Bagel: A Benchmark for Assessing Graph Neural Network Explanations

Mandeep Rathee[1], Thorben Funke[1], Avishek Anand[2], and Megha Khosla[2]

[1] L3S Research Center, Hannover, Germany
[2] {rathee,tfunke}@l3s.de
Delft University of Technology (TU Delft), Netherlands
{avishek.anand,M.Khosla}@tudelft.nl

Abstract. Evaluating interpretability approaches for graph neural networks (GNN) specifically is known to be challenging due to the lack of a commonly accepted benchmark. Given a GNN model, several interpretability approaches exist to explain GNN models with diverse (sometimes conflicting) evaluation methodologies. In this paper, we propose a benchmark for evaluating the explainability approaches for GNNs called Bagel. In Bagel, we first propose four diverse GNN explanation evaluation regimes – 1) *faithfulness*, 2) *sparsity*, 3) *correctness*, and 4) *plausibility*. We reconcile multiple evaluation metrics in the existing literature and cover diverse notions for a holistic evaluation. Our graph datasets range from citation networks and document graphs to graphs from molecules and proteins. We conduct an extensive empirical study on four GNN models and nine post-hoc explanation approaches for node and graph classification tasks. We release both the benchmarks and reference implementations and make them available at https://github.com/Mandeep-Rathee/Bagel-benchmark.

Keywords: Explainability · Graph Neural Networks · Interpretability

1 Introduction

Graph neural networks (GNNs) [39,16,17,42,10] are representation learning techniques that encode structured information into low dimensional space using a feature aggregation mechanism over the node neighborhoods. GNNs have shown state-of-the-art performance across many scientific fields in various important downstream applications, such as molecular data analysis, drug discovery, toxic molecule detection, and community clustering [4,9,44].

There have been benchmarks and datasets for the interpretability of machine learning models [41,20]. The rising number of applications of GNNs in several sensitive domains like medicine and healthcare [4,21] necessitates the need to explain their decision-making process. GNNs are inherently black-box and non-interpretable. Moreover, due to the complex interplay of node features and neighborhood structure in the decision-making process, general explanation approaches [22,27,35] cannot be trivially applied for graph models. Consequently, several explanation techniques [43,8,40,45,32,14,31,46] have been proposed for

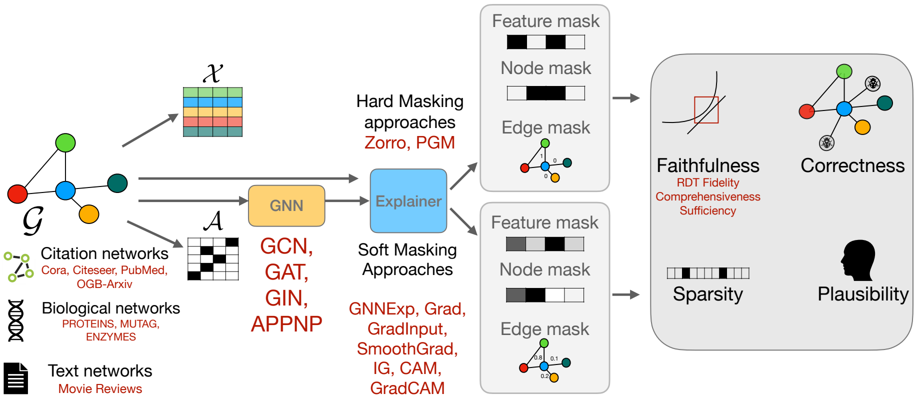


Fig. 1: An overview of the Bagel benchmark.

GNNs in the last few years. A known challenge in developing explanation techniques is evaluating the quality of explanations. This challenge also extends to the evaluation of explainability approaches for GNNs and is the focus of this work.

Existing approaches usually focus on a certain aspect of evaluation, sometimes even performed on synthetic datasets. For example, some works employ synthetic datasets with an already-known subgraph (sometimes referred to as the ground truth reason or simply the ground truth). Explanations are then evaluated based on their agreement with the ground truth. Such an evaluation is sometimes flawed as one cannot always guarantee that the GNN has used in the first place the seeded subgraph for its decision-making process [5]. Besides, there is no standardized procedure for comparing different GNN explanations. For example, feature attribution methods can generate soft masks (feature importance as a distribution) or hard masks (boolean selections) over features. Comparing hard and soft mask explanations needs a common and standardized protocol. Finally, the check for *human plausibility* and correctness have been ignored in the evaluation of GNN explainers. Human plausibility checks if a model predicts *right for the right reason*. On the other hand, the correctness of an explanation checks if the explainers is able to isolate spurious correlations and biases that are intentionally added to the training data as a proxy for biases present in real-world data.

To address the issues of a holistic evaluation and contribute a resource to the growing community on GNN explainability, we developed Bagel, a benchmark platform for evaluating explanation approaches for graph neural networks or GNNs. Bagel as depicted in Figure 1 covers diverse datasets [33,2,1,47] from the literature, a range of standardized metrics, and a modular, extendable framework for execution and evaluation of GNN explanation approaches, along with initial implementations of recent popular explanation methods. Bagel includes:

- Four diverse evaluation notions that evaluate the *faithfulness, sparsity, correctness, and plausibility* of GNN explanations on real-world datasets. While the first three metrics focus on evaluating the explainers, plausibility checks for explanations to be human congruent.
- Besides the widely used datasets for measuring the faithfulness of explanations, Bagel consists of new datasets for the plausibility of explanation approaches in our benchmark datasets.
- We unify multiple evaluations, metrics, domains, and datasets into an easy-to-use format that reduces the entry barrier for evaluating new approaches to explain GNNs. Additionally, we provide an extendable library to implement and evaluate GNN explainers.
- We conduct an extensive evaluation of GNNExplainer(GNNExp) [43], PGM-Explainer(PGM) [40], Zorro [8], Grad [34], GradInput[34], Integrated Gradient(IG) [38], SmoothGrad [36], CAM [26] and GradCAM [26] in Bagel.

We show that there is no clear winner in GNN explanation methods showing nuanced interpretations of the GNN explanation methods using the multiple metrics considered. We finally note that evaluating the effectiveness of explanations is an intrinsically human-centric task that ideally requires human studies. However, the goal of Bagel is to provide a fast and accurate evaluation strategy that is often desirable to develop new explainability techniques using empirical evaluation metrics before the human trial stage.

2 Background and Preliminaries

Graph Neural Networks. Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a graph with $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges. Let $\mathcal{A} \in \{0,1\}^{(n,n)}$ be the adjacency matrix of the graph where n is the number of nodes in the graph with $\mathcal{A}_{ij} = 1$ if there is an edge between node i and j and 0 otherwise. Let $\mathcal{X} \in \mathbf{R}^{(n,d)}$ be the features matrix where d is the number of features. For a given node $v \in \mathcal{V}$, $\mathbf{x}_v$ denotes its features vector, and $\mathcal{N}_v$ is a set of its neighbors. We denote the trained GNN model as f on the given graph. For each layer ℓ, the representation of node v is obtained by aggregating and transforming the representations of its neighboring nodes at layer $\ell - 1$

$$\boldsymbol{h}_v^{(\ell)} = \text{AGG}\left(\left\{\mathbf{x}_v^{(\ell-1)}, \left\{\mathbf{x}_u^{(\ell-1)} \mid u \in \mathcal{N}_v\right\}\right\}\right) \tag{1}$$

$$\mathbf{x}_v^{(\ell)} = \text{TRANSFORM}\left(\boldsymbol{h}_v^{(\ell)}, W^{(\ell)}\right) \tag{2}$$

where $W^{(\ell)}$ represents the weight matrix at layer ℓ. The aggregation function AGG function depends on the GNN model. For example, graph convolution network (GCN) [16] uses a degree-weighted aggregation of neighborhood features, whereas graph attention network (GAT) [39] learns neighborhood weights via an attention mechanism. The prediction can be obtained at the final layer using a *softmax function.* An additional pooling layer is applied for *graph classification* tasks before applying *softmax function.*

2.1 Post-hoc explanations and evaluation for GNNs

GNN Explanation. Post-hoc explainers for GNNs produce feature and local structure attributions where a combination of a masked set of nodes, edges, and features is retrieved as an explanation. To compute the explanation for a k-layer GNN, the k-hop neighborhood (i.e. its computational graph) of the node is utilized. For an explanation S, the explanation mask $M(S)$ is computed over the input nodes/edges and the features in the computational graph. Note that $M(S)$ could be binary or a continuous mask and contains the importance scores for the corresponding nodes/features. We note that as different explainers either return node or edge importance scores, for consistent comparison, we convert edge masks to node masks.

Bagel currently consists of 3 classes of post-explanation techniques: *gradient based*, *perturbation based* and *surrogate model* approaches. The gradient-based methods [34,38,36,26] are the simplest approaches for generating the explanation for any differentiable trained model. In these approaches, the importance scores for the explanation are usually computed using gradients of the input. The perturbation-based approaches [8,43,23,46,31] learns the important features and structural information by observing the predictive power of the model when noise is added to the input. The surrogate-based approaches [40,14,48] learn a simple interpretable model for the local neighborhood of a query node and its prediction. The explanations generated by this simple model are treated as the explanations of the original model. We note that Bagel is, in general, applicable for any explainer that returns binary (hard) or continuous (soft) importance scores (as depicted in Figure 1) for the input features/nodes/edges as an explanation.

2.2 Analysis of existing evaluation measure

Explanation Accuracy and true explanations. Evaluation of explanation methods for any predictive model is inherently tricky. Specifically, when evaluating already trained models, we are faced with the *lack of true explanations*. Collecting true explanations (sometimes referred to as ground truth) for GNNs is even more challenging due to the abstract nature of the input graphs. Moreover, depending on the explanation collection method, it is not always clear if the model used the collected explanation in its decision-making process. Nevertheless, some current works employ small synthetic datasets seeded with a ground truth subgraph. Consequently, metrics such as *explanation accuracy* [30,43] were proposed, which measure the agreement of found explanation with that of ground truth. Observing the false optimism of the accuracy metric for small explanation subgraphs, [8] proposed using *Precision* instead of accuracy.

Faithfulness to the model. An important notion for evaluating explanations is *faithfulness*, where the key idea is to measure how much the explanation characterizes the model's working. To measure faithfulness [30] degrade model performance by damaging the training dataset and measuring how each explanation method responds. The lack of ground truth again limits such a measure. [26] proposed to compute faithfulness as the difference of accuracy (or predicted probability) between the original predictions and the new predictions after masking out the input features found by the explanation. This was called *Fidelity* in their work. As the features cannot be removed in entirety to measure their impact [8] proposed RDT-Fidelity based on rate-distortion theory defined as the expected predictive score of an explanation over all possible configurations of the non-explanation features.

Explanation size. An important criterion to measure the goodness of an explanation is its size. For example, the full input is also a faithful explanation. However, humans find shorter explanations easier to analyze and reason. Works such as [26] measure the sparsity of an explanation as the fraction of features the explainer selects. Noting that this definition is not directly applicable to soft mask approaches, [8] proposes quantifying sparsity as entropy over the normalized distribution of explanation masks. We use the entropy-based sparsity metric as it can be applied both for hard and soft masking approaches.

Stability under input perturbations. The authors in [30] argued that the explanation should be stable under input perturbations. In particular, for graph classification, they perturbed test graphs by adding a few nodes/edges such that the final prediction remains the same as that for an unperturbed graph. Lower the change in explanation under perturbations better the stability. A challenge here is that there is no principled way to find the perturbations. For example, a part of the explanation might be altered under random perturbations even if the prediction is unchanged. In the following, we will see that the faithfulness metric of RDT-Fidelity already accounts for explanation stability without altering the explanation. We also note that there have been other benchmarks to study the robustness of GNN models [6,49]. However, we focus on explaining GNN model predictions rather than robustness. Having said this, we affirm that Bagel could be used in a complementary manner to these existing benchmarks to test trustworthy GNN models.

3 Bagel: A Unified Framework for Evaluating Explanations

We now present our framework Bagel for evaluating GNN explanations. Specifically, Bagel unifies existing and our proposed notions into two main classes. In the *first class* of measures, we aim to evaluate the explanation methods in the sense of whether they genuinely describe the model's workings. The first category includes three metrics: *faithfulness*, *sparsity*, and *correctness*. Faithfulness determines if an explanation alone can replicate the model's behavior. Sparsity focuses on rewarding shorter explanations. Correctness determines if the explanation model is able to detect any injected correlations responsible for altering model's behavior. The metrics in the second class are aimed at evaluating the GNN model itself. Here we propose *plausibility*, which measures how close the decision making process of the trained model (as revealed by explanations) is to human rationales.

3.1 Faithfulness: Can explanations approximate the model's behavior?

The key idea here is to evaluate the ability of the explanation to characterize the model's working. Unlike previous works, we argue that there is not a single measure of faithfulness that can be effectively used for all kinds of datasets and explanations. Consequently, we propose a set of two measures to quantify faithfulness depending on the dataset/explanation type.

- Rate distortion based fidelity. The fidelity of an explanation is usually measured by the ability of an explanation to approximate the model behavior [27]. For explanations that contain the feature attributions with or without structure attributions, we use the rate distortion theory based metric proposed in [8] to measure the fidelity of an explanation. In short, a subgraph of the node's computational graph and its set of features are relevant for a classification decision if the expected classifier score remains nearly the same when randomizing the remaining features.
  Let X denote the input node and features of the computational graph. In particular, X corresponds to a matrix of nodes in the computational graph and their corresponding feature values. As we use node and feature explanation masks, we compute the final $M(S)$ corresponding to some explanation S by an elementwise product of node and feature masks. The *RDT-Fidelity* of explanation $\mathcal{S}$ respect to the GNN f, input X and the noise distribution $\mathcal{N}$ is then given by
  $$\mathcal{F}(\mathcal{S}) = \mathbb{E}_{Y_{\mathcal{S}}|Z\sim\mathcal{N}}\left[\mathbb{1}_{f(X)=f(Y_{\mathcal{S}})}\right]. \tag{3}$$
  where the perturbed input is given by
  $$Y_{\mathcal{S}} = X \odot M(\mathcal{S}) + Z \odot (\mathbb{1} - M(\mathcal{S})), Z \sim \mathcal{N}, \tag{4}$$
  where $\odot$ denotes an element-wise multiplication, and $\mathbb{1}$ a matrix of ones with the corresponding size and $\mathcal{N}$ is a noise distribution. We choose the noise distribution as the global empirical distribution of the features. We sample the values from the underlying training data distribution. The purpose of adding noise is not to replace the unimportant features of input with 0, rather its value should not matter. Replacing unimportant features with 0 may cause side effects like in some datasets, the value 0 may represent some semantic meaning or bias towards some pooling strategy, for example, minpool. Also, the noise from global features distribution ensures that the perturbed data points are still in the same distribution as the original data [11].
  Connection to explanation stability. As shown in [8], explanations with high RDT-fidelity are highly stable. A high fidelity score implies that the explanation has high predictive power under perturbations of the rest of the input. Unlike the strategy of [30] to evaluate explanation stability, it is here ensured that the explanation itself is never altered.

The special case of dense feature representations. For some datasets, it is more appropriate to consider only structure based explanations. For example, when features themselves are dense representations extracted using some black-box embedding method, feature explanations as well as feature perturbations, might not make much sense. It is then more appropriate to check the abilities of the explanation with the rest of the nodes/edges removed and keep the features intact. Towards that, we employ the following measures of comprehensiveness and sufficiency, also used in [3].

- Comprehensiveness and Sufficiency. For explanations that contain only nodes or/and edges, we adapt the comprehensiveness and sufficiency measures of [3] for GNNs. Let $\mathcal{G}$ be the graph and $\mathcal{G}' \subseteq \mathcal{G}$ be the explanation graph with important (attribution) nodes/edges. In particular, $\mathcal{G}'$ is generated by removing all nodes/edges from $\mathcal{G}$ which are not part of the explanation.

  Let f be the trained GNN model and $f(\mathcal{G})_j$ be the prediction made by GNN for j^{th} class, where j is the predicted class. We measure fidelity by *comprehensiveness* (which answers the question if all nodes/edges in the graph needed to make a prediction were selected?) and *sufficiency* (if the extracted nodes/edges are sufficient to come up the original prediction?)

$$\textit{sufficiency} = f\left(\mathcal{G}\right)_j - f\left(\mathcal{G}'\right)_j, \quad \textit{comprehensiveness} = f\left(\mathcal{G}\right)_j - f\left(\mathcal{G}\backslash\mathcal{G}'\right)_j \quad (5)$$

A positive value of sufficiency implies that the probability prediction of f on $\mathcal{G}$ is higher than that of $\mathcal{G}'$, which tells us that nodes/edges in the $\mathcal{G}'$ are not sufficient to reach to the same or better prediction. A negative sufficiency score points out that the model f has a better prediction on $\mathcal{G}'$ than $\mathcal{G}$, which signifies that the explainer successfully eliminated specific noisy nodes which led to better performance. Similar arguments hold for comprehensiveness. In short, these measures should not be symmetric. The high *comprehensiveness* value shows that the prediction is most likely because of the explanation $\mathcal{G}'$ and low *comprehensiveness* value shows that $\mathcal{G}'$ is mostly not responsible for the prediction. Since most explainers retrieve soft masks, we employ aggregated *comprehensiveness* and *sufficiency* measures. In particular, we divide the soft masks into $|\mathcal{B}| = 5$ bins by using top $k \in \mathcal{B} = \{1\%, 5\%, 10\%, 20\%, 50\%\}$ of the explanation with respect to the soft masks values [29]. The aggregated *sufficiency* is defined as: $\frac{1}{|\mathcal{B}|}\left(\sum_{k=1}^{|\mathcal{B}|} f\left(\mathcal{G}\right)_j - f\left(\mathcal{G'}_k\right)_j\right)$. The aggregated *comprehensiveness* is defined in similar fashion.

3.2 Sparsity: Are the explanations non trivial?

High faithfulness ensures that the explanation approximates the model behavior well. However, the complete input ultimately determines the model behavior. Thus explanation sparsity is an important criterion for evaluation. Let p be the normalized distribution of explanation (feature) masks. Then sparsity of an explanation is given by the entropy $H(p)$ and is bounded from above by $\log(|M|)$ where M corresponds to a complete set of features or nodes. While an entire input can be a faithful explanation, evaluating an explanation with respect to its size is important. A shorter explanation is easier to analyze and is more humanely understandable. We adopt the entropy-based definition of sparsity as in [8] because of its applicability to both soft and hard explanation masks.

In particular, let p denote the normalized distribution of node/edge/feature masks. We compute the sparsity of an explanation as the entropy over the mask distribution: $H(p) = -\sum_{\phi \in M} p(\phi) \log p(\phi)$.

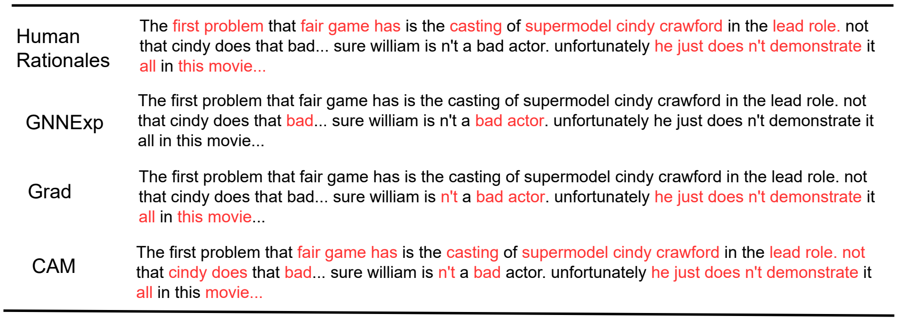
3.3 Correctness: Can the explanations detect externally injected correlations?

While the above measures are essential in that the given explanation is predictive, certain applications might need explanations for model debugging, for example, to detect any spurious correlations picked up by the model thereby increasing model bias. Towards that, we measure the correctness of an explanation in terms of its ability to recognize the *externally injected correlations*, which alters the model decision. A switch in the model decision is evidence of the use of these injected correlations in the actual decision-making process.

In particular, we first choose a set of incorrectly labeled nodes, V. To each such node v, we add edges to the nodes in the training data which have the same label as v. We call such edges *decoys.* We retrain the GNN model with the perturbed data. We measure the correctness of explanation $\mathcal{S}$ for nodes in V which are now correctly predicted in terms of precision and recall of the decoys in the returned explanation: $Precision_C = \frac{N_{de}}{N_e}, \quad Recall_C = \frac{N_{de}}{N_d}$, where N_{de} is the number of decoys in the obtained explanation, N_d total number of decoys injected and N_e is the size of the retrieved explanation. Note that our proposed approach of injecting correlations is different from using a synthetic graph with seeded ground truth. In particular, for the seeded graph approach, it is not always clear if the ground truth is actually picked up by the model to make its decision.

3.4 Plausibility: How close is the model's decision process to human rationales?

Human congruence or plausibility [19,18,37] tries to establish how close or congruent is the trained model to human rationales for solving a predictive task. Trained models often exhibit the *clever-hans effect*, that is, predictive models can adopt spurious correlations in the training data or due to misplaced inductive biases that have the right results for the wrong reasons. Towards this, data is collected from humans for perceptive tasks where humans explicitly provide their rationales. These human rationales are used as ground truth for evaluating if trained models are right for the right reasons. In Figure 2, we showcase a movie review and the explanations generated (in red) by different explainers. The true label for this review is negative, and the GCN makes the correct prediction for the review. For applications where obtaining human rationales is indeed possible, we propose the use of *token-level F1* for binary explanation masks and area under the precision-recall curve (AUPRC) for soft masks. The tokens are words in the input text and are modeled as nodes in the graph. The human rationales are binary masks over the nodes. The token level-F1 score is computed as macro-F1 for predicted binary explanation masks where human rationals serve the true labels. We also measure the area under the precision-recall curve for predicted



Human Rationales	The first problem that fair game has is the casting of supermodel cindy crawford in the lead role. not that cindy does that bad... sure william is n't a bad actor. unfortunately he just does n't demonstrate it all in this movie...
GNNExp	The first problem that fair game has is the casting of supermodel cindy crawford in the lead role. not that cindy does that bad... sure william is n't a bad actor. unfortunately he just does n't demonstrate it all in this movie...
Grad	The first problem that fair game has is the casting of supermodel cindy crawford in the lead role. not that cindy does that bad... sure william is n't a bad actor. unfortunately he just does n't demonstrate it all in this movie...
CAM	The first problem that fair game has is the casting of supermodel cindy crawford in the lead role. not that cindy does that bad... sure william is n't a bad actor. unfortunately he just does n't demonstrate it all in this movie...

Fig. 2: An anecdotal example of explanations generated by different explainers. The respective plausibility scores for the current example for GNNExp, Grad, and CAM are 0.50, 0.54, and 0.61 respectively. We observe that the explanation of CAM agrees best with human rationales.

soft explanation masks. Rather than fixing a threshold, AUPRC provides us a measure of precision-recall tradeoff across different decision thresholds. The reader might have noticed that this metric is similar to the explanation accuracy in earlier works. We argue against using the term *accurate* to measure plausibility as similarity to human rationale does not always guarantee that the model has learnt an explanation that contains the reasoning of the model itself and not only of the humans.

4 Experimental Setup

Models and Explainers. We demonstrate the use and advantage of the proposed framework by evaluating 9 explanation methods over 8 datasets and 4 GNN models. Currently, our benchmark consists of these GNN models: graph convolutional networks (GCN) [16], graph attention network (GAT) [39], the approximation of personalized propagation of neural predictions (APPNP) [17], and graph isomorphism network (GIN) [42]. The models were chosen based on their differences in (i) exploiting inductive biases (based on different feature aggregation strategies), (ii) test performance (see tables 4 and 5 in the Appendix) and (iii) response to injected correlations (see Table 1 and the corresponding discussion). We perform experiments with perturbation based approaches like GNNExplainer (GNNExp) [43] and Zorro [8], surrogate methods like PGM-Explainer (PGM) [40], and gradient-based approaches like Grad [34], GradInput[34], Integrated Gradient (IG) [38], SmoothGrad [36], CAM [26] and GradCAM [26]. GNNExp returns soft feature masks and edge masks. We transform the edge masks into node masks, in which we equally distribute the edge importance score to both nodes sharing the edge. The further details of these explainers are available in Appendix B. As already mentioned Bagel is extendable, and more approaches and explainers can be easily added.

4.1 Datasets

We now describe new and existing datasets used in our evaluation framework and the corresponding rationale.

New Dataset for Plausibility. To measure the plausibility of an explanation, we first require the corresponding human rationales. Since the existing graph datasets do not have such annotated information, we transform a text sentiment prediction task into a graph classification task. Specifically, we adopt the Movie Reviews dataset [47] from the ERASER benchmark [3]. The task here is binary classification, which differentiates between positive and negative movie reviews. Appendix A.1 gives a detailed description of dataset construction.

Dataset to measure Comprehensiveness and Sufficiency. Note that the comprehensiveness and sufficiency metrics are only applicable to node/edge explanations. We use the Movie Review dataset for these two measures too. The rationale is that the node features are generated using Glove and are not human-understandable. In this case, a structure-based explanation would be more meaningful than a feature-based one. Further, we evaluate comprehensiveness and sufficiency on molecule datasets including MUTAG [2], PROTEINS [1], and ENZYMES [24].

Datasets for Correctness. We employ two citation datasets Cora [33] and CiteSeer [33]. After injecting correlations/decoys corresponding to incorrectly labeled nodes as described in Section 3.3, we re-train the GNN model. The rationale behind adding homophily increasing correlations is the observation from previous works [15,50] that GNN's performance increases with higher homophily. A model will have picked up these correlations if the previous incorrect nodes are now correctly predicted. We further evaluate the correctness of the explanation only for newly correctly predicted nodes.

Datasets for RDT-Fidelity. We perform the *RTD-Fidelity* evaluation on both *node classification* and *graph classification* tasks. At node level, we use Citation datasets namely Cora, CiteSeer, PubMed, and ogbn-arxiv [12]. Table 4 in the Appendix shows the dataset statistics and GNNs performances. We select 300 nodes for Cora and CiteSeer, and PubMed and 1000 nodes for ogbn-arxiv randomly. For the graph classification task, we use MUTAG, PROTEINS, and ENZYMES datasets. We select 50 graphs for MUTAG and PROTEINS datasets and 200 graphs for ENZYMES dataset.

5 Result Analysis

5.1 Faithfulness

RDT-Fidelity. In Figure 3, we compare the RDT-Fidelity scores of various explanation methods. A common feature of Zorro and PGM is that they both learn the explanations from a sampled local dataset. The local dataset is created by perturbing the features of nodes from the computational graph (neighborhood of query nodes). While they employ different optimization strategies to find explanations, the result is a stable explanation that also reflects our results. The gradient-based explanations achieve the lowest fidelity. We also choose empty and random explanations as baselines. An empty explanation sets the importance scores for all nodes and features are set to 0. In case of a random explanation, we select nodes/features masks randomly from a uniform distribution. We observe that empty explanation performs similarly to GradInput. This is because the

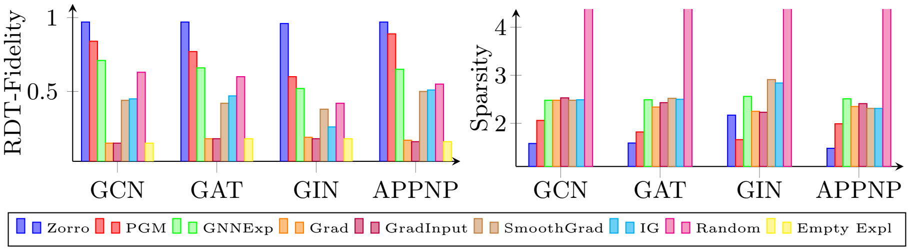


Fig. 3: Results for *RDT-Fidelity*(higher is better) and Sparsity(lower is better) on Cora dataset.

explanation mask output by GradInput is close to a zero vector. In the Appendix, we report RDT-Fidelity for the node/graph classification task in Tables 6 and 8.

Comprehensiveness and Sufficiency. We evaluate faithfulness for explanations for the Movie Reviews dataset using aggregated *comprehensiveness* and *sufficiency* measures. The results for soft-mask explanations are shown in Figure 4. GradCAM has the lowest *sufficiency*, which suggests that the explanations are sufficient to mimic the prediction of GNN models. On the other hand, the explanations generated by GradCAM with GCN and GIN suggest that an important part of the input still exists outside of the explanations required to approximate the GNN's prediction. Further, GNNExp, which so far outperformed gradient-based explanations for the node classification task, shows the worst sufficiency and comprehensiveness. Even if we use the complete feature set and only the node masks to evaluate explanations, the node masks for GNNExp are learned together with the feature explanations. This differs from gradient-based approaches, which ignore feature and structure explanation tradeoffs. The current performance of GNNExp indicates that it might not be appropriate to use entangled features and structure explanations independently. We report the *comprehensiveness* and *sufficiency* on the MUTAG, PROTEINS and ENZYMES datasets in the Appendix D (See Tables 11 to 13). In table 14, we report the *sufficiency* and *comprehensiveness* when the edge masks are used to generate induced subgraphs with different thresholds.

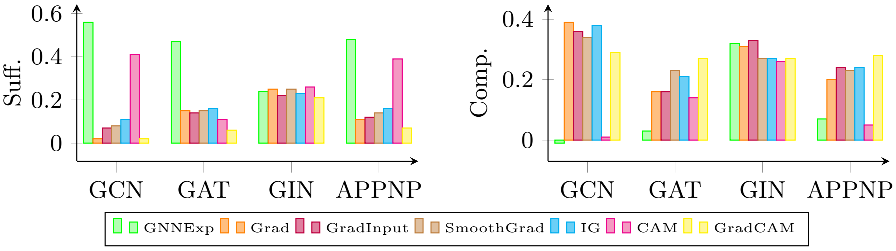


Fig. 4: Faithfulness as *comprehensiveness* and *sufficiency* measured for Movie Reviews dataset. Low sufficiency and high comprehensiveness indicate high faithfulness.

5.2 Sparsity

As already mentioned, a complete input could do well on all faithfulness measures. Therefore, we further look for sparser explanations. The results for node sparsity for explanations in the node classification task on the Cora dataset are provided in Figure 3. In the Appendix C, we report the node sparsity on the CiteSeer and PubMed datasets in Table 7. For the hard masking approaches (Zorro and PGM), Zorro outperforms PGM with all GNN models except for GIN. Conversely, there is no clear winner for the soft mask approach. The high sparsity for soft-masking approaches implies a near-uniform node attribution, and consequently, lower interpretability. In general, faithfulness and the sparsity of an explanation should be analyzed together. A uniformly distributed explanation mask could already provide an explanation with high faithfulness as it leads to using the complete input as an explanation. We also report feature sparsity for node classification in Appendix E (in Table 15). We observe similar trends on features level sparsity where Zorro outperforms over almost all datasets except SmothGrad when GIN is trained on CiteSeer. We further report node sparsity on MUTAG, PROTEINS, and ENZYMES in Appendix C(in table 9), where PGM outperforms over all three datasets.

5.3 Correctness

Table 1: The number of incorrectly labelled nodes (✗) decreases after addition of decoys. The number of new correctly labelled nodes after injecting decoys is listed under ✓.

Model	Cora			CiteSeer		
	✗	✓	↑(%)	✗	✓	↑(%)
GCN	88	79	89.7	329	229	69.6
GAT	86	85	98.8	311	301	96.7
GIN	6	6	100	56	56	100
APPNP	73	70	95.8	280	252	90.0

The correctness results corresponding to different models and explainers are reported for Cora (in Table 2) and CiteSeer (in Table 16). We report precision, recall, and F1 score by choosing the top k nodes for the soft explanations. The number of returned nodes is listed under $|\mathcal{S}|$ for hard masked approaches. Note that the number of decoys added per node is 10. For Table 2 and Table 16 we use $k = 20$. In Table 1, the effect of decoys can be seen where most of the earlier incorrectly classified nodes are now correctly classified except for GCN on CiteSeer. We also observe that the number of selected nodes for GIN is very low for Cora dataset (i.e., only a few nodes were initially incorrectly labeled). GNNExp outperforms all other based explainers in detecting the injected correlations for both Cora and CiteSeer (detailed results moved to Table 16 in the Appendix due to space constraints).

Comparing soft mask and hard mask approaches in this setting is tricky as for some approaches like Zorro, we cannot control the explanation size. For example, for GAT Zorro retrieved an explanation of size 40. A precision of 0.25 shows that it found all 10 injected correlations. Lack of feature ranking, as in soft mask approaches, makes it challenging to evaluate hard mask approaches for Correctness. For fairer evaluation, we plot the performance of soft mask approaches with different k in Appendix H. For example, the GNNExp shows

Table 2: Correctness of the explanation on Cora dataset. We use $k = 20$.

Methods	GCN				GAT				GIN				APPNP			
	P@k	R@k	F1	$\|\mathcal{S}\|$	P@k	R@k	F1	$\|\mathcal{S}\|$	P@k	R@k	F1	$\|\mathcal{S}\|$	P@k	R@k	F1	$\|\mathcal{S}\|$
HardMask																
Zorro	0.19	0.80	0.30	45	0.25	0.83	0.37	40	0.26	0.45	0.27	33	0.22	0.79	0.33	38
PGM	0.11	0.22	0.15	20	0.18	0.36	0.24	20	0.18	0.36	0.25	20	0.19	0.38	0.25	20
SoftMask																
GNNExp	**0.42**	**0.84**	**0.56**	20	**0.44**	**0.88**	**0.59**	20	**0.50**	**1.00**	**0.67**	20	**0.34**	**0.67**	**0.58**	20
Grad	0.23	0.46	0.31	20	0.29	0.58	0.39	20	0.30	0.60	0.40	20	0.33	0.67	0.45	20
GradInput	0.16	0.32	0.21	20	0.28	0.56	0.34	20	0.30	0.60	0.40	20	0.28	0.56	0.38	20
SmoothGrad	0.12	0.25	0.16	20	0.24	0.48	0.32	20	**0.50**	**1.00**	**0.67**	20	0.22	0.43	0.29	20
IG	0.16	0.32	0.22	20	0.24	0.49	0.33	20	**0.50**	**1.00**	**0.67**	20	0.28	0.55	0.37	20

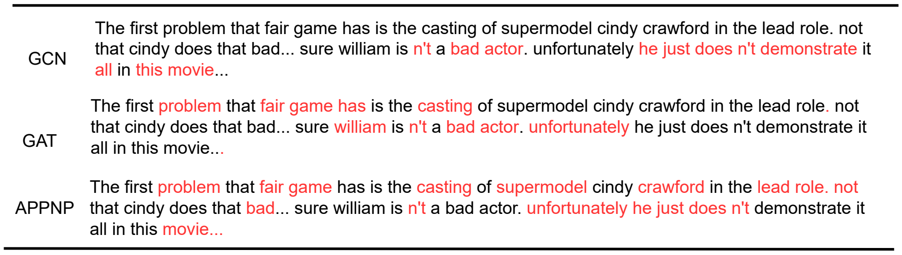
significant improvement when we increase the explanation size to 15. It is not surprising to see the performance degrades when we increase the size of the explanation further since it already had captured all injected decoys. Now it returns some irrelevant nodes in the explanation. Furthermore, in Table 18 and 19, we use the mean as a threshold to generate hard masks. As the mean threshold turns out to be very low for all approaches, almost all nodes of the computational graph are selected as the explanation. Consequently, we observe a very low correctness score (when measured in terms of precision).

5.4 Plausibility

Table 3 shows the *Plausibility* scores computed for explaining different GNN models. Recall that we compare explanations with human rationales to compute plausibility. The average size of human rationales over the test dataset is 165. To compute the token level F1 score, we use mean as a threshold to generate hard masks from soft masks.

We observe that all explainers assign the best plausibility scores to GCN. GIN obtains the second-best plausibility scores. We also observe that the overall difference in the plausibility scores over models is relatively small, with some exceptions like the combination of GIN and GNNExp. The corresponding explanation also has the largest size. This further highlights the issues of soft-hard mask conversion. AUPRC scores which directly use the soft masks are more stable. One surprising fact in these results is that even though other GNN models achieve higher test accuracy than GIN (see Table 5 in the Appendix). Overall, their explanations have similar plausibility as for GIN except for GNNExp. In such cases, an application user might want to look in more detail at specific correctly labeled instances to check if the model imitates human reasoning.

We now provide a concrete example of how the plausibility metric can be used in conjunction with the faithfulness metric to evaluate the model's decision-making process. From our previous example in Figure 2, we choose Grad, which achieves the best faithfulness score in this example. In Figure 5, we compare the explanations of different models provided by Grad explainer and compare the explanations based on plausibility. We observe that GCN achieves the highest faithfulness and plausibility scores.



GCN The first problem that fair game has is the casting of supermodel cindy crawford in the lead role. not that cindy does that bad... sure william is n't a bad actor. unfortunately he just does n't demonstrate it all in this movie...

GAT The first problem that fair game has is the casting of supermodel cindy crawford in the lead role. not that cindy does that bad... sure william is n't a bad actor. unfortunately he just does n't demonstrate it all in this movie...

APPNP The first problem that fair game has is the casting of supermodel cindy crawford in the lead role. not that cindy does that bad... sure william is n't a bad actor. unfortunately he just does n't demonstrate it all in this movie...

Fig. 5: An example to illustrate the use of *plausibility* in conjunction with *faithfulness* to select the model that best agrees with human rationales. We compare different models for Grad explanations because Grad explanations are highly faithful. Grad explanations over GCN agree best with human rationales.

Table 3: Plausibility for movie review dataset measured by auprc and F1 score (macro). $|\mathcal{S}|$ represents the average size of the explanations generated by the explainers.

Methods	GCN			GAT			GIN			APPNP		
	auprc	F1	$\|\mathcal{S}\|$	auprc	F1	$\|\mathcal{S}\|$	auprc	F1	$\|\mathcal{S}\|$	auprc	F1	$\|\mathcal{S}\|$
HardMask												
PGM	—	0.42	25	—	**0.43**	25	—	**0.43**	25	—	**0.43**	25
SoftMask												
GNNExp	0.46	**0.54**	168	0.43	**0.54**	149	0.45	0.35	410	0.45	0.53	158
Grad	0.44	**0.52**	265	0.38	0.51	158	0.40	**0.52**	156	0.38	0.50	255
GradInput	0.39	**0.51**	221	0.37	0.50	154	0.39	**0.51**	154	0.37	0.50	227
SmoothGrad	0.40	**0.52**	219	0.37	0.50	154	0.40	**0.52**	172	0.38	0.50	221
IG	0.37	0.49	225	0.37	0.50	188	0.39	**0.51**	186	0.38	0.50	219
CAM	0.54	**0.61**	224	0.40	0.51	177	0.44	0.55	156	0.44	0.53	195
GradCAM	0.67	0.34	175	0.67	**0.35**	191	0.67	0.34	166	0.67	0.34	188

6 Conclusion

We develop a unified, modular, extendable benchmark called Bagel to evaluate GNN explanations on four diverse axes: 1) *faithfulness*, 2) *sparsity*, 3) *correctness*, and 4) *plausibility*. Faithfulness measured via *RDT-Fidelity* can be employed for a wide set of tasks and datasets. We note that high RDT-Fidelity also implies high explanation stability. The *comprehensiveness* and *sufficiency* measures should be used to evaluate the faithfulness of structure-based explanations where perturbing features might not be feasible. It is important to measure the sparsity of the explanation to avoid the extreme case of using the whole input as an explanation. Correctness should be used carefully, as injecting appropriate correlations to change a model's decision is not always straightforward. Plausibility measures the joint utility of the explanation method and the trained GNN model with respect to human rationales. Assuming that the generated explanations are faithful to the model, one can use plausibility to check the model's congruence to human rationales. This means that the loss of plausibility can be either due to human-incongruent correlations or due to the non-faithfulness of the explainer. To fully interpret the results of plausibility, one should first check the explanation's faithfulness.

References

1. Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S., Smola, A.J., Kriegel, H.P.: Protein function prediction via graph kernels. Bioinformatics **21**(suppl_1), i47–i56 (2005)
2. Debnath, A.K., Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. Journal of medicinal chemistry **34**(2), 786–797 (1991)
3. DeYoung, J., Jain, S., Rajani, N.F., Lehman, E., Xiong, C., Socher, R., Wallace, B.C.: Eraser: A benchmark to evaluate rationalized nlp models. arXiv preprint arXiv:1911.03429 (2019)
4. Dong, T.N., Mucke, S., Khosla, M.: Mucomid: A multitask graph convolutional learning framework for mirna-disease association prediction. IEEE/ACM Transactions on Computational Biology and Bioinformatics (2022)
5. Faber, L., K. Moghaddam, A., Wattenhofer, R.: 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. pp. 332–341 (2021)
6. Fan, W., Jin, W., Liu, X., Xu, H., Tang, X., Wang, S., Li, Q., Tang, J., Wang, J., Aggarwal, C.: Jointly attacking graph neural network and its explanations. arXiv preprint arXiv:2108.03388 (2021)
7. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
8. Funke, T., Khosla, M., Rathee, M., Anand, A.: Z orro: Valid, sparse, and stable explanations in graph neural networks. IEEE Transactions on Knowledge and Data Engineering (2022)
9. Gaudelet, T., Day, B., Jamasb, A.R., Soman, J., Regep, C., Liu, G., Hayter, J.B.R., Vickers, R., Roberts, C., Tang, J., Roblin, D., Blundell, T.L., Bronstein, M.M., Taylor-King, J.P.: Utilizing graph machine learning within drug discovery and development. Briefings in Bioinformatics **22**(6) (05 2021). https://doi.org/10.1093/bib/bbab159, https://doi.org/10.1093/bib/bbab159, bbab159
10. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS (2017)
11. Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: A benchmark for interpretability methods in deep neural networks. In: Advances in Neural Information Processing Systems. pp. 9737–9748 (2019)
12. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. arXiv preprint arXiv:2005.00687 (2020)
13. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. arXiv preprint arXiv:2005.00687 (2020)
14. Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., Chang, Y.: Graphlime: Local interpretable model explanations for graph neural networks. arXiv:2001.06216 (2020)
15. Khosla, M., Setty, V., Anand, A.: A comparative study for unsupervised network representation learning. TKDE (2019)
16. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)

17. Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: Graph neural networks meet personalized pagerank. International Conference on Learning Representations (ICLR) (2019)
18. Lage, I., Chen, E., He, J., Narayanan, M., Kim, B., Gershman, S., Doshi-Velez, F.: An evaluation of the human-interpretability of explanation. arXiv preprint arXiv:1902.00006 (2019)
19. Lei, T., Barzilay, R., Jaakkola, T.: Rationalizing neural predictions. arXiv preprint arXiv:1606.04155 (2016)
20. Liu, Y., Khandagale, S., White, C., Neiswanger, W.: Synthetic benchmarks for scientific research in explainable machine learning. arXiv preprint arXiv:2106.12543 (2021)
21. Lu, H., Uddin, S.: A weighted patient network-based framework for predicting chronic diseases using graph neural networks. Scientific reports **11**(1), 1–12 (2021)
22. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in neural information processing systems. pp. 4765–4774 (2017)
23. Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., Zhang, X.: Parameterized explainer for graph neural network. arXiv preprint arXiv:2011.04573 (2020)
24. Morris, C., Kriege, N.M., Bause, F., Kersting, K., Mutzel, P., Neumann, M.: Tudataset: A collection of benchmark datasets for learning with graphs. In: ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020) (2020), `www.graphlearning.io`
25. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
26. Pope, P.E., Kolouri, S., Rostami, M., Martin, C.E., Hoffmann, H.: Explainability methods for graph convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10772–10781 (2019)
27. Ribeiro, M.T., Singh, S., Guestrin, C.: " 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. pp. 1135–1144 (2016)
28. Rousseau, F., Vazirgiannis, M.: Graph-of-word and tw-idf: new approach to ad hoc ir. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 59–68 (2013)
29. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R.: Evaluating the visualization of what a deep neural network has learned. IEEE transactions on neural networks and learning systems **28**(11), 2660–2673 (2016)
30. Sanchez-Lengeling, B., Wei, J., Lee, B., Reif, E., Wang, P., Qian, W.W., McCloskey, K., Colwell, L., Wiltschko, A.: Evaluating attribution for graph neural networks. Advances in neural information processing systems **33** (2020)
31. Schlichtkrull, M.S., De Cao, N., Titov, I.: Interpreting graph neural networks for nlp with differentiable edge masking. arXiv preprint arXiv:2010.00577 (2020)
32. Schnake, T., Eberle, O., Lederer, J., Nakajima, S., Schütt, K.T., Müller, K.R., Montavon, G.: Higher-order explanations of graph neural networks via relevant walks. IEEE transactions on pattern analysis and machine intelligence **44**(11), 7581–7596 (2021)
33. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI magazine **29**(3), 93–93 (2008)
34. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)

35. Singh, J., Anand, A.: Model agnostic interpretability of rankers via intent modelling. In: Conference on Fairness, Accountability, and Transparency (2020)
36. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
37. Strout, J., Zhang, Y., Mooney, R.J.: Do human rationales improve machine explanations? arXiv preprint arXiv:1905.13714 (2019)
38. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 3319–3328. JMLR. org (2017)
39. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. ICLR (2018)
40. Vu, M.N., Thai, M.T.: Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020 (2020)
41. Wiegreffe, S., Marasović, A.: Teach me to explain: A review of datasets for explainable nlp. arXiv preprint arXiv:2102.12060 (2021)
42. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (2019)
43. Ying, R., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnn explainer: A tool for post-hoc explanation of graph neural networks. arXiv preprint arXiv:1903.03894 (2019)
44. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 974–983 (2018)
45. Yuan, H., Tang, J., Hu, X., Ji, S.: Xgnn: Towards model-level explanations of graph neural networks. In: KDD '20. p. 430–438. Association for Computing Machinery (2020)
46. Yuan, H., Yu, H., Wang, J., Li, K., Ji, S.: On explainability of graph neural networks via subgraph explorations. arXiv preprint arXiv:2102.05152 (2021)
47. Zaidan, O., Eisner, J.: Modeling annotators: A generative approach to learning from annotator rationales. In: Proceedings of the 2008 conference on Empirical methods in natural language processing. pp. 31–40 (2008)
48. Zhang, Y., Defazio, D., Ramesh, A.: Relex: A model-agnostic relational model explainer. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society. pp. 1042–1049 (2021)
49. Zheng, Q., Zou, X., Dong, Y., Cen, Y., Yin, D., Xu, J., Xu, J., Yang, Y., Tang, J.: Graph robustness benchmark: Benchmarking the adversarial robustness of graph machine learning. In: Vanschoren, J., Yeung, S. (eds.) Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks. vol. 1 (2021), https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/6cdd60ea0045eb7a6ec44c54d29ed402-Paper-round2.pdf
50. Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., Koutra, D.: Beyond homophily in graph neural networks: Current limitations and effective designs. Advances in Neural Information Processing Systems **33**, 7793–7804 (2020)

Appendix

A More details on datasets

In this section we describe in detail the tasks and datasets used in our framework.

A.1 Datasets

For the *node classification* task, we use citation datasets [33], namely, Cora, CiteSeer, and PubMed and further, we use bigger dataset from OGB [12] namely ogbn-arxiv. Each paper represents a node in these citation datasets, and there is an edge between two papers if one cites the other. Each node has its input features vector and a label. We follow the semi-supervised setting from [16,39] or data split where only 20 nodes per label are used in training and 500 nodes for validation, and 1000 nodes for test data. For the ogbn-arxiv dataset, we follow the split from OGB [12]. The details are available in Table 4. We use the fully supervised split for the correctness experiment where all the nodes except validation and test data are in training data.

For the *graph classification* task, we use molecules datasets like PROTEINS [1], MUTAG [2] and ENZYMES [24] and text dataset like Movie Reviews [47]. In the PROTEINS dataset, each graph represents a protein, and the task is to classify the protein into enzymes or non-enzymes. The MUTAG dataset contains 188 chemical compounds, and the task is to classify whether the compound has a mutagenic effect on a bacterium. The Movie Reviews is a text dataset that we transform into a graph dataset. In the transformed dataset, each graph represents a movie review, and the task is to classify the review's sentiment as *Positive or Negative*. Table 5 reports the data statistics and model accuracy. We now describe the construction of Movie Reviews dataset. We use 80% of the graphs as the train set and 20% as the test set.

Construction of Movie Reviews Dataset. Each input instance or review is a passage of text, typically with multiple sentences. Each input review is annotated by humans which reflects the actual "human" reasons for predicting the review's sentiment. These annotations are extractive pieces of text, and we call them human rationales. We transform sentences into graphs using the graph-of-words approach [28]. We remove stopwords, such as "the" or "a" as a pre-processing step. The complete list of used stopwords is included in our repository. Each word is represented as a node, and all words within a sliding window of three are connected via edges. We use the output of a pre-trained Glove model [25] as features. Figure 6 provides an example of a graph from the Movie Reviews dataset.

A.2 Dataset statistics and Model Performance

B Details of explainers

In the following section, we summarize the GNN explanation approaches currently implemented in Bagel.

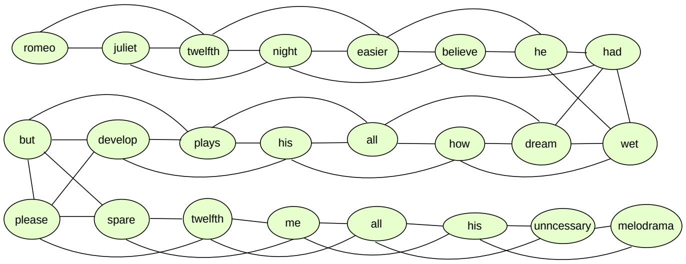


Fig. 6: An example for text to graph generation. The graph is corresponding to input sentences *"? romeo and juliet ' , and ? the twelfth night ' . it is easier for me to believe that he had a wet dream and that 's how all his plays develop , but please spare me all of this unnecessary melodrama."*

Table 4: Datasets statistics and model performance for node classification.

					Test Accuracy			
Dataset	Class	d	$\|V\|$	$\|E\|$	**GCN**	**GAT**	**GIN**	**APPNP**
Cora	7	1433	2708	10556	0.794	0.791	0.679	0.799
CiteSeer	6	3703	3327	9104	0.675	0.673	0.480	0.663
PubMed	3	500	19717	88648	0.782	0.765	0.590	0.782
ogbn-arxiv	40	128	169343	1166243	0.610	0.633	0.571	0.640

Table 5: Datasets statistics and model performance for Graph classification.

						Test Accuracy			
Dataset	#Graphs	Class	d	Avg. $\|V\|$	Avg. $\|E\|$	**GCN**	**GAT**	**GIN**	**APPNP**
MUTAG	188	2	7	17.9	39.6	0.76	0.76	0.79	0.76
PROTEINS	1113	2	3	39.1	145.6	0.66	0.58	0.69	0.58
ENZYMES	600	6	3	32.6	124.3	0.46	0.43	0.47	0.32
Movie Reviews	2000	2	300	500.8	1997.5	0.85	0.85	0.78	0.82

B.1 Grad

For a given node v and the trained GNN model $f(\mathbf{x}_v, \mathcal{G}, \phi)$, where ϕ is a set of parameters, the gradient based approach, Grad [34], generate an explanation by assigning importance scores to the input features. The score corresponding to a feature reveals the importance of this feature in making predictions. The high score corresponds to high importance. The explanation for node v is given by

$$\mathcal{S}(\mathbf{x}_v) = \frac{\partial f}{\partial \mathbf{x}_v} \tag{6}$$

the gradients of f with respect to input features $\mathbf{x}_v$.

B.2 GradInput

GradInput explanations are transformed versions of Gradient based explanations using element-wise multiplication with input features. The explanation for node v is given by:

$$\mathcal{S}(\mathbf{x}_v) = \frac{\partial f}{\partial \mathbf{x}_v} \odot \mathbf{x}_v \tag{7}$$

B.3 Integrated Gradient (IG)

The gradient based explanations are often noisy and can sometimes be insensitive with respect to the input. For example, if the learnt function f is a straight line, then the gradients are the same with respect to different inputs. IG [38] proposed the interpolation based gradient method, which cumulates the gradients along a straight path between input features vector $\mathbf{x}_v$ and a baseline vector $\mathbf{x}'$ of all zeros or ones. The explanation for node v is given by:

$$\mathcal{S}(\mathbf{x}_v) = (\mathbf{x}_v - \mathbf{x}') \times \int_{\alpha=0}^{1} \frac{\partial f\left(\mathbf{x}' + \alpha \times (\mathbf{x}_v - \mathbf{x}')\right)}{\partial \mathbf{x}_v} d\alpha \tag{8}$$

B.4 SmoothGrad

The SmoothGrad [36] claims that the gradients of model f may saturate, which means the importance score of a feature causes a substantial effect globally, but it shows a shallow effect locally. Also, the gradient based explanations are noisy. It proposed the mechanism of smoothing the noisy gradients by adding noise to the input. This process generates extra samples for training, and the explanations are more robust to the noise. The explanations of SmoothGrad are given by:

$$\hat{\mathcal{S}}(\mathbf{x}_v) = \frac{1}{n} \sum_{1}^{n} \mathcal{S}\left(\mathbf{x}_v + \mathcal{N}\left(0, \sigma^2\right)\right) \tag{9}$$

where n is the number of samples and $\mathcal{N}\left(0, \sigma^2\right)$ is the noise distribution.

B.5 CAM

CAM [26] explains the graph classification tasks only. For graph classification tasks, we use an additional global average pooling (GAP) layer to generate the representation of the graph. For a graph $\mathcal{G}$, let $f_{\mathcal{G}}$ be the trained model for the graph classification task, which can be represented as:

$$f_{\mathcal{G}}(\mathcal{A}, \mathcal{X}) = \sigma(\mathrm{GAP}(\mathbf{x}_u^{(L)} \mid u \in \mathcal{G}), w) \tag{10}$$

where L represents the final layer of GNN, $\mathbf{x}_u^{(L)}$ is the representation of node u at layer L and w represents the trainable parameters for the classification layer and σ is the activation function. CAM assumes that the final layer representations encode the input feature behavior that corresponds to prediction and the input feature importance is given by a weighted sum of different features. The explanation of the graph $\mathcal{G}$ is given by:

$$\mathcal{S}(\mathcal{G}) = \mathrm{ReLU}\left(\sum_k w_k \mathcal{X}_k^{(L)}\right) \tag{11}$$

where $\mathcal{X}_k^{(L)}$ represents the k^{th} feature at layer L.

B.6 GradCAM

GradCAM [26] is an extended version of CAM designed for the graph classification based GNN models, which does not need the global average pooling (GAP) layer. It also uses the final representation to generate the importance score. It assigns the gradient based weights to the representation. The weight for k^{th} feature at layer l is computed as:

$$\alpha_k^{(l)} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial f_{\mathcal{G}}}{\partial \mathcal{X}_k^{(l)}} \tag{12}$$

where N is the number of nodes in graph $\mathcal{G}$. Finally, the explanation of the graph is given by:

$$\mathcal{S}(\mathcal{G}) = \mathrm{ReLU}\left(\sum_k \alpha_k^l \mathcal{X}_k^{(l)})\right) \tag{13}$$

B.7 Zorro

Zorro [8] is a greedy-combinatorial algorithm that optimizes *RDT-Fidelity* to receive valid, sparse, and stable explanations. Zorro iteratively extends the explanation with the feature or node, which yields the highest increase in *RDT-Fidelity.* The addition of features and nodes is stopped when a pre-defined threshold is reached. Hence, it automatically determines the explanation size. For the formulation and details of *RDT-Fidelity* see Section 3.1. The authors in [8] evaluated two different *RDT-Fidelity* thresholds. Here, we performed the experiments using the higher threshold of $\tau = 0.98$. Zorro is designed for node classification and is model agnostic. The explanations consist of hard features and hard node masks.

B.8 PGM

PGM-Explainer (PGM) [40] is a surrogate based method that fits a simple and interpretable model on a sampled local dataset. The first step is data generation which is the collection of perturbed input for a given target node and its prediction. Secondly, the variable selector removes the irrelevant samples from the generated data. Finally, a Bayesian network is trained on the sampled data, and the explanations generated by the Bayesian network are treated as explanations of the GNN model for the target node. PGM retrieves hard masks over the nodes. PGM explains both node and graph classification tasks.

B.9 GNNExp

For a given graph $\mathcal{G}$, GNNExp [43] learns subgraph $\mathcal{G}' \subseteq \mathcal{G}$ which contains the important graph structure (mainly edges) and import features which are responsible for the prediction. For the subgraph, GNNExp learns the soft masks over edges and features, which are optimized using the mutual information between the prediction of GNN on $\mathcal{G}'$ and the prediction (Y) of GNN on $\mathcal{G}$. Mathematically,

$$\max_{\mathcal{G}'} MI\left(Y, \mathcal{G}'\right) = H(Y) - H\left(Y \mid \mathcal{G} = \mathcal{G}'\right) \tag{14}$$

where $H(.)$ is the entropy. GNNExp learns masks locally and is applicable to all graph based tasks like graph classification, node classification and link prediction.

C Results of RDT-Fidelity and Sparsity

In Tables 6 and 8, we report *RDT-Fidelity* for the graph and classification tasks. We observe that all methods, including the gradient-based approaches, perform relatively well except for the GCN model. PGM shows more consistent performance across all models and datasets. We leave out Zorro as it is not applicable for the graph classification task. In Tables 7 and 9, we report the Sparsity for the graph and node classification tasks.

D Results on Sufficiency and Comprehensiveness

In Tables 11 to 13, we report *sufficiency* and *comprehensiveness* for the graph classification task. In table 14, we report the *sufficiency* and *comprehensiveness* for edge-level explanations for the GCN model trained on molecules datasets. We calculate the *sufficiency* and *comprehensiveness* when the edge masks are used to generate induced subgraphs with different thresholds. GNNExp-Edge represents GNNExp when edge masks are directly used as the explanations. GradEdge represents gradients over edges. We also use random edge masks as a baseline. We observe that no single explainer consistently outperforms all GNNs on these three molecule datasets. GNNExp-Edge outperforms on MUTAG and ENZYMES datasets, and GradEdge outperforms on PROTEINS dataset.

Table 6: Results for *RDT-Fidelity* for node classification.

Methods	Cora				CiteSeer				PubMed			
	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP
HardMask												
Zorro	**0.97**	**0.97**	**0.96**	**0.97**	**0.97**	**0.97**	**0.97**	**0.96**	**0.96**	**0.97**	**0.97**	**0.96**
PGM	0.84	0.77	0.60	0.89	0.92	0.93	0.73	0.95	0.78	0.69	0.74	**0.96**
SoftMask												
GNNExp	0.71	0.66	0.52	0.65	0.68	0.69	0.51	0.62	0.67	0.73	0.67	0.72
Grad	0.15	0.18	0.19	0.17	0.17	0.19	0.28	0.18	0.37	0.43	0.42	0.37
GradInput	0.15	0.18	0.18	0.16	0.16	0.18	0.26	0.17	0.36	0.42	0.42	0.36
SmoothGrad	0.44	0.42	0.38	0.50	0.54	0.57	0.45	0.62	0.52	0.53	0.67	0.59
IG	0.45	0.47	0.26	0.51	0.53	0.70	0.45	0.62	0.52	0.56	0.68	0.59
Empty	0.15	0.18	0.18	0.16	0.16	0.18	0.26	0.17	0.36	0.42	0.42	0.36
Random	0.63	0.60	0.42	0.55	0.59	0.57	0.52	0.52	0.75	0.67	0.67	0.70

Table 7: Results for sparsity (computed as entropy over mask distribution) for node classification. The lower the score, the sparser the explanation.

Methods	Cora				CiteSeer				PubMed			
	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP
HardMask												
Zorro	**1.58**	**1.59**	2.17	**1.48**	**1.26**	**1.09**	1.58	**1.07**	**1.51**	**1.31**	2.18	**1.25**
PGM	2.06	1.82	**1.66**	1.99	1.47	1.59	**1.10**	1.54	1.64	1.16	**1.62**	2.93
SoftMask												
GNNExp	2.48	2.49	2.56	2.51	1.67	1.67	1.70	1.68	2.70	2.71	2.71	2.71
Grad	2.48	2.34	2.25	2.35	1.70	1.61	1.55	1.60	2.91	2.76	3.11	2.73
GradInput	2.53	2.43	2.23	2.41	1.61	1.58	1.54	1.52	3.02	2.94	3.41	2.81
SmoothGrad	2.48	2.52	2.91	2.31	1.77	1.77	1.93	1.66	2.89	3.02	3.23	2.54
IG	2.49	2.50	2.84	2.31	1.76	1.77	1.91	1.66	2.84	2.89	3.06	2.58
Random Expl.	7.71	7.71	7.71	7.71	7.92	7.92	7.92	7.92	9.69	9.69	9.69	9.69

Table 8: *RDT-Fidelity* for graph classification.

Methods	MUTAG				PROTEINS				ENZYMES			
	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP
HardMask												
PGM	0.58	**1.00**	0.60	**1.00**	**0.97**	**1.00**	0.89	**1.00**	0.35	**0.99**	0.28	**0.98**
SoftMask												
GNNExp	**0.73**	0.96	**0.75**	0.99	0.38	0.46	0.23	**1.00**	**0.79**	0.89	**0.68**	0.79
Grad	0.39	0.86	0.58	0.96	0.93	0.90	**0.90**	0.94	0.52	0.30	0.31	0.70
GradInput	0.38	0.87	0.58	0.96	0.93	0.91	**0.90**	0.94	0.53	0.30	0.32	0.69
SmoothGrad	0.38	0.86	0.59	0.96	0.93	0.90	**0.90**	0.94	0.52	0.31	0.31	0.69
IG	0.38	0.85	0.59	0.96	0.93	0.92	**0.90**	0.94	0.53	0.31	0.32	0.70
CAM	0.39	0.85	0.59	0.97	0.93	0.91	**0.90**	0.95	0.52	0.30	0.32	0.69
GradCAM	0.38	0.85	0.58	0.96	0.93	0.90	0.89	0.95	0.52	0.29	0.31	0.69

Table 9: Sparsity for graph classification.

Methods	MUTAG				PROTEINS				ENZYMES			
	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP
HardMask												
PGM	**1.37**	**1.37**	**1.36**	**1.36**	**2.34**	**2.34**	**2.23**	**2.24**	**1.86**	**1.86**	**1.85**	**1.85**
SoftMask												
GNNExp	2.78	2.67	2.77	2.78	3.65	3.64	3.66	3.65	3.26	3.25	3.27	3.26
Grad	2.77	2.70	2.61	2.68	3.35	3.20	2.97	3.23	3.45	3.16	3.26	3.33
GradInput	2.76	2.59	2.56	2.59	3.24	3.16	2.90	3.18	3.29	3.01	3.14	3.28
SmoothGrad	2.75	2.63	2.67	2.66	3.24	3.17	2.91	3.19	3.29	3.05	3.14	3.28
IG	2.63	2.75	2.62	2.65	3.29	3.17	2.91	3.18	3.31	3.23	3.17	3.32
CAM	2.75	2.65	2.23	2.71	3.33	3.18	2.80	3.15	3.44	3.43	3.19	3.30
GradCAM	2.86	2.86	2.86	2.86	3.34	3.34	3.34	3.34	3.47	3.47	3.48	3.47

Table 10: Faithfulness as *comprehensiveness* and *sufficiency* measured for Movie Reviews dataset. Low sufficiency and high comprehensiveness indicate high faithfulness.

Methods	GCN		GAT		GIN		APPNP	
	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.
GNNExp	0.56	-0.01	0.47	0.03	0.24	0.32	0.48	0.07
Grad	**0.02**	**0.39**	0.15	0.16	0.25	0.31	0.11	0.20
GradInput	0.07	0.36	0.14	0.16	0.22	**0.33**	0.12	0.24
SmoothGrad	0.08	0.34	0.15	0.23	0.25	0.27	0.14	0.23
IG	0.11	0.38	0.16	0.21	0.23	0.27	0.16	0.24
CAM	0.41	0.01	0.11	0.14	0.26	0.26	0.39	0.05
GradCAM	**0.02**	0.29	**0.06**	**0.27**	**0.21**	0.27	**0.07**	**0.28**

E Results on Feature Sparsity

In Table 15, we report feature sparsity for node classification task. We do not report feature sparsity for PGM, since it does not retrieve features in explanation.

F Results on Correctness on CiteSeer

We report correctness for CiteSeer dataset in Table 16.

G Experiments on ogbn-arxiv Dataset

We perform experiment for RDT-Fidelity and sparsity on the ogbn-arxiv dataset [13]. The dataset statistics and performances of GNNs are available in

Table 11: Faithfulness as *comprehensiveness* and *sufficiency* measured for MUTAG dataset. Low *sufficiency* and high *comprehensiveness* indicates high faithfulness.

Models	GCN		GAT		GIN		APPNP	
	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.
GNNExp	**0.03**	0.17	0.09	**0.12**	0.64	0.67	0.01	0.04
Grad	0.11	0.01	0.16	0.02	0.61	0.62	**-0.04**	**0.64**
GradInput	0.10	**0.63**	0.14	0.03	0.61	0.62	**-0.04**	0.56
SmoothGrad	0.11	0.23	0.15	0.02	0.58	0.63	**-0.04**	0.56
IG	0.08	0.21	0.12	0.03	0.61	**0.79**	**-0.04**	0.56
CAM	0.15	0.01	0.14	-0.01	0.61	0.62	0.11	-0.06
GradCAM	0.07	0.07	**0.07**	0.07	**0.40**	0.61	0.04	0.01

Table 12: Faithfulness as *comprehensiveness* and *sufficiency* measured for PROTEINS dataset. Low *sufficiency* and high *comprehensiveness* indicates high faithfulness.

Models	GCN		GAT		GIN		APPNP	
	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.
GNNExp	0.13	0.38	**0.04**	0.11	0.25	0.50	**0.03**	0.04
Grad	0.12	0.47	0.07	**0.27**	0.26	0.42	0.14	**0.05**
GradInput	0.21	0.53	0.12	0.06	0.26	0.42	0.13	**0.05**
SmoothGrad	0.21	0.54	0.11	0.08	0.23	0.46	0.13	0.04
IG	0.19	0.39	0.19	0.03	0.30	0.45	0.14	0.04
CAM	**0.08**	0.48	0.19	-0.01	0.36	0.44	0.17	-0.05
GradCAM	0.16	**0.58**	0.13	0.12	**0.19**	**0.61**	0.13	0.03

Table 13: Faithfulness as *comprehensiveness* and *sufficiency* measured for ENZYMES dataset. Low *sufficiency* and high *comprehensiveness* indicates high faithfulness.

Models	GCN		GAT		GIN		APPNP	
	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.
GNNExp	**0.06**	0.33	**0.08**	**0.43**	0.34	0.62	-0.01	0.12
Grad	0.07	0.34	0.17	0.39	0.33	0.55	0.09	-0.05
GradInput	0.14	0.33	0.21	0.35	0.39	0.51	0.08	-0.05
SmoothGrad	0.14	0.32	0.21	0.36	0.41	0.51	0.09	-0.06
IG	0.14	0.33	0.24	0.32	0.41	0.53	0.09	-0.04
CAM	0.11	0.27	0.16	0.23	0.46	0.42	0.11	-0.10
GradCAM	0.07	**0.37**	0.10	0.39	**0.18**	**0.64**	**-0.02**	**0.23**

Table 14: Faithfulness as *comprehensiveness* and *sufficiency* measured for MUTAG, PROTEINS and ENZYMES dataset using edge masks. Low sufficiency and high comprehensiveness indicates high faithfulness.

Methods	MUTAG		PROTEINS		ENZYMES	
	Suff.	Comp.	Suff.	Comp.	Suff.	Comp.
GNNExp-Edge	0.09	0.14	**0.33**	**0.13**	0.14	0.17
GradEdge	**0.07**	**0.37**	0.41	0.11	**0.13**	**0.19**
Random	0.08	0.17	0.43	0.13	0.17	0.11

Table 15: Results for feature sparsity for node classification.

Methods	Cora				CiteSeer				PubMed			
	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP
HardMask												
Zorro	**2.69**	**3.07**	**4.34**	**3.18**	**2.58**	**2.60**	4.68	**2.78**	**2.55**	**2.58**	**3.21**	**2.86**
SoftMask												
GNNExp	7.27	7.27	7.27	7.27	8.21	8.21	8.21	8.21	6.21	6.21	6.21	6.21
Grad	4.08	4.22	4.45	4.08	4.19	4.28	4.41	4.18	4.41	4.51	4.89	4.46
GradInput	4.07	4.25	4.37	4.08	4.17	4.29	**4.33**	4.17	4.41	4.51	4.92	4.47
SmoothGrad	5.04	5.97	6.88	5.57	6.34	6.92	7.73	6.52	5.83	5.94	6.04	5.85
IG	4.20	4.36	4.83	4.11	4.28	4.43	4.65	4.25	4.28	4.59	5.04	4.38
Random Expl.	7.07	7.07	7.07	7.07	8.08	8.02	8.02	8.02	6.02	6.02	6.02	6.02

Table 4. In Table 17, we compare the RDT-Fidelity and Sparsity on ogbn-arxiv dataset. We evaluate RDT-Fidelity and sparsity over 1000 randomly selected nodes. We do not add Zorro and PGM due to their long run time on the ogbn-arxiv dataset.

H Different strategies to compute hard mask from soft mask to measure Correctness

We report correctness for Cora and CiteSeer in Table 18 and 19 with mean as a threshold for generating hard masks.

In Figure 7, we report token level macro-F1 (ma-F1) and micro-F1 (mi-F1) scores with different topk ∈ [5,10,15,20,25] with different GNN models. Figure 8 and 9 show the correctness for Cora and CiteSeer respectively with different *topk*.

I Limitations

The major limitation of this work is that Bagel only focuses on the *post-hoc explanation* approaches. This work does not cover *interpretable-by-design*

Table 16: Correctness of the explanation for node classification on CiteSeer dataset. We use $k = 20$.

Methods	GCN				GAT				GIN				APPNP			
	P@k	R@k	F1	$\|\mathcal{S}\|$	P@k	R@k	F1	$\|\mathcal{S}\|$	P@k	R@k	F1	$\|\mathcal{S}\|$	P@k	R@k	F1	$\|\mathcal{S}\|$
HardMask																
ZORRO	0.17	0.59	0.24	36	0.31	0.41	0.33	14	0.19	0.53	0.25	35	0.20	0.61	0.25	40
PGM	0.17	0.34	0.23	20	0.18	0.36	0.24	20	0.19	0.39	0.25	20	0.17	0.34	0.23	20
SoftMask																
GNNExp	**0.47**	**0.94**	0.63	20	**0.46**	**0.92**	0.62	20	**0.47**	**0.95**	0.63	20	**0.46**	**0.92**	0.61	20
Grad	0.04	0.08	0.06	20	0.28	0.56	0.38	20	0.23	0.46	0.31	20	0.23	0.46	0.31	20
GradInput	0.02	0.04	0.02	20	0.25	0.50	0.29	20	0.26	0.51	0.34	20	0.09	0.17	0.12	20
IG	0.02	0.04	0.03	20	0.23	0.46	0.31	20	0.37	0.73	0.49	20	0.13	0.25	0.17	20
SmoothGrad	0.01	0.02	0.02	20	0.32	0.64	0.43	20	0.29	0.58	0.38	20	0.06	0.12	0.08	20

Table 17: RDT-Fidelity and Sparsity on ogbn-arxiv dataset.

		RDT-Fidelity				Sparsity			
Mask	Methods	GCN	GAT	GIN	APPNP	GCN	GAT	GIN	APPNP
Soft	GNNExplainer	0.51	0.47	0.28	0.53	5.33	5.25	5.63	5.28
	Grad	0.14	0.17	0.10	0.15	4.23	4.17	5.08	3.78
	GradInput	0.14	0.16	0.11	0.14	4.42	4.43	5.34	3.93
	IG	0.21	0.23	0.18	0.27	4.48	4.42	5.35	3.93
	SmoothGrad	0.21	0.23	0.13	0.27	4.43	4.43	5.34	3.93
	Zero Exp.	0.14	0.17	0.11	0.15	na	na	na	na
	Random Exp.	0.72	0.71	0.54	0.74	11.84	11.84	11.84	11.84

Table 18: Correctness of the explanation on Cora dataset. We use mean as a threshold to generate hard masks. $|\mathcal{S}|$ represents size of the explanations.

Methods	GCN			GAT			GIN			APPNP		
	Pre.	Rec.	$\|\mathcal{S}\|$	Pre.	Rec.	$\|\mathcal{S}\|$	Pre.	Rec.	$\|\mathcal{S}\|$	Pre.	Rec.	$\|\mathcal{S}\|$
HardMask												
ZORRO	0.19	0.80	45	0.25	0.83	40	0.26	0.45	33	0.22	0.79	38
PGM	0.11	0.22	20	0.18	0.36	20	0.18	0.36	20	0.19	0.38	20
SoftMask												
GNNExplainer	0.11	1.00	108	0.11	1.00	109	0.26	1.00	39	0.11	1.00	107
Grad	0.12	0.99	94	0.12	1.00	91	0.24	0.90	38	0.12	0.99	94
GradInput	0.11	1.00	98	0.11	1.00	106	0.26	1.00	39	0.11	1.00	99
IG	0.11	1.00	98	0.11	1.00	100	0.26	1.00	39	0.11	1.00	101
SmoothGrad	0.11	1.00	104	0.11	1.00	102.	0.26	1.00	39	0.11	1.00	102

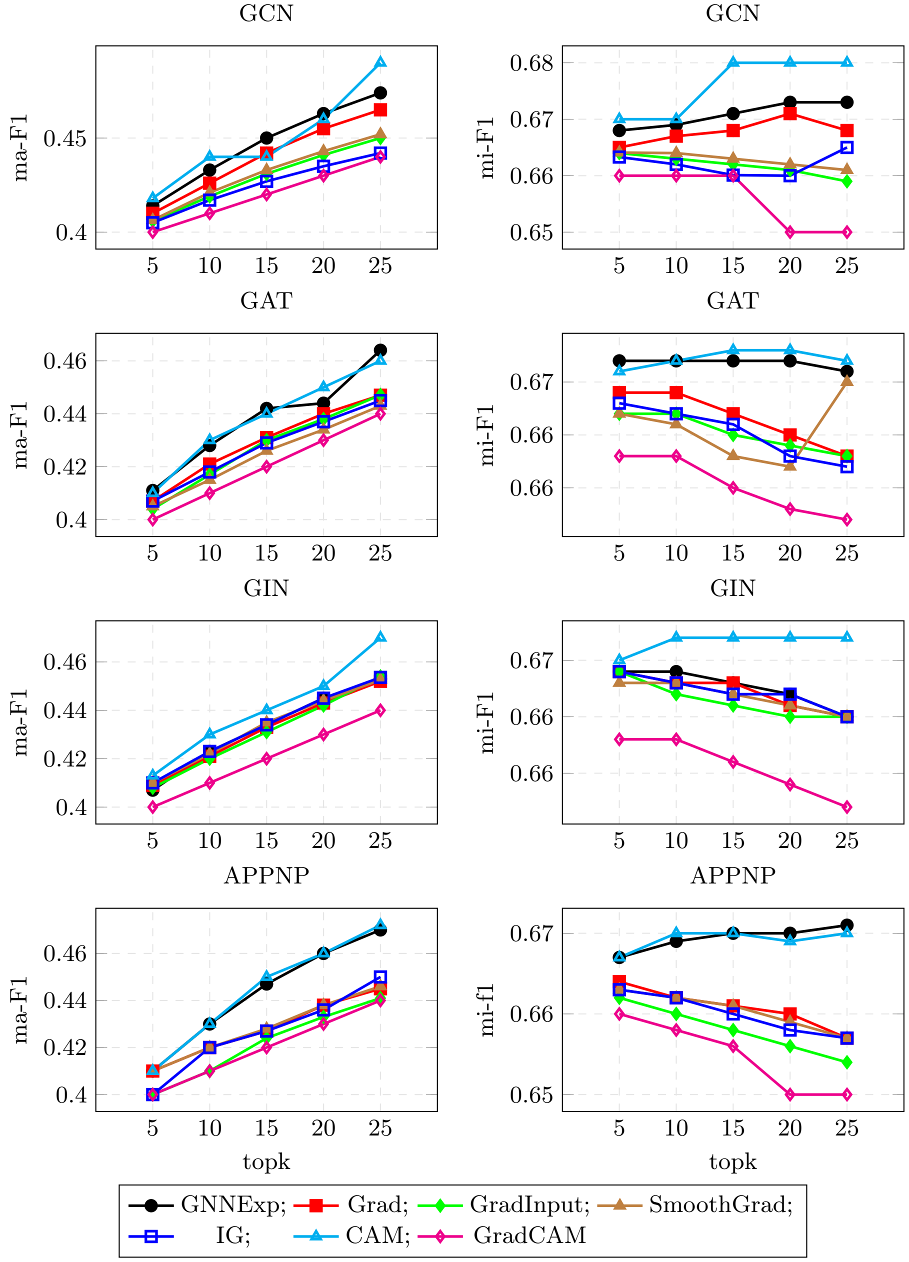


Fig. 7: Plausibility for Movie Reviews dataset.

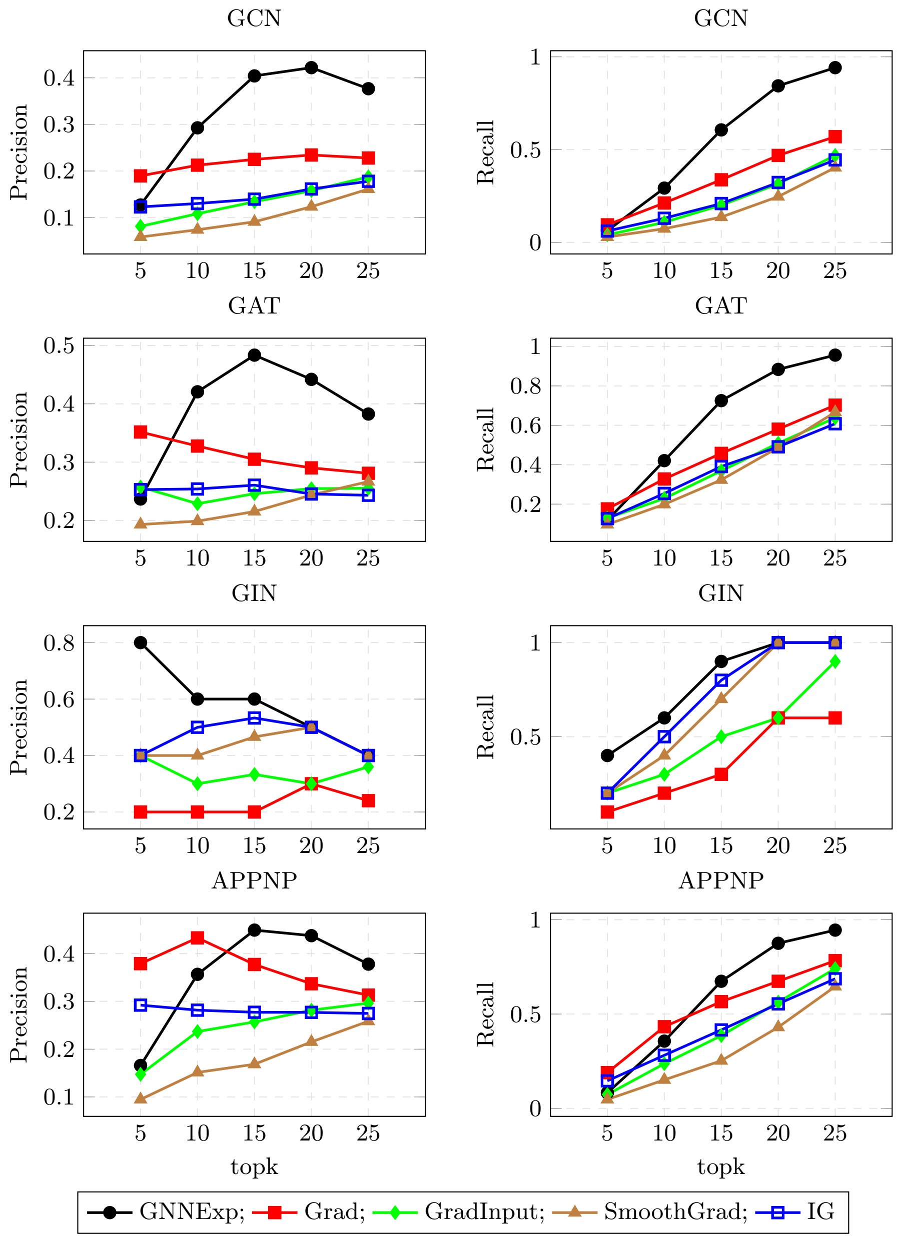


Fig. 8: Correctness for Cora dataset.

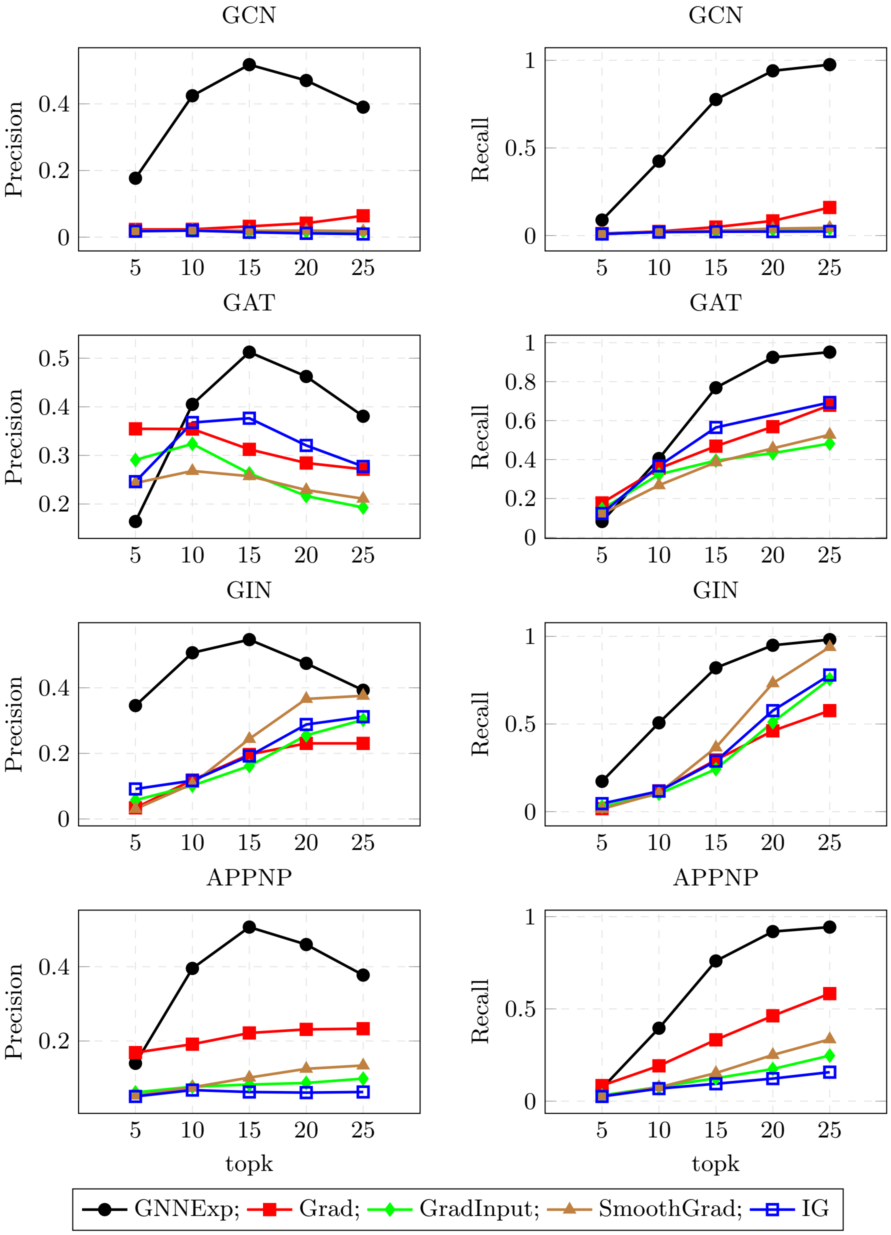


Fig. 9: Correctness for CiteSeer dataset.

Table 19: Correctness of the explanation for CiteSeer dataset. We use mean as a threshold to generate hard masks. $|\mathcal{S}|$ represents size of the explanations.

Methods	GCN			GAT			GIN			APPNP		
	Pre.	Rec.	$\|\mathcal{S}\|$	Pre.	Rec.	$\|\mathcal{S}\|$	Pre.	Rec.	$\|\mathcal{S}\|$	Pre.	Rec.	$\|\mathcal{S}\|$
SoftMask												
ZORRO	0.17	0.59	36	0.31	0.41	14	0.19	0.53	35	0.20	0.61	40
PGM	0.17	0.34	20	0.18	0.36	20	0.17	0.34	20	0.17	0.34	20
HardMask												
GNNExplainer	0.11	1.00	99	0.10	1.00	111	0.19	1.00	60	0.10	1.00	106
Grad	0.11	0.97	92	0.11	0.99	99	0.19	0.98	58	0.11	0.99	97
GradInput	0.11	1.00	95	0.10	1.00	106	0.19	1.00	59	0.11	1.00	102
IG	0.11	1.00	95	0.10	1.00	106	0.19	1.00	59	0.11	1.00	101
SmoothGrad	0.11	1.00	99	0.10	1.00	110	0.19	1.00	60	0.10	1.00	106

and *counter factual based explanation* methods. Though we tried our best to use datasets with varying graph properties and distributions, we believe this benchmark has enormous scope to expand to multiple graph datasets with varying graph properties. We strive to increase the number of more varied datasets in Bagel in the future.

J Run time for training and re-training of GNN models

In Table 20, we report the run time for training and re-training of GNN models for Correctness experiments. For a better understanding of run time for training a GNN model, we also report training time on ogbn-arxiv dataset.

Table 20: Run time in seconds of different GNNs on Cora, CiteSeer and ogbn-arxiv dataset.

Model	Cora		CiteSeer		ogbn-arxiv
	train	re-train	train	re-train	train
GCN	1.09	1.15	1.30	1.90	7.87
GAT	1.14	1.78	1.76	2.11	16.23
GIN	1.49	1.57	1.89	2.01	6.15
APPNP	0.93	1.01	1.10	1.51	6.76

K Details on design of decoys

In principle, the design of decoys is based on the domain and the domain knowledge. But in general, care should be taken to verify that the added decoys are learnt by the models. Typically in graph datasets, homophily plays a significant role in learning neighborhood decision structure. With this in mind, we added the decoys to increase the homophily so that enough incorrectly classified nodes are correctly classified. We verified that this indeed is the case in our experiments. We report the number of correctly classified nodes after adding decoys in Table 21.

Table 21: The number of incorrectly labelled nodes (✗) decreases after addition of different number of decoys. The number of new correctly labelled nodes after injecting decoys is listed under ✓. Here k represents number of decoys.

	Cora				CiteSeer			
Model	✗	✓			✗	✓		
		k=5	k=10	k=15		k=5	k=10	k=15
GCN	88	74	79	81	329	191	229	305
GAT	86	77	85	85	311	256	301	311
GIN	6	6	6	6	56	56	56	56
APPNP	73	62	70	73	280	205	252	264

L Details for training GNNs

We run all our experiments on a server with Intel Xeon Silver 4210 CPU and an NVIDIA A100 GPU. For *node classification* task, we train 2-layer GNNs for 200 epochs. The size of hidden layers for GCN, GAT, GIN, and APPNP are 64, 64, 32, and 64, respectively. We use Adam optimizer with a learning rate of 0.01 and 5e-4 weight decay. We use all citation datasets from Pytorch-Geometric [7]. We follow the semi-supervised setting from [16,39,7] for data split where only 20 nodes per label are used in training and 500 nodes for validation and 1000 nodes for test data. Further details are available in Table 4. For evaluating Correctness, we use the fully supervised split where all the nodes except validation and test data are in training data. We follow data split from the OGB benchmark [12] for ogbn-arxiv dataset.

For the *graph classification* task, we use 2-layer GNNs, and in addition, we use a pooling layer to generate the representation of the graph. Specifically, we use the average pooling layer. Finally, a linear layer is applied to make the prediction over classes. The size of hidden layers for GCN, GAT, GIN, and APPNP are 64, 64, 32, and 64, respectively. We use 16 as the batch size for the training. We do not use any weight decay for graph classification. For all graph datasets except ENZYMES, we use 0.01 learning rate. For ENZYMES, we use 0.001 learning rate. We use 80% of the graphs as the train set and 20% as the test set using the random seed for all molecules dataset. We follow the data split from eraser benchmark [3] for the Movie Reviews dataset.