained in Section 4.2 and the feature perturbations results. When an adversarial attack is compared to random perturbations (Fig. 3), we observe higher deterioration in stability, which is expected since adversarial edge flip attack aims every possible edge in the graph rather than only considering nonexistent edges. Similar to feature perturbation, GNNExplainer (transductive) is affected more by the adversarial attack.

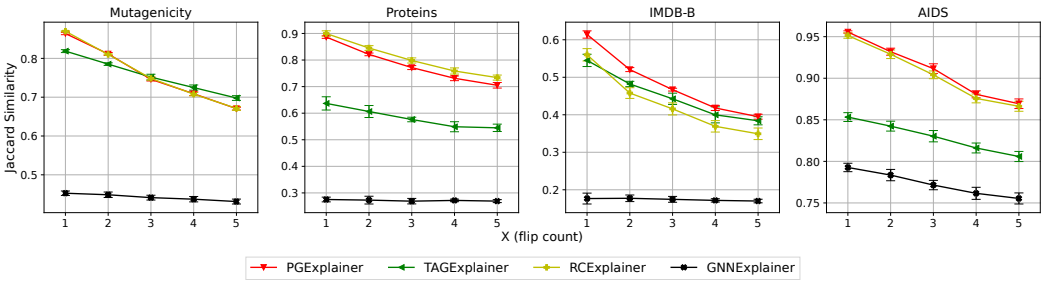


Figure H: Stability of factual explainers against random edge flip in Jaccard similarity. The stability of explanations drops when the flip count increases. GNNEXPLAINER (transductive) is the worst method for these two datasets.

C.2 COUNTERFACTUAL EXPLAINERS

Stability against topological noise: In this section, we investigate the influence of topological noise on datasets on both the performance and generated explanations of counterfactual explainers. For inductive methods (RCEXPLAINER and CLEAR), we utilize explainers trained on noise-free data and only infer on the noisy data. However, for the transductive method CF^2, we retrain the model using the noisy data.

Figure I presents the average Jaccard similarity results, indicating the similarity between the counterfactual graph predicted as an explanation for the original graph and the noisy graphs at varying levels of perturbations. Additionally, Figure J demonstrates the performance of different explainers in terms of sufficiency and size as the degree of noise increases. This provides insights into how these explainers handle higher levels of noise.

RCExplainer outperforms other baselines by a significant margin in terms of size and sufficiency across datasets, as shown in Fig. J. However, the Jaccard similarity between RCExplainer and CF^2 for counterfactual graphs is nearly identical, as shown in Fig. I. CF^2 benefits from its transductive training on noisy graphs. CLEAR's results are not shown for Proteins and Mutagenicity datasets due to scalability issues. In the case of IMDB-B dataset, CLEAR is highly unstable in predicting counterfactual graphs, indicated by a low Jaccard index (Fig. I). Additionally, CLEAR demonstrates high sufficiency but requires a large number of edits, indicating difficulty in finding minimal-edit counterfactuals (Fig. J).

Overall, RCExplainer seems to be the model of choice when topological noise is introduced, and it is significantly faster than CF^2 because it is inductive. Further, it is better than CLEAR as the latter does not scale for larger datasets and is inferior in terms of sufficiency and size as well.

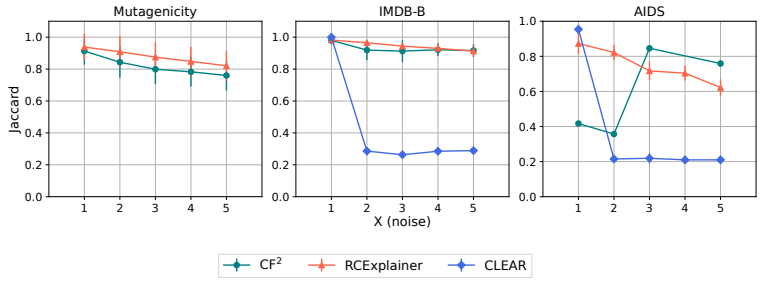
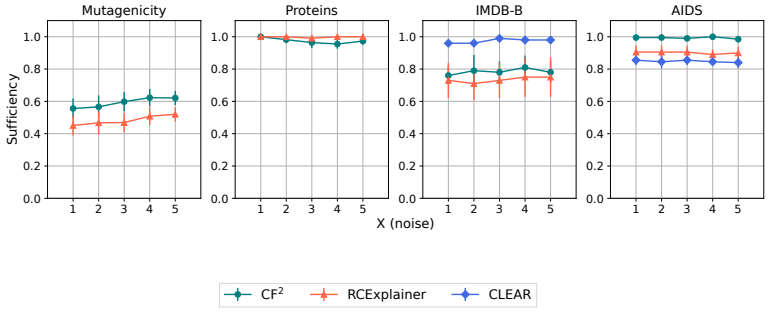
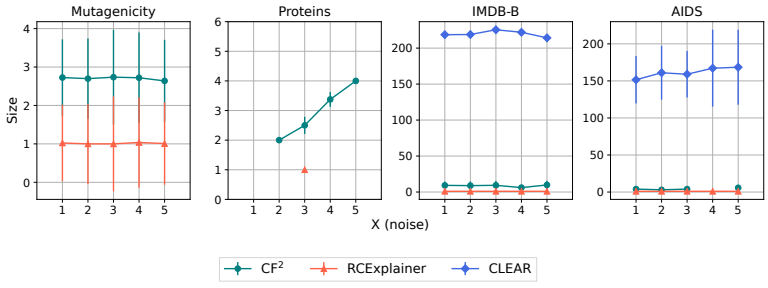


Figure I: Stability of counterfactual explainers against topological noise (Jaccard). We omit CLEAR for Mutagenicity and Proteins as it throws an OOM error for these datasets. The absence of markers representing CF^2 in the Protein dataset's plot indicates that counterfactual graphs were not predicted at the corresponding noise values by the method. Overall, RCExplainer performs best in terms of the Jaccard index.



(a) Sufficiency



(b) Size

Figure J: Performance evaluation of counterfactual explainers against topological noise. We omit CLEAR for Mutagenicity and Proteins as it throws an OOM error for these datasets. RCExplainer is more robust to noise in both metrics: (a) sufficiency and (b) size.

Stability against explainer instances: Table M provides an overview of the stability exhibited among explainer instances trained using three distinct seeds. Notably, we observe a substantial Jaccard index, indicating favorable stability, in the case of RCEXPLAINER and CF^2 explainers. Conversely, CLEAR fails to demonstrate comparable stability. These findings align with the outcomes derived from Table 5. Specifically, when RCEXPLAINER and CF^2 are successful in identifying a counterfactual, the resultant counterfactual graphs are obtained through a small number of perturbations. Consequently, the counterfactual graphs exhibit similarities to the original graph, rendering them akin to one another. However, this trend does not hold for CLEAR, as it necessitates a significantly greater number of perturbations.

Table M: Stability against explainer instances. Note that stability with respect to a graph is computable only if both explainer instances find their counterfactual. "NA" indicates no such graph exists.

	RCExplainer			CF^2			CLEAR		
Dataset / Seeds	$1vs2$	$1vs3$	$2vs3$	$1vs2$	$1vs3$	$2vs3$	$1vs2$	$1vs3$	$2vs3$
Mutagenicity	0.96 ±0.06	0.96 ±0.04	0.98 ±0.03	0.90 ±0.09	0.89 ±0.1	0.89 ±0.11	OOM	OOM	OOM
Proteins	0.95 ±0.0	0.94 ±0.0	0.90 ±0.0	NA	NA	NA	OOM	OOM	OOM
Mutag	0.98 ±0.03	0.98 ±0.03	1.0 ±0.0	1.0 ±0.0	1.0 ±0.0	1.0 ±0.0	0.55 ±0.01	0.53 ±0.01	0.54 ±0.02
IMDB-B	0.99 ±0.01	1.0 ±0.0	0.99 ±0.01	0.96 ±0.05	0.95 ±0.06	0.94 ±0.07	0.28 ±0.0	0.27 ±0.0	0.28 ±0.0
AIDS	0.84 ±0.04	0.96 ±0.06	0.84 ±0.04	NA	1.0 ±0.0	NA	0.19 ±0.02	0.20 ±0.03	0.19 ±0.04
ogbg-molhiv	0.99 ±0.03	0.99 ±0.04	0.99 ±0.04	0.801 ±0.149	0.764 ±0.14	0.784 ±0.144	OOM	OOM	OOM

Similarly, as additional results, we also present the variations in terms of sufficiency (Table N) and the explanation size (Table O) produced by the methods using three different seeds. In terms of sufficiency, the methods show stability while varying the seeds. However, the results are drastically different for explanation size. Table O present the results. We observe that RCEXPLAINER is consistently the most stable method while CF^2 is worse. The worst stability is shown by CLEAR and this observation is consistent with the previous results.

Table N: Stability of sufficiency produced by counterfactual explainers against the explainer instances (seeds). The best explainers for each dataset (row) are highlighted in gray, yellow and cyan shading for seeds 1, 2, and 3, respectively. OOM indicates that the explainer threw an out-of-memory error.

	RCExplainer			CF^2			CLEAR		
Dataset / Seeds	1	2	3	1	2	3	1	2	3
Mutagenicity	0.4 ±0.06	0.40 ±0.05	0.41 ±0.05	0.50 ±0.05	0.49 ±0.06	0.52 ±0.05	OOM	OOM	OOM
Proteins	0.96 ±0.02	0.96 ±0.02	0.96 ±0.02	1.0 ±0.0	1.0 ±0.0	0.98 ±0.02	OOM	OOM	OOM
Mutag	0.4 ±0.12	0.6 ±0.12	0.55 ±0.1	0.9 ±0.12	0.85 ±0.2	0.9 ±0.12	0.55 ±0.1	0.55 ±0.1	0.65 ±0.12
IMDB-B	0.72 ±0.11	0.72 ±0.11	0.72 ±0.11	0.81 ±0.07	0.82 ±0.08	0.81 ±0.07	0.96 ±0.02	0.96 ±0.02	0.96 ±0.02
AIDS	0.91 ±0.04	0.91 ±0.04	0.91 ±0.04	0.98 ±0.02	1.0 ±0.0	0.99 ±0.01	0.84 ±0.03	0.82 ±0.03	0.83 ±0.04
ogbg-molhiv	0.90 ±0.02	0.88 ±0.01	0.90 ±0.01	0.96 ±0.01	0.97 ±0.00	0.96 ±0.00	OOM	OOM	OOM

Table O: Stability of *explanation size* produced by explainers against the explainer instances (seeds). NA indicates the inability to find a counterfactual. OOM indicates that the explainer threw out-of-memory error. The best explainers for each dataset (row) are highlighted in gray, yellow and cyan shading for seeds 1, 2, and 3, respectively.

	RCExplainer			CF^2			CLEAR		
Dataset / Seeds	1	2	3	1	2	3	1	2	3
Mutagenicity	1.01 ±0.19	1.0 ±0.0	1.25 ±0.0	2.78 ±0.98	2.85 ±1.07	2.95 ±1.37	OOM	OOM	OOM
Proteins	1.0 ±0.0	1.0 ±0.0	1.0 ±0.0	NA	NA	3.0 ±0.0	OOM	OOM	OOM
Mutag	1.1 ±0.22	1.0 ±0.0	1.0 ±0.0	1.0 ±0.0	1.25 ±0.35	1.0 ±0.0	17.15 ±1.62	15.6 ±1.86	19.05 ±1.31
IMDB-B	1.0 ±0.0	1.0 ±0.0	1.0 ±0.0	8.57 ±4.99	8.29 ±4.50	9.01 ±5.58	218.62 ±0.0	182.25 ±0.0	181.38 ±0.0
AIDS	1.0 ±0.0	1.0 ±0.0	1.0 ±0.0	5.25 ±0.35	NA	6.0 ±0.0	164.95 ±47.93	162.32 ±45.70	185.29 ±78.92
ogbg-molhiv	1.0 ±0.0	1.0 ±0.0	1.02 ±0.42	10.45 ±4.43	9.69 ±4.18	10.24 ±4.87	OOM	OOM	OOM

Stability against GNN architectures: Table P shows the stability of the explainers across different GNN architectures. Similar to our factual setting (Table 8), we assess the stability by computing the Jaccard coefficient between the explained predictions of the indicated GNN architecture and the default GCN model. Unsurprisingly, the stability of the explainer highly depends on the dataset.

RCEXPLAINER is also the most stable among all the explainers, and the produced high values indicate that the method is agnostic towards the variations in different message aggregating schemes of the architectures.

We further look into the stability of the counterfactual methods in terms of sufficiency (Table Q) and the explanation size (Table R) across different GNN architectures. The sufficiency results (Table Q) show large variations produced by the same method on the same dataset due to the different architectures and message passing schemes. For instance, RCEXPLAINER produces sufficiency of .10 and .93 on the AIDS dataset for GAT and GIN, respectively. In terms of explanation size(Table R), RCEXPLAINER is stable against different GNN architectures. However, consistent with previous stability results, CF^2 is more unstable than RCEXPLAINER and the worst stability is shown by CLEAR.

Table P: Stability of counterfactual explainers against the GNN architecture. We report the Jaccard coefficient of explanations obtained for GAT, GIN and GRAPHSAGE against the explanation provided over GCN. The higher the Jaccard, the more is the stability. The best explained for each dataset (row) are highlighted in gray, yellow and cyan shading for architectures GAT, GIN, and GRAPHSAGE, respectively. GRAPHSAGE is denoted by SAGE. NA indicates one or both of the architectures were unable to identify a counterfactual for the graphs. OOM indicates that the explainer threw an out-of-memory error.

	RCEXplainer			CF^2			CLEAR		
Dataset / Architecture	GAT	GIN	SAGE	GAT	GIN	SAGE	GAT	GIN	SAGE
Mutagenicity	0.95 ±0.05	0.94 ±0.06	0.95 ±0.03	0.79 ±0.13	0.75 ±0.16	0.84 ±0.10	OOM	OOM	OOM
Proteins	0.88 ±0.0	NA	0.88 ±0.0	NA	NA	NA	OOM	OOM	OOM
Mutag	0.94 ±0.0	NA	0.90 ±0.02	NA	NA	NA	0.86 ±0.0	NA	0.72 ±0.04
IMDB-B	0.99 ±0.01	0.98 ±0.0	0.98 ±0.01	NA	0.93 ±0.0	NA	0.60 ±0.0	0.70 ±0.0	0.76 ±0.0
AIDS	0.89 ±0.03	NA	NA	0.74 ±0.0	0.73 ±0.11	0.72 ±0.12	0.25 ±0.04	0.54 ±0.04	0.66 ±0.04
ogbg-molhiv	0.96 ±0.02	0.96 ±0.01	0.96 ±0.02	0.63 ±0.12	0.13 ±0.14	0.61 ±0.16	OOM	OOM	OOM

Table Q: Stability in terms of *sufficiency* of counterfactual explainers against the GNN architectures. OOM indicates that the explainer threw out-of-memory error. The best explainers for each dataset (row) are highlighted in gray, yellow, cyan, and pink shading for GCN, GAT, GIN, SAGE, respectively. RCExplainer outperforms other baselines on a majority of the datasets and architectures. CLEAR also is stable in terms of sufficiency but has a much larger explanation size compared to other baselines(Refer Table R).

	RCEXplainer				CF^2				CLEAR			
Dataset / Architecture	GCN	GAT	GIN	SAGE	GCN	GAT	GIN	SAGE	GCN	GAT	GIN	SAGE
Mutagenicity	0.4 ±0.06	0.38 ±0.04	0.6 ±0.04	0.59 ±0.06	0.50 ±0.05	0.64 ±0.04	0.57 ±0.08	0.62 ±0.03	OOM	OOM	OOM	OOM
Proteins	0.96 ±0.02	0.88 ±0.04	0.3 ±0.05	0.46 ±0.08	1.0 ±0.0	1.0 ±0.0	0.76 ±0.06	0.79 ±0.02	OOM	OOM	OOM	OOM
Mutag	0.4 ±0.12	0.7 ±0.19	1.0 ±0.0	0.45 ±0.19	0.9 ±0.12	0.9 ±0.12	0.45 ±0.33	0.7 ±0.19	0.55 ±0.1	0.8 ±0.19	1.0 ±0.0	0.05 ±0.1
IMDB-B	0.72 ±0.11	0.89 ±0.02	0.54 ±0.06	0.39 ±0.04	0.81 ±0.07	1.0 ±0.0	0.98 ±0.02	0.99 ±0.02	0.96 ±0.02	0.68 ±0.08	0.22 ±0.11	0.32 ±0.11
AIDS	0.91 ±0.04	0.10 ±0.04	0.93 ±0.03	0.86 ±0.05	0.98 ±0.02	0.92 ±0.04	0.96 ±0.01	0.96 ±0.02	0.84 ±0.03	0.80 ±0.04	0.74 ±0.04	0.84 ±0.02
ogbg-molhiv	0.90 ±0.02	0.80 ±0.01	0.56 ±0.01	0.20 ±0.01	0.96 ±0.01	0.96 ±0.01	0.90 ±0.01	0.59 ±0.01	OOM	OOM	OOM	OOM

Table R: Stability of *explanation size* produced by explainers against different GNN architectures. OOM indicates that the explainer threw out-of-memory error. NA indicates that the explainer could not identify a counterfactual for the graphs. The best explainers for each dataset (row) are highlighted in gray, yellow, cyan, and pink shading for GCN, GAT, GIN, SAGE respectively. RCExplainer outperforms other counterfactual baselines.

	RCEXplainer				CF^2				CLEAR			
Dataset / Architecture	GCN	GAT	GIN	SAGE	GCN	GAT	GIN	SAGE	GCN	GAT	GIN	SAGE
Mutagenicity	1.01 ± 0.18	1.33 ± 2.06	1.0 ± 0.0	1.03 ± 0.29	2.81 ± 1.12	3.90 ± 2.08	5.93 ± 2.97	3.32 ± 1.61	OOM	OOM	OOM	OOM
Proteins	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.97 ± 7.75	3.5 ± 0.5	4.0 ± 0.0	2.62 ± 1.22	2.04 ± 1.3	OOM	OOM	OOM	OOM
Mutag	1.0 ± 0.0	NA	NA	1.0 ± 0.0	2.0 ± 1.07	1.0 ± 0.0	20.36 ± 3.94	1.4 ± 0.8	38.12 ± 3.41	37.4 ± 3.61	NA	45.76 ± 7.94
IMDB-B	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	7.78 ± 3.98	NA	6.0 ± 0.0	7.17 ± 3.89	424.0 ± 192.26	441.53 ± 70.97	475.42 ± 75.76	350.45 ± 86.62
AIDS	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	NA	4.21 ± 3.07	2.0 ± 0.0	3.22 ± 2.15	222.84 ± 47.02	667.01 ± 94.35	212.77 ± 43.18	201.09 ± 38.85

Stability to feature noise: Table S presents the impact of feature noise on counterfactual explanations in Mutag and Mutagenicity. We observe that CF^2 and CLEAR are markedly more stable than

RCEXPLAINER in Mutag. This outcome is not surprising, considering that RCEXPLAINER exclusively addresses topological perturbations, while both CF^2 and CLEAR accommodate perturbations encompassing both topology and features. In Mutaganecity, RCEXPLAINER exhibits slightly higher stability than CF^2.

Table S: Stability of counterfactual explainers against feature perturbation on "Mutag" and "Mutagenecity" datasets. We do not report results for CLEAR on Mutagenicity since it runs out of GPU memory.

(a) Mutag

	RCExplainer			CF^2			CLEAR		
Noise% / Metric	Sufficiency	Size	Jaccard	Sufficiency	Size	Jaccard	Sufficiency	Size	Jaccard
0 (no noise)	0.4 ± 0.12	1.10 ± 0.22	1.0 ± 0.0	0.90 ± 0.12	1.0 ± 0.0	1.0 ± 0.0	0.55 ± 0.1	17.15 ± 1.62	1.0 ± 0.0
10	1.0 ± 0.0	NA	NA	0.6 ± 0.2	2.17 ± 0.31	0.19 ± 0.0	0.55 ± 0.1	16.35 ± 1.68	0.98 ± 0.01
20	0.75 ± 0.22	1.0 ± 0.0	NA	0.25 ± 0.16	1.95 ± 0.80	NA	0.55 ± 0.1	18.1 ± 1.94	0.55 ± 0.01
30	1.0 ± 0.0	NA	NA	0.6 ± 0.3	1.0 ± 0.0	0.29 ± 0.0	0.55 ± 0.1	19.1 ± 2.91	0.57 ± 0.02
40	1.0 ± 0.0	NA	NA	0.6 ± 0.2	2.8 ± 0.0	0.29 ± 0.0	0.55 ± 0.1	14.7 ± 1.78	0.58 ± 0.02
50	1.0 ± 0.0	NA	NA	0.8 ± 0.1	1.5 ± 0.0	NA	0.55 ± 0.1	16.05 ± 2.48	0.54 ± 0.02

(b) Mutagenicity

	RCEXplainer			CF^2		
Noise% / Metric	Sufficiency	Size	Jaccard	Sufficiency	Size	Jaccard
0 (no noise)	0.4 ± 0.06	1.01 ± 0.19	1.0 ± 0.0	0.50 ± 0.05	2.78 ± 0.98	1.0 ± 0.0
10	0.43 ± 0.04	1.01 ± 0.19	0.96 ± 0.04	0.49 ± 0.04	2.11 ± 1.06	0.92 ± 0.06
20	0.47 ± 0.06	1.0 ± 0.0	0.95 ± 0.04	0.53 ± 0.06	1.73 ± 0.94	0.86 ± 0.08
30	0.50 ± 0.04	1.0 ± 0.0	0.94 ± 0.04	0.61 ± 0.04	1.68 ± 0.91	0.86 ± 0.09
40	0.52 ± 0.05	1.0 ± 0.0	0.93 ± 0.06	0.54 ± 0.05	1.47 ± 0.73	0.87 ± 0.09
50	0.49 ± 0.05	1.0 ± 0.0	0.93 ± 0.06	0.67 ± 0.01	1.54 ± 1.06	0.86 ± 0.08

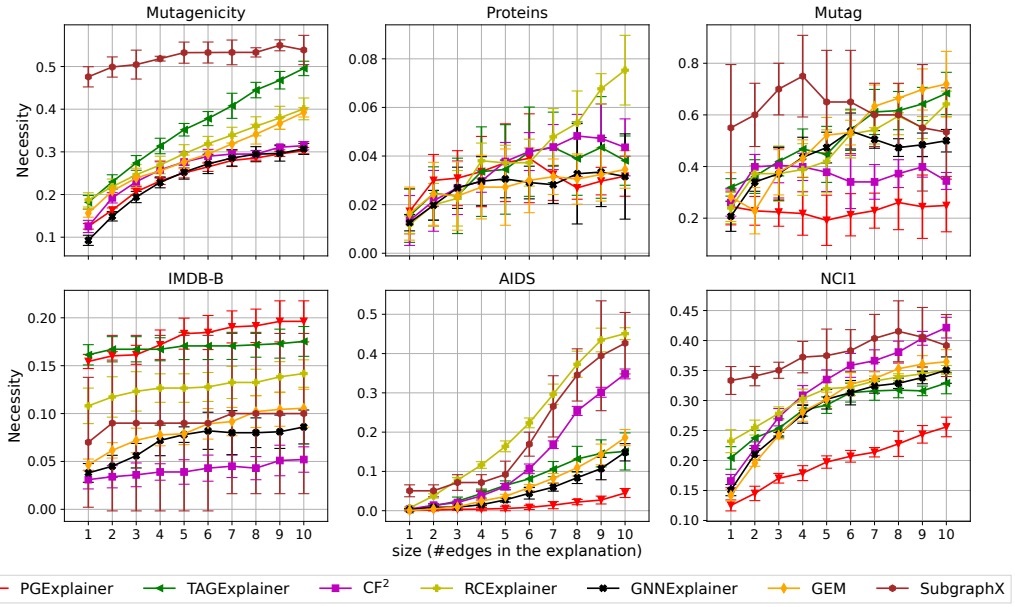


Figure K: Necessity of various factual explainers against the explanation size. The necessity increases with the removal of the explanations.

D NECESSITY: RESULTS FOR SEC. 4.3

For necessity, we remove explanations from graphs and measure the ratio of graphs for which the GNN prediction on the residual graph is flipped. We expect that removing more explanations will lead to more necessity; however, we do not necessarily expect necessity to be higher because our factual explainers are not trained to make the residual graph counterfactual. Fig. K presents the necessity performance on six datasets for all factual methods with varying explanation sizes from 1 to 10. The trend aligns with our expectations, as the removal of more explanations increases the necessity. The value of necessity varies between datasets. Proteins and IMDB-B datasets have graphs with larger sizes (in terms of the number of edges), which aligns with the small necessity score. On the other hand, datasets with relatively smaller graphs have higher necessity scores. Note that our factual methods do not optimize residual graphs to be counterfactuals; this might be another reason for the low values.

E REPRODUCIBILITY: RESULTS FOR SEC. 4.3

Reproducibility can be measured two different ways. (1) Retraining using only explanation graphs called Reproducibility^{+}, (2) retraining using only residual graphs called Reproducibility^{-}. Both metrics is a ratio of an GNN accuracy compared to the original GNN accuracy. We provide the math definitions in Table 3. In our figures, we separated SubgraphX to an independent table, because we could only obtain explanations of test graphs for SubgraphX (refer to our disclaimer in Sec. A.4)

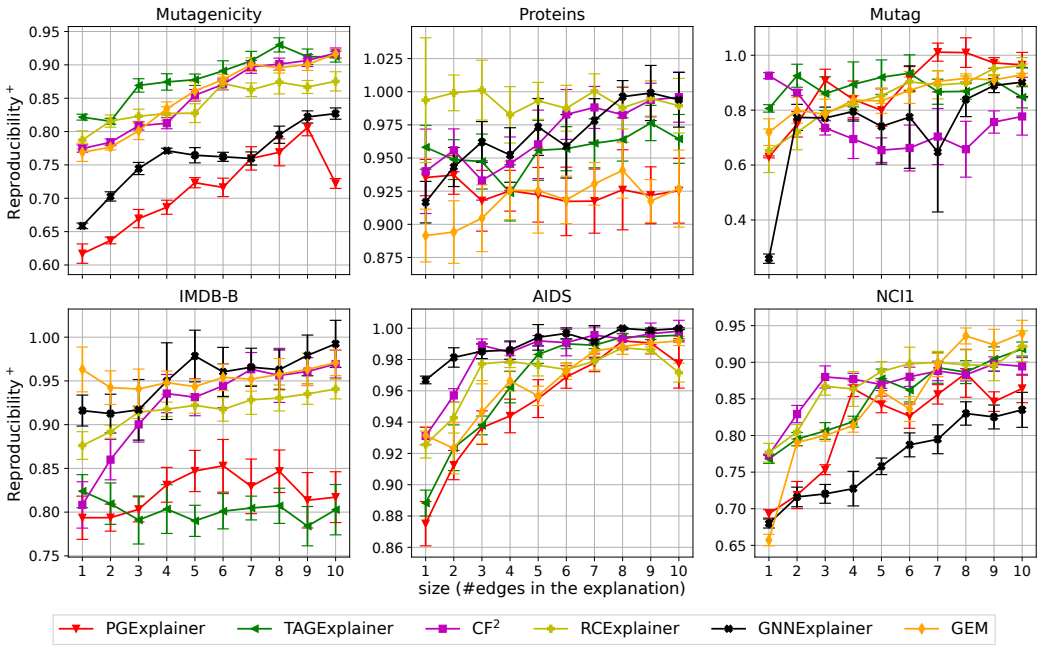


Figure L: Reproducibility^{+} of factual explainers against size. Performance tends to rise with more edges in the explanations; however, a small number of edges does not guarantee a performance of 1.0.

	Size (#edges in the explanation)									
Datasets	1	2	3	4	5	6	7	8	9	10
Mutagenicity	1.11 ± 0.02	1.07 ± 0.03	1.08 ± 0.02	1.05 ± 0.02	1.05 ± 0.02	1.05 ± 0.02	1.03 ± 0.02	1.01 ± 0.02	0.97 ± 0.02	1.02 ± 0.02
Mutag	0.86 ± 0.43	0.43 ± 0.53	0.76 ± 0.5	1.08 ± 0.0	0.11 ± 0.32	0.22 ± 0.43	1.08 ± 0.0	0.86 ± 0.43	0.97 ± 0.32	1.08 ± 0.0
IMDB-B	1.07 ± 0.16	1.1 ± 0.17	1.05 ± 0.15	1.08 ± 0.06	1.09 ± 0.08	1.07 ± 0.08	1.05 ± 0.06	1.06 ± 0.06	1.02 ± 0.05	1.05 ± 0.09
AIDS	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.01	1.0 ± 0.01	0.99 ± 0.01	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
NCI1	1.05 ± 0.04	1.09 ± 0.03	1.09 ± 0.03	1.08 ± 0.02	1.09 ± 0.02	1.09 ± 0.02	1.07 ± 0.03	1.07 ± 0.03	1.05 ± 0.03	1.04 ± 0.04

Table T: Reproducibility^{+} in SubgraphX. Since we use small number of graphs for SubgraphX, the variance of the results are very high, thus unreliable.

Figure L and Table T illustrate the Reproducibility^{+} performance of seven factual methods against the size of explanations for six datasets. Reproducibility increases with more edges in the explanations for the most cases, as expected. However, reaching a score of 1.0 is challenging even when selecting the most crucial edges. This suggests that explanations do not capture the full picture for GNN predictions.

Figure M and Table U illustrate the Reproducibility^{-} performance of seven factual methods against the size of explanations for six datasets. Reproducibility remains high even when the most crucial edges from the graphs are removed. This demonstrates that the explainers hardly capture the real cause of the GNN predictions.

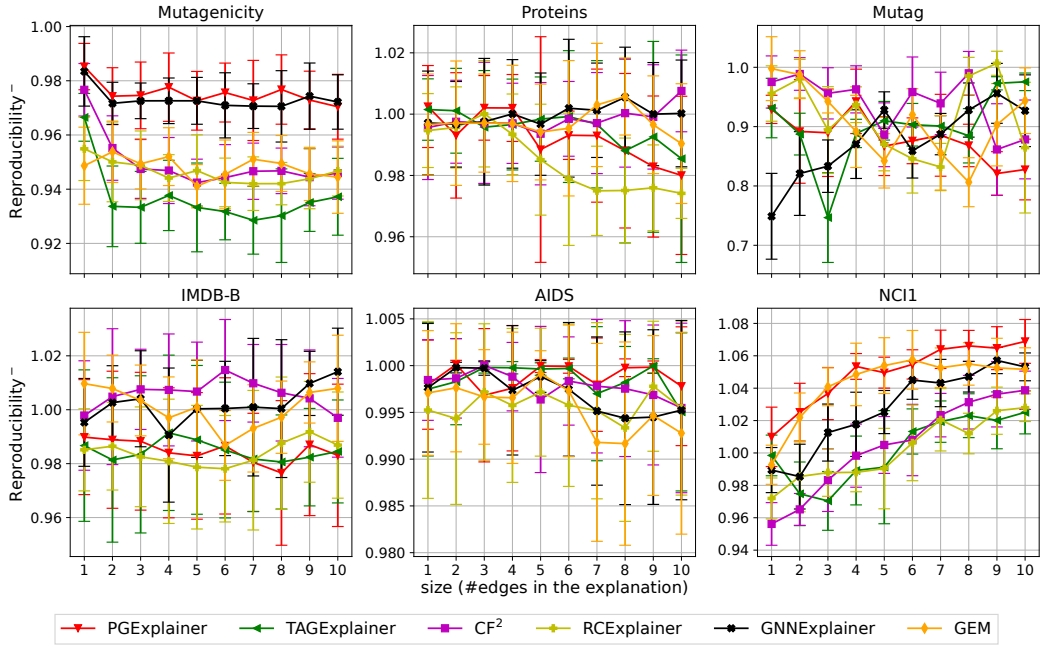


Figure M: Reproducibility^{-} of factual explainers against size. Even when the most crucial edges are taken out, the performance still remains close to 1.0.

	Size (#edges in the explanation)									
Datasets	1	2	3	4	5	6	7	8	9	10
Mutagenicity	1.06 ± 0.03	1.01 ± 0.02	1.01 ± 0.02	1.01 ± 0.03	1.0 ± 0.03	1.0 ± 0.03	1.01 ± 0.03	0.99 ± 0.03	1.0 ± 0.03	0.99 ± 0.03
Mutag	1.08 ± 0.0	1.08 ± 0.0	0.97 ± 0.32	1.08 ± 0.0	0.97 ± 0.32	0.97 ± 0.32	1.08 ± 0.0	1.08 ± 0.0	1.08 ± 0.0	1.08 ± 0.0
IMDB-B	0.82 ± 0.28	0.96 ± 0.14	0.86 ± 0.27	0.85 ± 0.28	0.86 ± 0.28	0.82 ± 0.28	0.84 ± 0.27	0.85 ± 0.28	0.85 ± 0.28	0.86 ± 0.28
AIDS	1.0 ± 0.0	1.0 ± 0.01	1.0 ± 0.01	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.01	0.99 ± 0.01	1.0 ± 0.01	1.0 ± 0.01	1.0 ± 0.01
NCI1	0.9 ± 0.06	0.91 ± 0.05	0.89 ± 0.07	0.93 ± 0.07	0.92 ± 0.07	0.92 ± 0.06	0.91 ± 0.05	0.9 ± 0.04	0.91 ± 0.05	0.92 ± 0.05

Table U: Reproducibility^{-} in SubgraphX. Since we use small number of graphs for SubgraphX, the variance of the results are very high, thus unreliable.

F VISUALIZATION OF EXPLANATIONS

In Figs. N and O, we engage in a visual analysis of the explanations provided by various GNN explainers. The graphs presented in these figures represent mutagenic molecules sourced from the Mutag dataset. Several insights emerge from this analysis.

Factual: The mutagenic attribute of the molecule in Fig. N stems from the presence of the NO_2 group attached to the benzene ringYing et al. (2019b); Debnath et al. (1991). As a result, the optimal explanation entails pinpointing this specific benzene ring in conjunction with the NO_2 group. Notably, we observe that while certain explainers identify fragments of this subgraph, with CF^2 achieving the highest overlap, many also highlight bonds originating from regions outside the authentic explanatory context. Adding to the intrigue, the explanation offered by RCEXPLAINER stands out due to its compactness, resulting in commendable statistical performance. However, this succinct explanation

lacks meaning in the eyes of a domain expert. Consequently, a pressing need arises for real-world datasets endowed with ground truth explanations, a resource that the current field unfortunately lacks.
Counterfactuals: Fig. O illustrates two molecules, with Molecule 1 (top row) being identical to the one shown in Fig. N. The optimal explanation involves eliminating the NO_2 component, a task accomplished solely by CF^2 in the case of Molecule 1. While the remaining explanation methods can indeed alter the GNN prediction by implementing the changes described in Figure O, two critical insights emerge. First, statistically, RCEXPLAINER is considered a better explanation than CF^2 since its size is 1 compared to 3 of CF^2. However, our interaction with multiple chemists clearly indicated their preference towards CF^2 since eliminates the entire NO_2 group. Second, chemically infeasible explanations are common as evident from CLEAR for both molecules and CF^2 in molecule 2. Both fail to adhere to valency rules, a behavior also noted in Sec. 4.4.

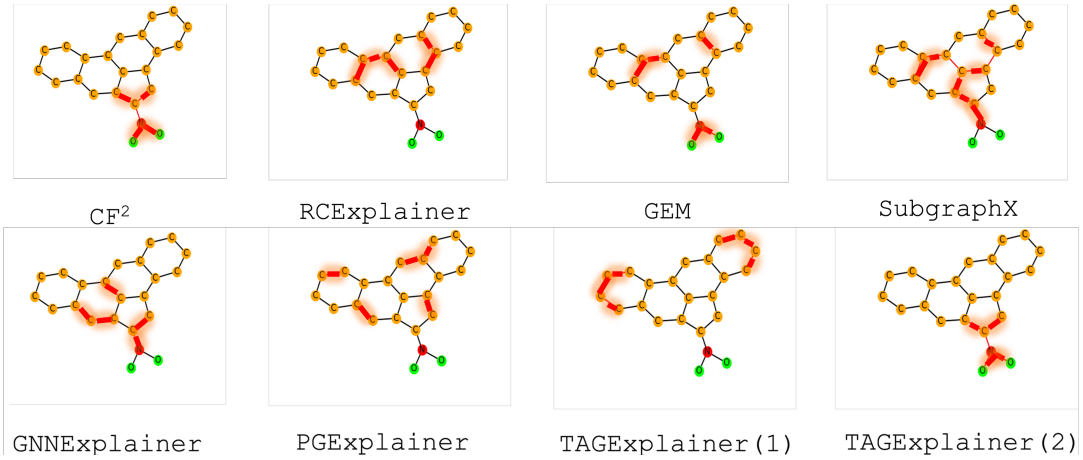


Figure N: Visualization of factual explanations on a mutagenic molecule from the Mutag dataset. The explanations contain the edges highlighted in red.

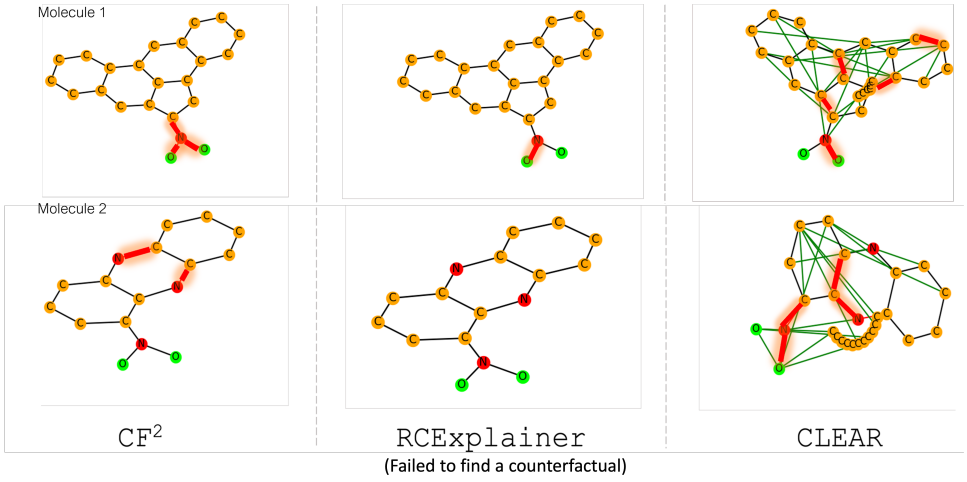


Figure O: Visualization of counterfactual explanations on Mutag dataset. Edge additions and deletions are represented by green and red colors respectively.

G ADDITIONAL EXPERIMENTS ON SPARSITY METRIC

Counterfactual explainers: Sparsity is defined as the proportion of edges from the graph that is retained in the counter-factual Yuan et al. (2022); a value close to 1 is desired. In node classification, we compute this proportion for edges in the $\mathcal{N}_v^\ell$, i.e., the ℓ-hop neighbourhood of the target node v. We present the results on sparsity metric on node and graph classification tasks in Table V and W,

respectively. For node classification, we see CF-GNNEXPLAINER continues to outperform (Table V). The results are consistent with our earlier results in Table 6. Similarly, Table W shows that RCExplainer continues to outperform in the case of graph classification (earlier results show a similar trend in Table 5).

Factual explainers: For experiments with factual explainers, we report results of the necessity metric on varying degrees of sparsity (Recall Fig. K). Note that size acts as a proxy for sparsity in this case. This is because sparsity only involves the normalized size where it is normalized by the number of total edges. So sparsity can be obtained by normalizing all perturbation sizes in the plots to get the sparsity metric. For counterfactual explainers, we do not supply explanation size as a parameter. Hence, computing the sparsity of the predicted counterfactual becomes relevant.

Table V: Results on sparsity of counterfactual explainers for node classification. CF-GNNExplainer consistently produces the best results that are shown in gray.

Method/Dataset	Tree-Cycles	Tree-Grid	BA-Shapes
CF-GNNEXPLAINER	0.93 ±0.02	0.95 ±0.03	0.99 ±0.00
CF^2	0.52 ±0.14	0.58 ±0.14	0.99 ±0.00

Table W: Results on sparsity of counterfactual explainers for graph classification. Best results are shown in gray. RCEXPLAINER consistently outperforms the other methods.

Method/Dataset	Mutagenicity	Mutag	Proteins	AIDS	IMDB-B	ogbg-molhiv
RCExplainer	0.96 ±0.00	0.94 ±0.01	0.94 ±0.02	0.91 ±0.00	0.98 ±0.00	0.96 +- 0.00
CF^2	0.90 ±0.01	0.94 ±0.0	NA	0.99 ±0.01	0.89 ±0.04	0.62 ±0.05
CLEAR	OOM	0.88 ±0.07	OOM	0.66 ±0.04	0.87 ±0.06	OOM

H TAGEXPLAINER VARIANTS

TAGExplainer has two stages. We define TAGExplainer (1) as when we apply only the first stage and get explanations, whereas TAGExplainer (2) applies both stages. Figure P compares performance of these two variants.

I FACTUAL EXPLAINERS ON OGBG-MOLHIV

Figure R demonstrates a new dataset OGBG-Molhiv for three factual explainers. On this dataset, three factual methods are close to each other in terms of sufficiency performance.

J EXISTING BENCHMARKING STUDIES ON GNN EXPLAINABILITY

GraphFrameX Amara et al. (2022) and GraphXAI Agarwal et al. (2023) represent two notable benchmarking studies. While both investigations have contributed valuable insights into GNN explainers, certain unresolved investigative aspects persist.

- **Inclusion of counterfactual explainability:** GraphFrameX and GraphXAI have focused on factual explainers for GNNs. Prado-Romero et al. (2023) has discussed methods and challenges, but benchmarking on counterfactual explainers remains underexplored.
- **Achieving Comprehensive coverage:** Existing literature encompasses seven perturbation-based factual explainers. However, GraphFrameX and GraphXAI collectively assess only GnnExplainer Ying et al. (2019b), PGExplainer Luo et al. (2020), and SubgraphX Yuan et al. (2021).
- **Empirical investigations:** How susceptible are the explanations to topological noise, variations in GNN architectures, or optimization stochasticity? Do the counterfactual explanations provided align with the structural and functional integrity of the underlying domain? To what extent do these explainers elucidate the GNN model as opposed to the underlying data? Are there standout explainers that consistently outperform others in terms of performance? These are critical empirical inquiries that necessitate attention.

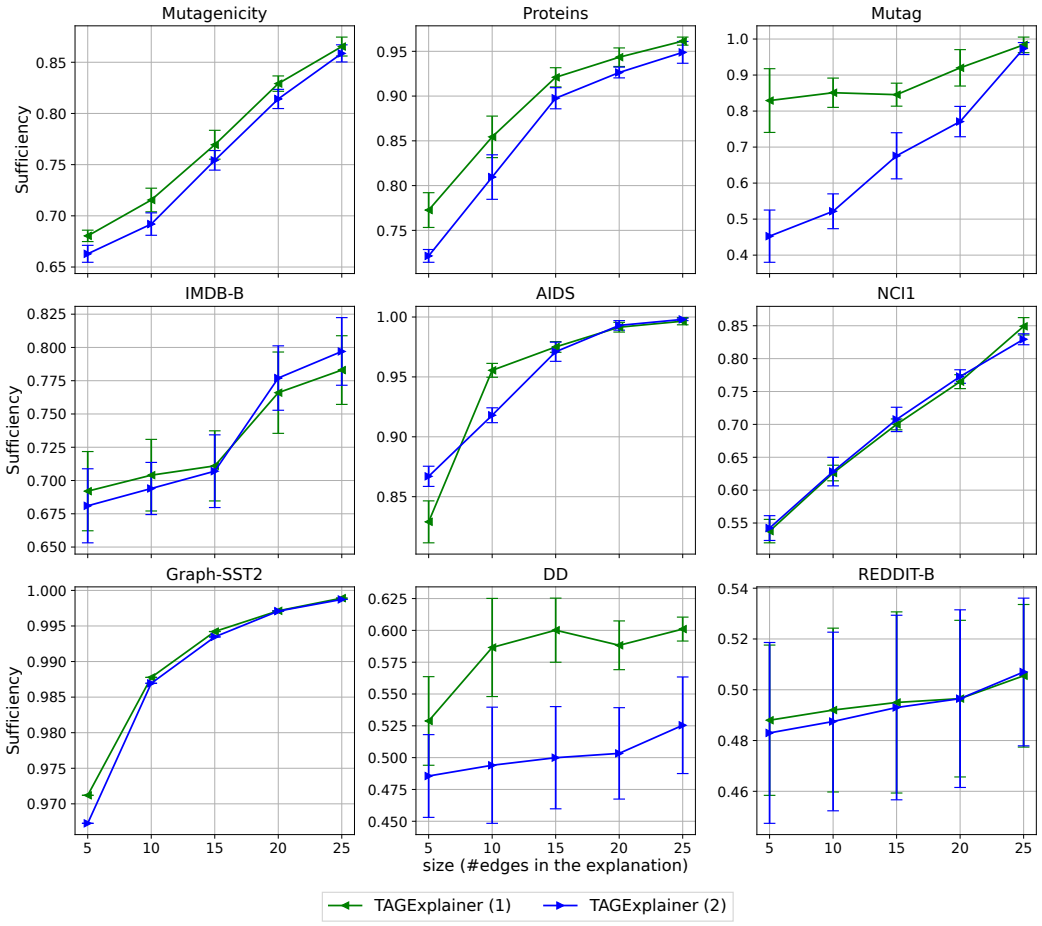


Figure P: Sufficiency of TAGExplainer variants against size. Applying the second stage does not help much for TAGExplainer.

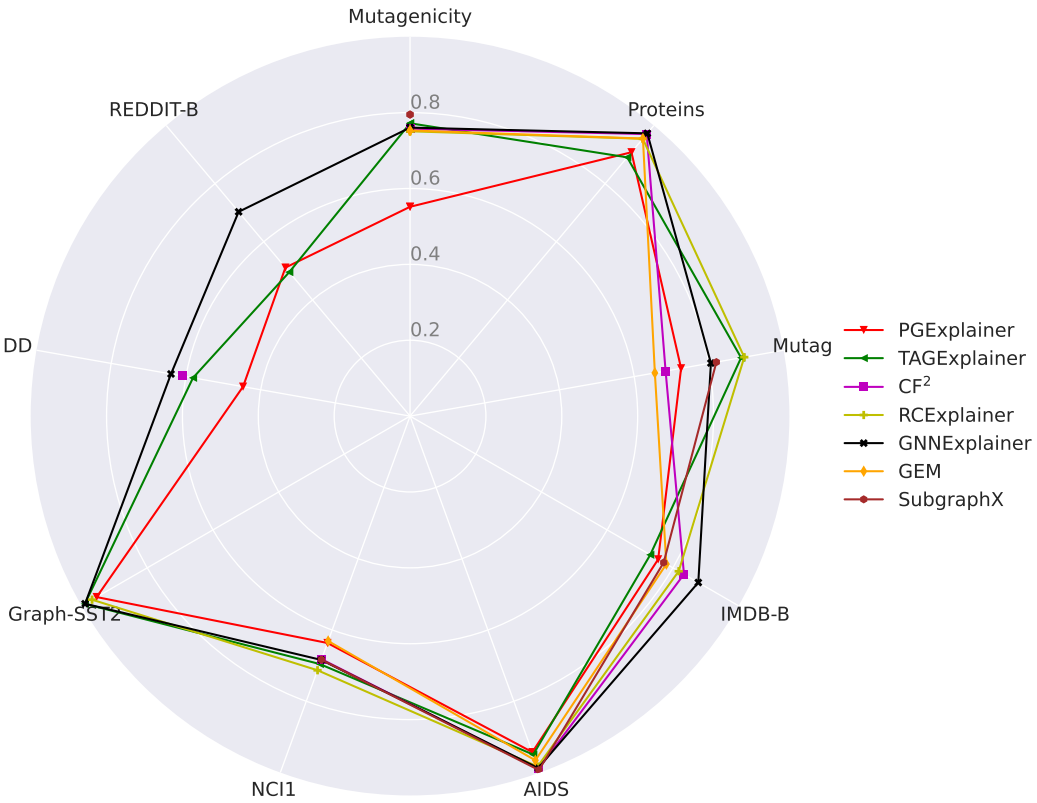


Figure Q: Spiderplot for sufficiency performance averaged for different sizes of explanations. Even though there is no clear winner method, GNNEXPLAINER and RCEXPLAINER appear among the top performers in the majority of the datasets. We omit those methods for a dataset that throw an out-of-memory (OOM) error and are not scalable.

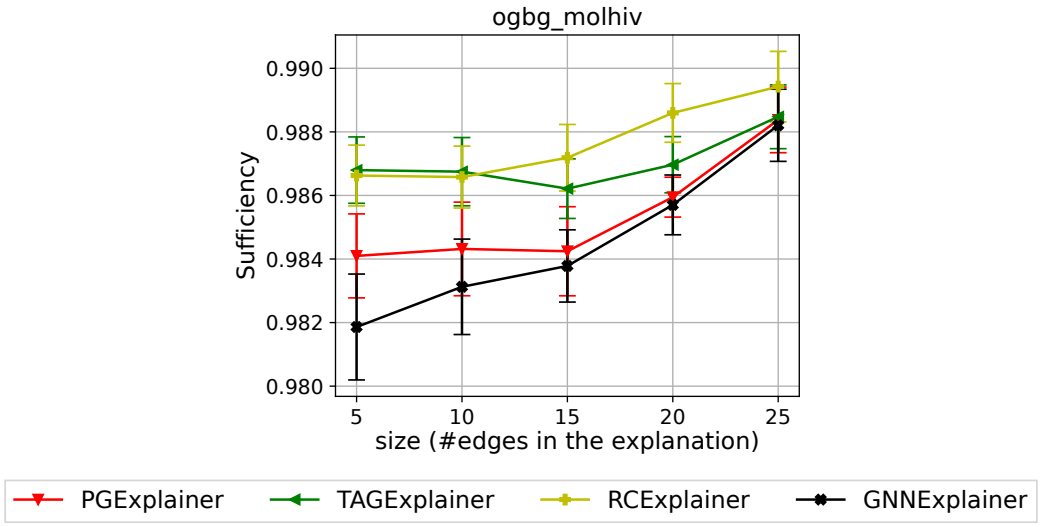


Figure R: Sufficiency of the factual explainers against the explanation size for ogbg-molhiv dataset.

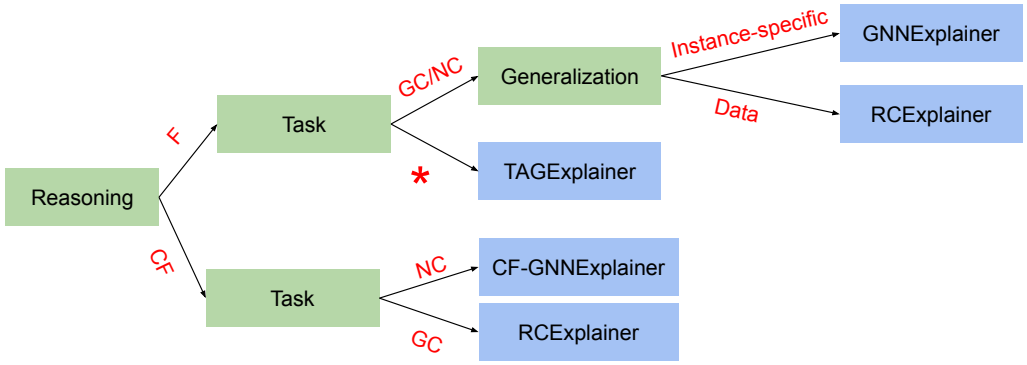


Figure S: Flowchart of our recommendations of explainers in different scenarios. Note that "F", "CF", "NC", "GC", and "*" denote factual, counterfactual, node classification, graph classification, and generalized tasks respectively.

K RECOMMENDATIONS IN PRACTICE

The choice of the explainer depends on various factors and we make the following recommendations.

- For counter-factual reasoning, we recommend RCExplainer for graph classification and CF-GNNExplainer for node classification.

- For factual reasoning, if the goal is to do node or graph classification, we need to first decide if we need an inductive reasoner or transductive. While an inductive reasoner is more suitable we want to generalize to large volumes of unseen graphs, transductive is suitable for explaining single instances. In case of inductive, we recommend RCExplainer, while for transductive GNNExplainer stands out as the method of choice. These two algorithms had the highest sufficiency on average. In addition, RCExplainer displayed stability in the face of noise injection.
  - For different task generalization beyond node and graph classification, one can consider using TAGExplainer.
  - In high-stakes applications, we recommend RCExplainer due to consistent results across different runs and robustness to noise.

- For both types of explainers, the transductive methods are slow. So, if the dataset is large, it is always better to use an inductive explainer over transductive ones.

While the above is a guideline, we emphasize that there is no one-size-fits-all benchmark for selecting the ideal explainer. The choice depends on the characteristics of the application in hand and/or the dataset. The above flowchart takes these factors into account to streamline the decision process. We have now added a flowchart to help the user in selecting the most appropriate. This recommendation has now also been added as a flowchart (Fig. S).