Multi-objective Explanations of GNN Predictions

Yifei Liu*, Chao Chen†, Yazheng Liu*, Xi Zhang*, Sihong Xie†

*Key Laboratory of Trustworthy Distributed Computing and Service (MoE), BUPT †Computer Science and Engineering Dept, Lehigh University

liuyifei@bupt.edu.cn, chc517@lehigh.edu, {LiuYZ,zhangx}@bupt.edu.cn, xiesihong1@gmail.com

Abstract—Graph Neural Network (GNN) has achieved state-of-the-art performance in various high-stake prediction tasks, but multiple layers of aggregations on graphs with irregular structures make GNN a less interpretable model. Prior methods use simpler subgraphs to simulate the full model, or counterfactuals to identify the causes of a prediction. The two families of approaches aim at two distinct objectives, "simulatability" and "counterfactual relevance", but it is not clear how the objectives can jointly influence the human understanding of an explanation. We design a user-study to investigate such joint effects, and use the findings to design a multi-objective optimization (MOO) algorithm to find Pareto optimal explanations that are well-balanced in simulatability and counterfactual. Since the target model can be of any GNN variants and may not be accessible due to privacy concerns, we design a search algorithm using zero-th order information without accessing the architecture and parameters of the target model. Quantitative experiments on nine graphs from four applications demonstrate that the Pareto efficient explanations dominate single-objective baselines that use first-order continuous optimization or discrete combinatorial search. The explanations are further evaluated in robustness and sensitivity to show their capability of revealing convincing causes, while being cautious about the possible confounders. The diverse dominating counterfactuals can certify the feasibility of algorithmic recourse, that can potentially promote algorithmic fairness where humans are participating in the decision-making using GNN.

I. INTRODUCTION

Graphs represent relations between entities and have been used to model social networks [43], biological networks [56], and online reviews [35]. On prediction tasks on graphs, such as node classification, link prediction, and graph classification [20], [8], [45], [17], GNN exploits the relations to aggregate information in a neighborhood of each node to achieve state-of-the-art predictive performance. However, the aggregations over many nodes multi-hops away make the GNN predictions too opaque to be understood and trusted by humans. Explanations of the GNN predictions try to simplify the computation to deliver societal merits, such as justifying the predictions, fulfilling legal regulation [13], and algorithmic recourse [44], [38], [4]. For example, when warning an online shopper about frauds detected using GNN on a review graph [35], the user may ask "why I am a victim of frauds" and expect an explanation such as "the website you're viewing has connections with certain suspicious IP addresses". We focus on explaining GNN predictions made on graph nodes.

According to [21], "*To explain an event is to provide some information about its causal history*" and the explanation of a prediction can be defined in two ways. First, an explanation can be a causal chain consisting of a forward mapping from inputs and model parameters (the "causes" X) to model

prediction (the "outcome" Y) via steps of computations. An explanation with good simulatability would allow humans to more easily forward simulate the causal chain (possibly a simplified version). Second, an explanation can be in the form of counterfactuals [28]: an event X is said to have caused event Y, if in the counterfactual where X did not happen, Y would not have happened. Counterfactuals allow humans to see the impact of X on Y, and prior works show that humans do counterfactual reasoning in their day-to-day life [28], [5]. We define counterfactual relevance as the amount of change in the probability that Y happens when the cause X is altered. There are additional desiderata. To give humans a better sense of causal relationship, an explanation of an outcome Y should be robust to perturbations irrelevant to the cause X but sensitive to changes in X. Diverse counterfactuals for algorithmic recourse with minimal changes in the decision subjects, e.g., a human on a dating site, allow human agency in the decision-making [44].

Explaining GNN is gaining more attention, and yet there is no study of the interactions between the two metrics, simulatability and counterfactual relevance, from the human and computation perspectives. Gradient-based methods [3], [33] use magnitudes of gradients to highlight important edges or node features. Such methods aim at counterfactuals since the gradients indicate how fast the prediction (the "outcome") changes with respect to small perturbations in the highlighted input (the "cause"). Learning-based explanation methods, including GNNExplainer [49] and GNNLIME [18], extract a simpler surrogate model to faithfully approximate a GNN prediction and thus promote simulatability, without concerning counterfactual relevance. Explanation methods based on gradients [49] need to access the target model as a whitebox and may break privacy and security constraints. See Table I and related work for comparisons.

Inspired by the two modes of human thinking studied in psychology [19], we hypothesize that human perception of an explanation is a function of both metrics. We conjecture a

TABLE I: Prior explanation methods *v.s.* the proposed method.

→Algorithms ↓ Properties	Diverse-CF	Recourse	Convex	DeepLIFT	LIME	GNNLIME	GNNExplainer	Attention	The proposed
Surrogate					*	*	*	*	
Gradient	*			*					
Search		*	*						*
Simulatability	*	*	*	*	*		*	*	*
Counterfactual	*	*		*	*		*		*
Blackbox	*	*							*
Human Evaluation									*

Fig. 1: Simulatability and counterfactual relevance interact.

cognitive process where humans first intuitively make sense of the outcome in a lightweight forward simulation using an explanation (System 1), and then perform more effortful counterfactual reasoning (System 2) to figure out a cause of the outcome. If the explanation is rejected due to low simulatability in the first phase, humans will be less willing to seek for the causes. Fig. 1 shows a pictorial representation of the hypothesis, with four categories of explanations. Gradient-based explanations are in the high counterfactual relevance, low simulatability category (region **A**), and GNNExplainer has no guarantee of high counterfactual relevance but aims to achieve high simulatability (region **D**). Table III in Section V shows the quantitative evaluation of these methods.

To test the above hypotheses on GNN, we adopt simple (small, acyclic, and connected) subgraphs as explanations for forward simulation of a GNN prediction on a node. Each explanation is associated with a counterfactual explanation that has some elements removed from the explanation to flip the prediction. We generate explanations and counterfactuals in the four categories shown in Fig. 1 and measure how simulatability and counterfactual relevance interact to influence human perception of the explanations. Statistical analyses show that: 1) a low simulatability can, but not always, prevent the adoption of an explanation, making counterfactual reasoning less relevant. 2) conditioned on a high simulatability, high counterfactual relevance improves human acceptance of the explanation.

Given the joint effect of the two metrics on humans, current methods do not jointly maximize simulatability and counterfactual relevance. Since the two metrics can be competing and trade-offs are necessary, we define Pareto efficient explanations and formulate a multi-objective optimization problem to model the trade-offs. Since the target model is a blackbox, we design a depth-first search algorithm that accesses the zeroth order information of the model, i.e., the predictions, to identify Pareto efficient subgraph explanations. Explanation search algorithms, such as those based on (mixed) integer programming [44], [38] and subgraph enumeration [52], employ similar searches, and yet they are single-objective optimization. Though less expensive, gradient-based approaches [50] are white-box methods and only find node/edge importance, while the generation of connected graphs still requires exhaustive search. Further, we provide an analysis on the lack of robustness of gradient-based GNN explanations. In the contrast, we empirically verify the robustness and sensitivity of the optimal explaining subgraphs found by the proposed algorithm. Although we strike for causal explanations, we are cautious and formulate GNN using Structural Equation Model (SEM) to prove that confounders can exist in a subgraph explanation and users must be cautioned that the found

counterfactuals are not "the" causes of the predictions. Lastly, we extensively verified that the proposed algorithm dominates single-objective baselines in both metrics on 9 datasets.

II. PROBLEM FORMULATION

Assume that we have a GNN of L layers trained to predict class distributions of the nodes on a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges connecting the nodes. Let $\mathcal{N}(v_i)$ be the set of neighbors of $v_i \in V$. On layer l, $l = 1, \ldots, L$ and for any node v_i, $i = 1, \ldots, |V|$, GNN computes $\mathbf{h}_i^{(l)}$ using messages sent from $\mathcal{N}(v_i)$ to v_i, by the following operations:

$$\mathbf{m}_{ji}^{(l)} = \text{MSG}\left(\mathbf{h}_j^{(l-1)}, \mathbf{h}_i^{(l-1)}\right), \quad (1)$$

$$\mathbf{a}_i^{(l)} = \text{AGG}\left(\left\{\mathbf{m}_{ji}^{(l)} | v_j \in \mathcal{N}(v_i)\right\}\right), \quad (2)$$

$$\mathbf{h}_i^{(l)} = \text{UPDATE}\left(\mathbf{a}_i^{(l)}, \boldsymbol{\theta}^{(l)}\right). \quad (3)$$

The MSG function computes the message vector sent from v_j to v_i (e.g., $\mathbf{m}_{ji}^{(l)} = \mathbf{h}_j^{(l-1)}$). The AGG function aggregates the messages sent from all $v_j \in \mathcal{N}(v_i)$ to v_i and can be the element-wise sum, average, or maximum of the messages. The UPDATE function uses parameter $\boldsymbol{\theta}^{(l)}$ to map $\mathbf{a}_i^{(l)}$ to $\mathbf{h}_i^{(l)}$. One example is $\mathbf{h}_i^{(l)} = (\boldsymbol{\theta}^{(l)})^\top \mathbf{a}_i^{(l)}$, followed by some non-linear mapping such as ReLU. The input node feature vector $\mathbf{x}_i$ for v_i is regarded as $\mathbf{h}_i^{(0)}$. The output of the GNN on node v_i is $\mathbf{h}_i^{(L)}$, which can be softmaxed to the node class distribution $\mathbf{y}_i$ (a vector of class probabilities). The parameters of GNN, $\boldsymbol{\theta}^{(l)}$, $l = 1, \ldots, L$, are trained end-to-end on labeled nodes on G. We define an explanation of the prediction $\mathbf{y}_i$ to be a subgraph G_i of G that contains the target node v_i [49]. Besides being agnostic to the above details of architecture and parameters, we desire the following properties of the explanations.

Simulatability. A comprehensible explanation should be simulatable, defined by the following two aspects. The simplicity of an explanation is related to the limit of human cognitive bandwidth [27] and sparsity is used as a proxy of simplicity [10], [14], [49]. We say that the explaining subgraph G_i is C-sparse if G_i contains no more than C nodes. Due to the sparsity, G_i does not allow full computation taken on the full graph G, and the *faithfulness* of G_i measures how much the G_i can reproduce $\mathbf{y}_i$ generated on G. Similar to [42], we measure faithfulness using the symmetric KL-divergence between the prediction $\mathbf{y}_i$ on G and $\mathbf{y}_i'$ on G_i (the larger, the better):

$$\nu(G_i) = -(\text{KL}(\mathbf{y}_i || \mathbf{y}_i') + \text{KL}(\mathbf{y}_i' || \mathbf{y}_i)) \leq 0. \quad (4)$$

Counterfactual relevance. Let the above-defined subgraph G_i be a "fact". A counterfactual $\tilde{G}_i$ of G_i is a perturbation of G_i. We restrict the counterfactual to be a strict subgraph of G_i. Let the difference between G_i and $\tilde{G}_i$ be denoted by $\Delta(G_i, \tilde{G}_i)$, the size of which is represented by $|\Delta(G_i, \tilde{G}_i)|$, so that $G_i = \tilde{G}_i + \Delta(G_i, \tilde{G}_i)$ means adding $\Delta(G_i, \tilde{G}_i)$ to $\tilde{G}_i$ reconstructs G_i. The class distributions of v_i generated by the target GNN model on G_i and $\tilde{G}_i$ are denoted by $\mathbf{y}_i'$ and $\tilde{\mathbf{y}}_i'$, respectively. We define the counterfactual relevance [28] of the tuple $(G_i, \tilde{G}_i)$ when explaining $\mathbf{y}_i$ as

$$\mu(G_i, \tilde{G}_i) = \frac{1}{|\Delta(G_i, \tilde{G}_i)|}(\nu(G_i) - \nu(\tilde{G}_i)). \quad (5)$$
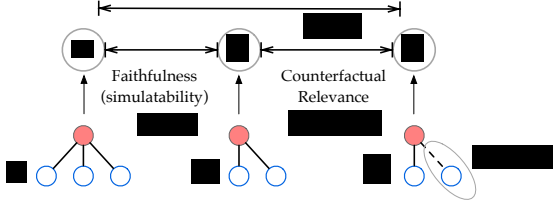
Fig. 2: Two explanation metrics. $\mathbf{y}_i$ is the GNN prediction of v_i on the full graph G, $\mathbf{y}'_i$ is the GNN prediction on G_i, and $\tilde{\mathbf{y}}'_i$ on $\tilde{G}_i$. Faithfulness is measured by Eq. (4). The smaller the $|\nu(G_i)|$, the more faithful. $\Delta(G_i, \tilde{G}_i)$ is circled by the dashed line. The larger the $|\mu(G_i, \tilde{G}_i)|$ (Eq. (5)), the more counterfactual relevance.

$\mu(G_i, \tilde{G}_i)$ can be positive, negative or zero. Because $\nu(G_i)$ represents the faithfulness, the absolute $|\mu(G_i, \tilde{G}_i)|$ measures the change in the class distribution of v_i approximated by the fact G_i and the counterfactual $\tilde{G}_i$. When $|\mu(G_i, \tilde{G}_i)|$ is large, the portion $\Delta(G_i, \tilde{G}_i)$ removed from G_i is likely to be the cause of $\mathbf{y}'_i$ [15]. The normalizer $|\Delta(G_i, \tilde{G}_i)|$ makes sure that the same difference $\nu(G_i) - \nu(\tilde{G}_i)$ caused by a small $\Delta(G_i, \tilde{G}_i)$ will be more desirable than that caused by a larger $\Delta(G_i, \tilde{G}_i)$. It also prohibits extreme counterfactuals that remove all nodes except the target v_i. These quantities are demonstrated in Fig. 2.

III. HOW HUMANS PERCEIVE EXPLANATIONS

> "System 1 operates automatically and quickly ... System 2 allocates attention to the effortful mental activities ..."
>
> *Daniel Kahneman, Nobel laureate*

We conducted a human subject study to find the roles of the two metrics in the human perception of explanations. The two modes of thinking, System 1 and System 2, are extensively studied in psychology, as quoted above. We conjecture that forward simulations help humans quickly screen an explanation using System 1, while reasoning using the counterfactual is a more deliberate process that requires System 2, so that humans will conduct counterfactual reasoning only after the explanation has passed System 1 screening. Simulatability and counterfactual relevance measure how well an explanation and an associated counterfactual are received by the two Systems.

According to Fig. 1, on the Cora dataset, we sample five target nodes and for each node we generate subgraphs with low and high simulatability. This leads to ten explaining subgraphs for each subject to evaluate the simulatability. For each explaining subgraph G_i, we further generate two counterfactuals $\tilde{G}_i$ that are subgraphs of G_i, with different counterfactual relevance. Fig. 3 shows one sample test case. For each of the five nodes, a subject will see the original graph G where GNN produced the prediction $\mathbf{y}_i$, the explanation G_i that produced $\mathbf{y}'_i$, and two counterfactuals that generate two $\tilde{\mathbf{y}}'_i$. The full graph is considered to be too complicated for interpretation, while G_i is more intelligible. The two counterfactuals allow a subject to evaluate if a removed part $\Delta(G_i, \tilde{G}_i)$ is a plausible cause of the prediction $\mathbf{y}'_i$. For each graph, we color the nodes based on the GNN's prediction, so that a subject can relate a prediction to the neighbors. We show the predicted class
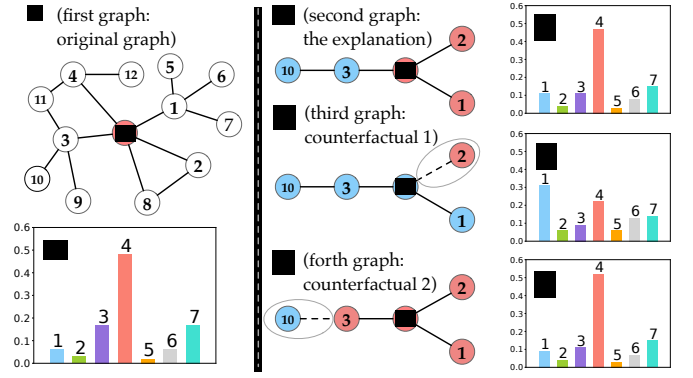
Fig. 3: Sample explanations in the human study. Each node is a paper on Cora. *Left:* the large graph containing v_i, whose prediction is to be explained. Predicted distributions over 7 classes are shown in histograms. *Right:* subgraphs explaining the prediction of v_i, along with the class distributions predicted on the individual subgraphs (top/middle: explanation/counterfactual found by GNN-MOExp, bottom: a counterfactual with a small counterfactual relevance. Counterfactuals are constructed by removing the dashed edges).

distributions in histograms, so that the predictions across the (sub)graphs can be compared conveniently. The subjects were not told about the two metrics of the explanations but needed to understand, analyze, and then rate the explanations.

To avoid bias, we frame the survey as an evaluation of a graph-based search engine and recruited subjects with search experience using Google Scholar. The authors of this paper are excluded. The two counterfactuals are randomly ordered. Each subject is further trained on two additional sample cases. During the test phase, we ask subjects the following questions after each test case and collect feedback (r_* in the parentheses) in a 5-point Likert scale *(1-very little (won't accept),2-little,3-not sure,4-a little, 5-very well)*:

a. Simulatability (r_a): How well do you think the **second** subgraph is reproducing the prediction computed in the **first** graph?

b. Counterfactual-1 (r_b): How much do you think the removed component in the **third** subgraph is an important factor leading to the histogram for the second subgraph, had it not been removed?

c. Counterfactual-2 (r_c): Same as above but replace the the **third** subgraph with the **forth** subgraph.

d. Explanation acceptance (r_d): How much will you accept the probabilities, if they were computed on the second subgraph rather than the first?

A. Analysis of human feedback

The questions quantitatively reveal the human perception of the two explanation metrics. Let the responses to the questions a, b, c, and d be r_a, r_b, r_c, and r_d, respectively. r_a measures the subject's perceived simulatability of the explanation G_i. The difference between r_b and r_c measures the preference of a subject between two alternative counterfactuals $\tilde{G}_i$. r_d measures the subject's overall acceptance of G_i as an explanation based on its simulatability and the plausibility of the causes found using the counterfactuals. After filtering out an obvious outlier (the responses to all questions are the same),
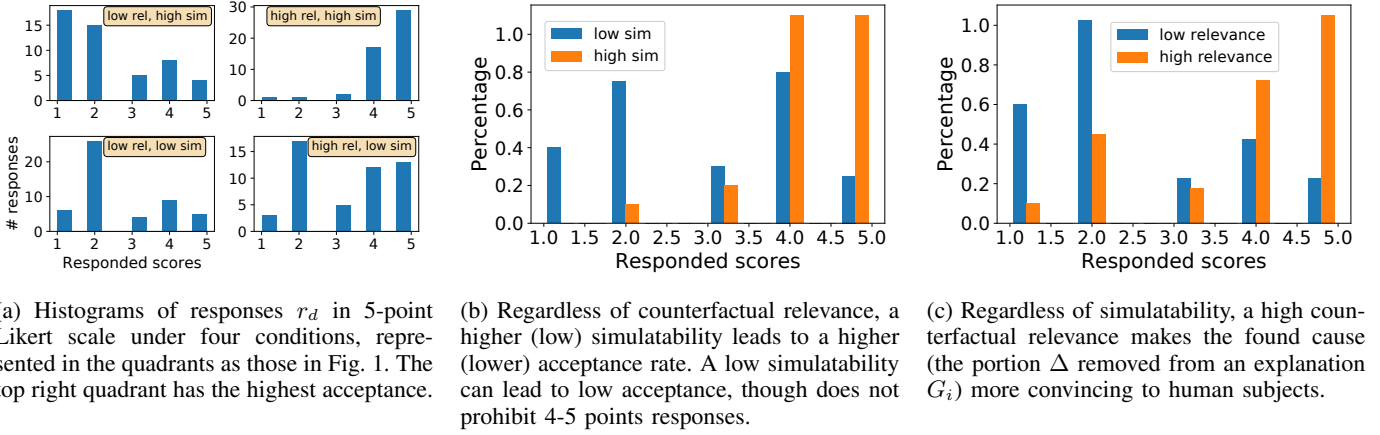
(a) Histograms of responses r_d in 5-point Likert scale under four conditions, represented in the quadrants as those in Fig. 1. The top right quadrant has the highest acceptance.

(b) Regardless of counterfactual relevance, a higher (low) simulatability leads to a higher (lower) acceptance rate. A low simulatability can lead to low acceptance, though does not prohibit 4-5 points responses.

(c) Regardless of simulatability, a high counterfactual relevance makes the found cause (the portion Δ removed from an explanation G_i) more convincing to human subjects.

Fig. 4: User study. Fig. 4a and 4b show that simulatability is necessary but not sufficient for explanation acceptance.

we have 10 subjects' responses to 10 test cases, leading to 100 scores for each of the four questions. We draw the following conclusions based on statistical analyses.

High simulatability helps acceptance that can be boosted by high counterfactual relevance. Using responses r_d, a two-way analysis of variance (ANOVA) shows that the two metrics interact strongly (p-value < 0.00001). Fig. 4a and 4b confirm that a high simulatability is a prerequisite of explanation acceptance, with high counterfactual relevance being the second condition. A low simulatability leads to more mixed acceptance, regardless of counterfactual relevance. There are some numbers of acceptance with low simulatability, due to the subjects' in-depth analysis of the cases that leads to a final acceptance.

Simulatability can predict acceptance of explanations. We conducted a t-test on the responses r_d from two groups: one has cases with low simulatability and the other has cases with high simulatability. The p-value is almost zero, indicating that the degree of acceptance differs significantly between the groups. The t-statistic is -6.4. After taking into account the within-group variances and the sample size, we conclude that the acceptance of a less simulatable explanation is less than that of a more simulatable explanation. Fig. 4b further confirm this conclusion.

A higher counterfactual relevance makes a reason more likely perceived as "the cause". While there can be several factors that jointly lead to the GNN prediction $\tilde{\mathbf{y}}_i$, humans tend to accept the one with high counterfactual relevance as "the cause", compared to those with low counterfactual relevance. We conducted a t-test between the responses r_b and r_c. The tests show that a higher counterfactual relevance is more convincing (all p-value < 0.01), regardless of simulatability (see Fig. 4c). However, when simulatability is low, the presented "cause" is less convincing (see bottom two subfigures of Fig. 4a). **Caution**: "the cause" presented by a counterfactual may not be the only or the true cause of the prediction $\tilde{\mathbf{y}}_i$, due to confounders. See Section IV-B.

IV. MULTI-OBJECTIVE EXPLANATIONS OF GNN

Given the human study results, we aim to solve the following multi-objective optimization problem.

Definition 1. *Given a graph G and a GNN model $\boldsymbol{\theta}$, on any target node $v_i \in V$, extract an explanation subgraph $G_i \subset G$ and a counterfactual subgraph $\tilde{G}_i$, where $v_i \in \tilde{G}_i \subset G_i$, G_i contains no more than C nodes and is acyclic, so that $\nu(G_i)$ and $|\mu(G_i, \tilde{G}_i)|$ are maximized:*

$$\max_{G_i, \tilde{G}_i} \quad F(G_i, \tilde{G}_i) = (\nu(G_i), |\mu(G_i, \tilde{G}_i)|) \tag{6}$$
$$\text{s.t.} \quad v_i \in \tilde{G}_i \subset G_i \subset G, \ |G_i| \leq C, \ G_i \text{ acyclic},$$

For simplicity of the explanation, we restrict G_i to contain no more than C nodes [27]. The limit to C nodes also reduces the degree, coreness, and centrality of any nodes in G_i, and improves human reaction time when reasoning with G_i [24]. We restrict the explanations to be acyclic graphs [46], since a cycle can lead to self-proof and explanations such as "*Alice is a database researcher because she cited a paper of Bob, who is a database researcher since he cited Alice's paper*".

The optimization is bi-objective and the objective vector function F has two scalar objectives. We don't use a single scalar objective function, such as $\nu(G_i) + \lambda|\mu(G_i, \tilde{G}_i)|$, not only because that λ can be hard to specify, but also that trading one objective for the other is not desirable according to the human subject study (either low simulatability or counterfactual relevance suppresses human acceptance of the explanation and the counterfactual). Beyond being multi-objective, the solution space of all possible G_i, defined by the constraints in the above optimization problem, is exponentially large and discrete and no polynomial-time algorithm is known to search the space. The gradient-based methods in [49], [33] and the search-based methods in [38], [44], [30], [53], [46] can only maximize one of the objective functions and do not guarantee Pareto optimality, i.e., efficient trade-off between objectives. We follow the search-based explanation generation paradigm, but aim at finding the Pareto front and selecting one particular Pareto efficient explanation with well-balanced objectives.

A. Search for Pareto optimal explanations

The algorithm, GNN-MOExp (Graph Neural Network Multi-Objective Explanations) is shown in Fig. 5. We first apply a depth-first search (DFS) to explore the space of
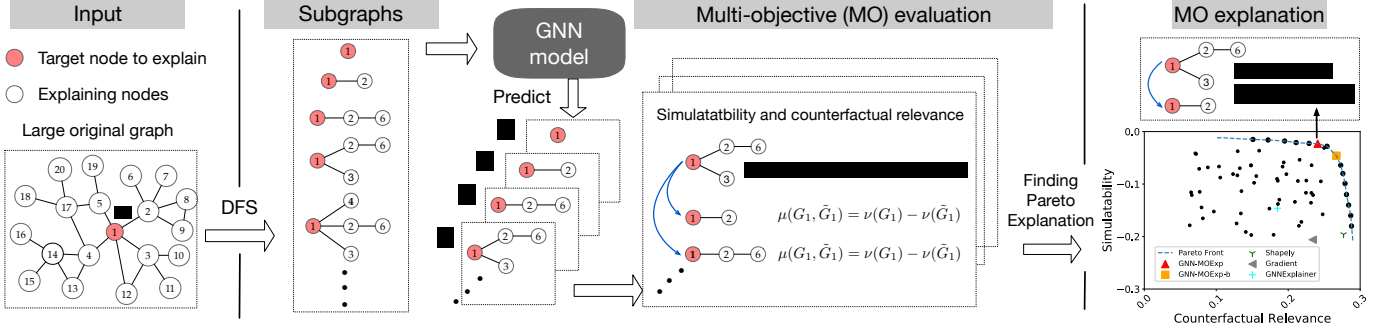
Fig. 5: The workflow of finding Pareto optimal GNN explanations with high simulatability and counterfactual relevance. The subgraphs are enumerated by DFS and the two objectives are computed on the enumerated explanations and their counterfactuals $(G_i, \tilde{G}_i)$ for each node v_i to be explained. Pareto optimal explanation $(G_i^*, \tilde{G}_i^*)$ that are high (but not necessarily the highest) in both metrics are selected.

subgraphs G_i for v_i. Since the prediction $\mathbf{y}_i$ of the target v_i does not depend on nodes that are more than L hops away from v_i, the search is restricted to the dependent neighbors. A canonical ordering of the edges is determined by a breadth-first search (BFS) before running the DFS, ensuring no subgraph will be enumerated more than once. The BFS also canonically numbers the nodes to avoid isomorphism test during graph lookup: the same graph will be represented by a unique array of edges with canonical node numbering. Starting from the subgraph containing only v_i, the DFS expands the subgraph by adding an un-visited edge adjacent to the current subgraph. The constraints in Eq. (6) are used in pruning the search space. After all valid candidate subgraphs containing the edge have been explored, the edge is flagged and will not be visited in future. The enumeration will be completed when all edges within the neighborhood are processed.

The GNN model has to be run on each enumerated subgraph G_i and the two metrics $\nu(G_i)$ and $\mu(G_i, \tilde{G}_i)$ are computed by Eq. (4) and Eq. (5). Since G_i contains at most C nodes, the cost is low. To avoid repetitive calculation of $\nu(G_i)$ when calculating $\mu(G_i, \tilde{G}_i)$, a hash table is used to record $\nu(G_i)$ for each subgraph. G_i becomes a counterfactual of all subgraphs that are the descents of G_i in the DFS search tree.

After evaluating each subgraph and its counterfactuals, we need to find the optimal explanation so that both metrics are high. However, the two metrics can be competing and it is hard to find an explanation that outperforms all others in *both* metrics. We aim to find Pareto optimal (efficient) explanations, that are optimal in the sense that it cannot be outperformed by another explanation in *both* metrics [26]. We need the following definitions.

Definition 2. *(Pareto dominance) Let $F_1(G) = \nu(G)$ and $F_2(G) = |\mu(G, \tilde{G})|$. $\tilde{G}_1 \subset G_1 \subset G$, $\tilde{G}_2 \subset G_2 \subset G$. If $(G_1, \tilde{G}_1)$ Pareto dominates $(G_2, \tilde{G}_2)$, then $\forall i \in \{1,2\}, F_i(G_2, \tilde{G}_2) \leq F_i(G_1, \tilde{G}_1) \wedge \exists i \in \{1,2\}, F_i(G_2, \tilde{G}_2) < F_i(G_1, \tilde{G}_1)$, denoted as $(G_2, \tilde{G}_2) \prec (G_1, \tilde{G}_1)$.*

Definition 3. *(Pareto optimality). $(G_1, \tilde{G}_1)$ is Pareto optimal if and only if $\nexists (G_2, \tilde{G}_2) \prec (G_1, \tilde{G}_1)$.*

Definition 4. *(Pareto optima) The set of all Pareto optimal solutions: $P_s := \{(G_1, \tilde{G}_1) | \nexists (G_2, \tilde{G}_2), (G_1, \tilde{G}_1) \prec (G_2, \tilde{G}_2)\}$.*

Definition 5. *(Pareto optimal front). The set consists of the*

function values of the Pareto optimal set: $P_F := \{F(G_i, \tilde{G}_i) \mid (G_i, \tilde{G}_i) \in P_s\}$.

However, explanations $(G_i, \tilde{G}_i)$ on the Pareto front can be low in one objective while being high in another, and is thus not useful. We design a simple method to find Pareto optimal explanations that are: 1) dominating other explanations, and 2) likely simultaneously optimal in individual metrics (without guarantee). In particular, we sort the explanations and their counterfactuals $(G_i, \tilde{G}_i)$ along the simulatability and counterfactual relevance, independently. Let the ranking position of $(G_i, \tilde{G}_i)$ in the two rankings be denoted by $r_1(G_i, \tilde{G}_i)$ and $r_2(G_i, \tilde{G}_i)$ (the smaller the better). We define the comprehensive ranking $R(G_i, \tilde{G}_i)$ be

$$R(G_i, \tilde{G}_i) = r_1(G_i, \tilde{G}_i) + r_2(G_i, \tilde{G}_i). \tag{7}$$

Finally, we select the $(G_i, \tilde{G}_i)$ with the best comprehensive ranking, denoted by $(G_i^*, \tilde{G}_i^*)$ as the final explanation.

One possible baseline is to use the so-called preference vector to select a Pareto optimal solution that satisfies some weighted balance between the objectives [25]. We found this method hard to use in our case: the two objectives are of different ranges, which vary across different target nodes. In contrast, the ranking-based approach handles the heterogeneity. We did not present this baseline since it significantly underperforms our method. A more competitive baseline is to find $(G_i, \tilde{G}_i)$ whose rankings in the two objectives are well balanced. We compare our approach with this baseline in the experiments. Since the Pareto front is non-convex and contains dents that have well-balanced but low objective values, the above baseline may not work well.

The explanation chosen by the comprehensive ranking is in the Pareto front, as shown by the following theorem.

Theorem 6. *The ranking-based method finds a solution $(G_i^*, \tilde{G}_i^*)$ that's on the Pareto front.*

Proof. If $(G_i^*, \tilde{G}_i^*)$ is not a Pareto optimal solution, then there is $(G_i, \tilde{G}_i)$ that dominates $(G_i^*, \tilde{G}_i^*)$. By definition, $(G_i, \tilde{G}_i)$ must be ranked higher than $(G_i^*, \tilde{G}_i^*)$ in at least one objective, while in the other objective the two are at least equal. According to the definition of comprehensive ranking, $R(G_i, \tilde{G}_i) < R(G_i^*, \tilde{G}_i^*)$ and $(G_i, \tilde{G}_i)$ would have been chosen by the explanation selection algorithm. $\square$

Complexity of the Algorithm. Regarding the DFS, in the best case, v_i is on one end of a linear chain and the time complexity is $O(1)$. In the worst case, the number of subgraphs of a complete graph with n nodes is exponential, and the complexity is $O(e^n)$. Many real-world graphs are sparse and the complexity is more likely to be polynomial. The depth of GNN L is usually limited (≤ 3) due to the over-smoothing effect of aggregation [22] and the number of nodes searched depends on the size of the L-hop neighborhood of the target node. We show in Fig. 9 that the running time of the subgraph search is practically low.

It seems that one has to find the Pareto front and then use the comprehensive ranking to find the best explanation. To eliminate all dominated solutions, the time complexity is quadratic in the number of enumerated subgraphs. However, Theorem 6 says that the comprehensive ranking already points to a solution on the Pareto front and the overall time complexity is just linear in the number of enumerated subgraphs, using the heap data structure.
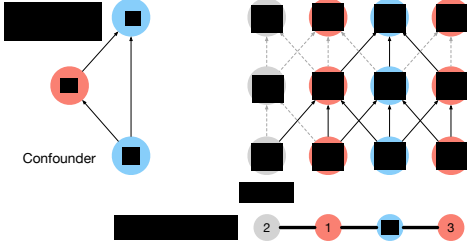
Fig. 6: Confounders in GNN. *Left*: a structural causal model with F being the common cause of both X and Y, and Z is a confounder that makes $P(Y|do(x)) \neq P(Y|X = x)$. *Right*: at the bottom, G_i explains the prediction $h_i^{(2)}$ on node v_i and node 2 can be removed from G_i as an intervention to obtain a counterfactual explanation $\tilde{G}_i$. Above G_i is a computation graph that represents the rollout of the structural equations (1)-(3). Arrows are dependencies among the nodes on the computational graph and dashed lines are not relevant to $h_i^{(2)}$. The variable $h_i^{(0)}$ is a common cause of $h_1^{(1)}$ and $h_i^{(2)}$ and therefore confounds the effect of $h_2^{(0)}$ on $h_i^{(2)}$ through $h_1^{(1)}$. There are other confounders, and the effect of the intervention on $h_i^{(2)}$ should be adjusted for all confounders.

B. Confounders

Confounders are variables that impact both causes and outcome [31]. Fig. 6 shows the concepts of confounder that leads to the Back-Door adjustment:

$$P(Y|do(x)) = \sum_z P(Y|X = x, Z = z)P(Z = z), \quad (8)$$

which is in general not the same as $P(Y|X = x)$. For G_i in the figure, the counterfactual explanation $\tilde{G}_i$ is obtained by the intervention of removing $\Delta = \{2\}$ from G_i. Humans may think that Δ is "the cause" of the output h_i^2. However, this is not true due to confounders, as shown in Fig. 6.

C. Connection to Shapley values

There is a close relationship between counterfactual explanations and Shapley values [39], [7]. As an explanation,

Shapley values are the importance of the factors contributing to the predictions to be explained. One can consider the portion Δ removed from a subgraph G_i as a contributor, and by averaging Δ's contributions over all possible G_i that contain Δ (denoted by $\mathcal{S}(\Delta; G)$), we obtain the Shapley value of Δ:

$$\text{SV}(\Delta) := \frac{1}{|\mathcal{S}(\Delta; G)|} \sum_{G_i \in \mathcal{S}(\Delta; G)} \mu(G_i, G_i - \Delta). \quad (9)$$

The contribution $\mu(G_i, G_i + \Delta)$ follows the definition of Shapley values and can be positive, negative, or zero. Instead, counterfactual relevance $|\mu(G_i, G_i - \Delta)|$ is always non-negative and gives the magnitude of the importance of Δ.

D. Robustness and sanity check of explanations

An accurate explanation of a prediction should vary according to the underlying mechanism that generates the prediction [1], and should remain the same under irrelevant perturbations [11].
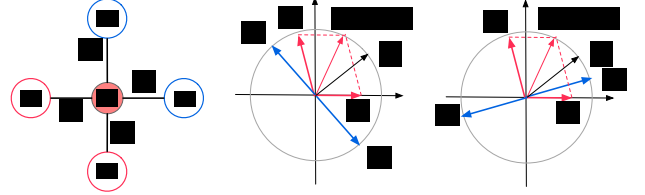

Fig. 7: Manipulate a GNN explanation. *Left*: original graph. *Center*: messages $\mathbf{h}_1$ and $\mathbf{h}_2$ cause the prediction on v_i, while $\mathbf{h}_3$ and $\mathbf{h}_4$ are irrelevant. *Right*: $\mathbf{h}_3$ and $\mathbf{h}_4$ are rotated to perturb a gradient-based explanation, though the prediction of class y remains the same.

Definition 7. *The robustness of a subgraph explanation G_i is the degree of the change in G_i under perturbations that are irrelevant to the mechanism that generates $\mathbf{y}_i$.*

We assume a one-layer GNN ($L = 1$) with parameter $\boldsymbol{\theta} \in \mathbb{R}^{K \times d}$, where K is the total number of classes to be predicted and d is the number of features of the nodes. We use the graph in Fig. 7 *Left* to demonstrate the difference in the robustness of explanations found by GNN-MOOExp and prior gradient-based methods. Gradient-based methods [33], [49], [3] find explanations using the gradient of the following faithfulness loss function with respect to a mask $M \in [0, 1]^{|V| \times |V|}$ over the adjacency matrix $A \in \{0, 1\}^{|V| \times |V|}$:

$$\ell(M; G, \boldsymbol{\theta}) = -\sum_{y=1}^{K} \mathbb{1}[y_i = y] \log P(y|A \odot M; \boldsymbol{\theta}),$$

$$P(y|A \odot M; \boldsymbol{\theta}) = \text{softmax}\left(\sum_{j=1}^{n} M_{ij} A_{ij} \mathbf{h}_j^\top \boldsymbol{\theta}_y\right),$$

where $\boldsymbol{\theta}_y$ is the y-th row of $\boldsymbol{\theta}$. As we are explaining a GNN prediction, $y_i = y$ is the predicted class and not necessarily the ground truth class of v_i. $\mathbf{h}_j$ is the input feature vectors of the neighbor v_j of v_i. The target GNN model will set all entries of M to 1 so that all neighbors of v_i are retained. In Fig. 7 *center*, the neighbors' features satisfy $\mathbf{h}_3 = -\mathbf{h}_4$ so that the relevant neighbors to $\mathbf{y}_i$ are just v_1 and v_2, with representations $\mathbf{h}_1$ and $\mathbf{h}_2$, whose sum is closer to $\boldsymbol{\theta}_y$ than to $\boldsymbol{\theta}_{y'}$ for any $y' \neq y$. The gradient of ℓ w.r.t. M_{ij} is

$$(\mathbb{1}[y_i = y] - P(y))A_{ij}\boldsymbol{\theta}_y^\top \mathbf{h}_j. \quad (10)$$

The importance of the edge (i, j) is the magnitude of the above gradient, essentially determined by the correlation between $\boldsymbol{\theta}_y$ and $\mathbf{h}_j$. In Figure 7 *center*, since both $\mathbf{h}_3$ and $\mathbf{h}_4$ are orthogonal to $\boldsymbol{\theta}_y$, gradient-based methods will never have v_3 and v_4 in their explanations. When $\mathbf{h}_3$ and $\mathbf{h}_4$ are rotated so that $\mathbf{h}_3$ is more similar to $\boldsymbol{\theta}_y$ than $\mathbf{h}_2$ while $\mathbf{h}_3 = -\mathbf{h}_4$ remains, the gradient-based explanation will include v_3, even the prediction remains the same. The rotations are irrelevant to how $\mathbf{h}_1 + \mathbf{h}_2$ leads to the prediction y_i. On the other hand, $\boldsymbol{\theta}_y$ is closer to $\mathbf{h}_1 + \mathbf{h}_2$ than to $\mathbf{h}_1 + \mathbf{h}_3$ or $\mathbf{h}_2 + \mathbf{h}_3$ if only subgraphs with three nodes ($C = 3$) are allowed. As a result, GNN-MOExp still finds the same optimal subgraph containing v_i, v_1, and v_2, even after the rotations and is thus more robust.

Another aspect is that an explanation should faithfully reflect how a *changing* $\mathbf{y}_i$ is generated and is different from simulatability that focuses on explaining a *static* mechanism that generates a fixed $\mathbf{y}_i$. Formally,

Definition 8. *A sanity check of an explanation G_i of a GNN model's prediction $\mathbf{y}_i$ verifies if G_i changes when the mechanism that generates $\mathbf{y}_i$ changes.*

A sanity check is a necessary (but not a sufficient) condition for an explanation to be a faithful surrogate of the full model: not passing the sanity check indicates that an explanation is not reflecting the input-output relationship encoded by the GNN. When debugging a GNN model to identify whether the model or the graph data are manipulated or polluted, passing the sanity check means the explanations can reveal the malicious attacks to the model or data. The prior work [1] proposed a sanity check for deep neural networks on images and does not address sanity checks for GNN on graphs. We conduct sanity checks for GNN-MOExp in Section V-C.

V. EXPERIMENTS

TABLE II: Nine networks from four application domains.

Datasets	Classes	Nodes	Edges	Edge/Node	Features
Cora	7	2,708	10,556	3.90	1,433
Citeseer	6	3,321	9,196	2.78	3,703
PubMed	3	1,9717	44,324	2.24	500
Musae-F	4	2,2470	342,004	15.22	4,714
Musae–G	2	37,700	578,006	15.33	4,005
Amazon-C	4	13,752	574,418	41.77	767
Amazon-P	6	7,650	287,326	37.56	745
Coauthor-C	13	18,333	327,576	17.87	6,805
Coauthor-P	2	34,493	991,848	28.76	8,415

A. Datasets and Baselines

Datasets and experimental settings. We drew real-world datasets from four applications for the node classification task. The dataset details are provided in the supplement.

- In citation networks (Citeseer, Cora, PubMed) [20], each paper has bag-of-words features, and the goal is to predict the research area of each paper.
- We adopt Musae-Facebook (Musae-F) and Musae-Github (Musae-G) [36] from social networks. Nodes represent official Facebook pages (or Github developers), and edges

are mutual likes (or followers) between nodes. Node features are extracted from site descriptions (or developer's location, repositories starred, employer).
- Amazon-Computer (Amazon-C) and Amazon-Photo (Amazon-P) [40] are segments of the Amazon co-purchase graph, where nodes represent goods, edges indicate that two goods are frequently bought together, and node features are the bag-of-words representation of product reviews.
- Coauthor-Computer and Coauthor-Physics are co-authorship graphs based on the Microsoft Academic Graph from the KDD Cup 2016. We represent authors as nodes, that are connected by an edge if they co-authored a paper [40]. Node features represent paper keywords for each author's papers.

We randomly divide each graph into three portions with a ratio of *training : validation : test = 50 : 20 : 30*. The GNN is trained on the training set and all explanation methods are evaluated on the test set.

Baselines. We adopt the following baselines that generate subgraph explanations. Except the baseline Shapley, all baselines compute the weights of edges in the neighborhood of the target node v_i. The explanation G_i is generated by iteratively adding edges adjacent to the current subgraph until C nodes are included in G_i. The edges with higher weights will be considered first. The counterfactual $\tilde{G}_i$ of the baselines are generated in the same way as GNN-MOExp by trying different enumerated subgraphs. $\nu(G_i)$ and $\mu(G_i, \tilde{G}_i)$ of G_i for each baseline are calculated by Eq. (4) and Eq. (5). We describe the details of the baseline:

- **Random** (RND) assigns random weights to edges.
- **Embedding** (EMB) uses DeepWalk [32] to embed the nodes, and the weight of an edge is calculated based on the cosine similarity between the embeddings of two nodes.
- **Gradient** (Grad) [3] use the magnitudes of gradients of GNN output w.r.t. edges to find salient subgraphs.
- **GAT** [45] learns attention weights over neighbors of any node for message aggregation to predict the output of the GNN on v_i, and the attention weights on the edges are extracted as edge weights.
- **GNNExplainer** (GNNExp) [49] learns to mask edges so that the masked graph maximally preserve the predictions of $\mathbf{y}_i$, and the mask matrix provides the edge weights.
- **PGExplainer** (PGExp) [23] trains a deep neural network to parameterize the generation of explanations. The subgraphs generated by the explainer are evaluated.
- **Shapley** picks Δ and $G_i \in \mathcal{S}(\Delta; G)$, defined in Eq. (9), with the highest counterfactual relevance, and use the selected Δ and G_i to generate the counterfactual $\tilde{G}_i$.
- **GNN-MOExp-b** (MOEB) is similar to GNN-MOExp, while the strategy is to select explanations that are most balanced in both metrics.

B. Quantitative Results

Average simulatability and counterfactual relevance across all test nodes are reported in Table III. We conclude that:

- Gradient does not perform badly in counterfactual relevance (best in three datasets and second places in 2 datasets), but it performs worst or the second-worst in simulatability except

TABLE III: Overall performance (the higher (↑) the simulatability and the counterfactual relevance, the better). ○ indicates the runner-up methods and ● indicates the best method certified by statistically significant t-tests (pairwise t-test at 5% significance level). The worst performances are underlined and the second-worst performances are under wave lines.

Datasets	Simulatability (↑)									Counterfactual Relevance (↑)								
	RND	EMB	Grad	GAT	GNNExp	PGExp	Shapley	MOEB	GNN-MOExp	RND	EMB	Grad	GAT	GNNExp	PGExp	Shapley	MOEB	GNN-MOExp
Cora	-0.196	-0.252	-0.530	-0.243	-0.213	-0.272	-0.256	-0.108○	**-0.049●**	0.240	0.260	0.330	0.243	0.225	0.217	**0.615●**	0.455	0.467○
Citeseer	-0.051	-0.054	-0.066	-0.050	-0.056	-0.058	-0.068	-0.044○	**-0.039**	0.114	0.116	0.116	0.115	0.113	0.112	**0.178●**	0.156	0.159○
PubMed	-0.081	-0.110	-0.365	-0.117	-0.086	-0.125	-0.129	-0.041○	**-0.010●**	0.112	0.129	0.200	0.117	0.100	0.099	**0.330●**	0.235	0.248○
Musae-F	-0.972	-1.035	-0.899	-0.872	-0.911	-0.895	-0.346○	-1.313	**-0.199●**	0.613	0.653	0.438	0.546	0.576	0.520	0.696	**1.260●**	0.806○
Musae-G	-0.118	-0.120	-0.693	-0.110	-0.144	-0.220	-0.030○	-0.308	**-0.005●**	0.112	0.119	**0.527●**	0.118	0.126	0.126	0.247	0.366○	0.213
Amazon-C	-0.129	-0.126	-0.350	-0.134	-0.144	-0.175	-0.049○	-0.298	**-0.031●**	0.094	0.095	0.258○	0.089	0.087	0.061	0.201	**0.312●**	0.215
Amazon-P	-0.163	-0.180	-0.458	-0.175	-0.203	-0.231	-0.058○	-0.339	**-0.034●**	0.122	0.132	0.315○	0.123	0.111	0.090	0.257	**0.377●**	0.277
Coauthor-C	-0.216	-0.243	-0.745	-0.264	-0.245	-0.341	-0.097○	-0.411	**-0.038●**	0.183	0.205	**0.568●**	0.214	0.184	0.184	0.268	0.457○	0.263
Coauthor-P	-0.146	-0.144	-0.720	-0.220	-0.159	-0.295	-0.057○	-0.314	**-0.035●**	0.133	0.141	**0.534●**	0.149	0.138	0.167	0.208	0.367○	0.206
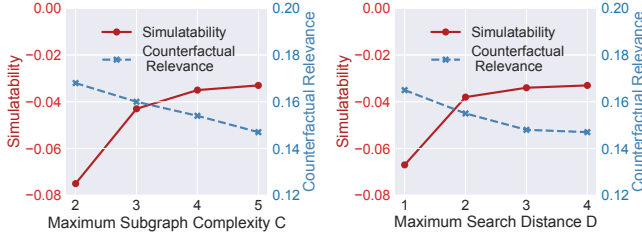
Fig. 8: Parameters sensitivity of maximum subgraph complexity C and maximum search distance D on Citeseer.

Fig. 9: Average Running time for each node of maximum search distance D and maximum subgraph complexity C on Cora and Citeseer.

the Musae-F dataset. That's because the gradients indicate the most effective perturbations of the edges to change a prediction. However, these edges do not constitute a graph to maximally preserve the GNN prediction. Based on the human study, Grad should be first excluded.

- GAT, GNNExplainer, and PGExp are outperformed by GNN-MOExp in both metrics on all datasets. Clearly, these baselines do not explicitly optimize both objectives.
- MOEB has the worst or second-worst simulatability on the latter 6 datasets, though it is the runner-up on the first three. Based on the human study, MOEB is not guaranteed to generate explanations that will likely be accepted.
- Shapley has the best counterfactual relevance on the first three datasets, with GNN-MOExp as the runner-up. On the remaining 6 datasets, GNN-MOExp outperforms or is close to Shapley. On simulatability, GNN-MOExp outperforms Shapley on all datasets.
- GNN-MOExp is the best in simulatability on all baselines on all datasets, and is frequently outperforming or competitive with the feasible runner-ups (Grad and MOEB are not feasible due to their low simulatability).

Parameters Sensitivity. We search subgraphs of C nodes involving vertices that are D hops away from the target node (by default $D = L$, the depth of the target GNN). The sensitivity analyses of these parameters are shown in Fig. 8. We can see that the performance of simulatability becomes better as the parameters increase, while the performance of counterfactual relevance becomes lower. We let $C \leq 5$ since large explaining subgraphs go against explanation simplicity and simulatability. Since L is usually small ($= 2$ in our experiments) to avoid over-smoothing [22], we can see the performance level off when $D \geq 2$.
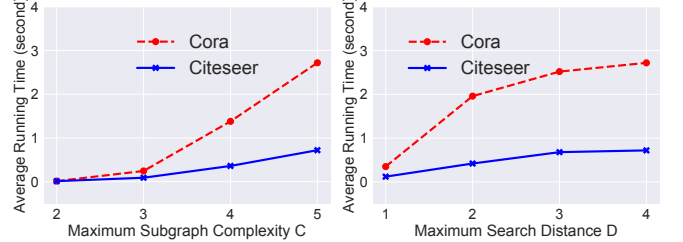
One bottleneck of applying GNN-MOExp to real-world graphs is its running time [37]. In Fig. 9 we can see that the running time increases as the search space grows with D and C. However, on average, enumerating and evaluating all acyclic and connected subgraphs of a target node on Cora and Citeseer with some very high node degrees, take no more than 3 seconds on a commodity computer. With an incremental implementation, a newly added edge only leads to enumerating new subgraphs containing the new edge. Given the reasonable running time, the capability of guaranteeing Pareto optimality and simultaneous high simulatability and counterfactual relevance is a unique advantage that gradient-based methods do not have. Explaining GNN with a quality guarantee is a must-have when GNN is used in user-centric applications, such as graph-based recommendation systems [51].

C. Robustness and Sanity check

We design two ways to perturb GNN predictions. We can link a vertex to the target node v_i and add a message $\mathbf{m}_{j'i}^{(L)}$ to Eq. (2) at the last layer of GNN:

$$\tilde{\mathbf{a}}_i^{(L)} = \text{AGG}\left(\left\{\mathbf{m}_{ji}^{(L)} | v_j \in \mathcal{N}(v_i)\right\} \cup \left\{\mathbf{m}_{j'i}^{(L)}\right\}\right), \quad (11)$$

where $\tilde{\mathbf{a}}_i^{(L)}$ is the perturbed activation. We measure the strength of the perturbation caused by $\mathbf{m}_{j'i}^{(L)}$ using

$$d_m\left(\mathbf{m}_{j'i}^{(L)}, \boldsymbol{\theta}_y^{(L)}\right) = -\cos\left(\mathbf{m}_{j'i}^{(L)}, \boldsymbol{\theta}_y^{(L)}\right), \quad (12)$$

where y is the predicted class of v_i before the perturbing edge is added. Second, we randomize the GNN parameters $\boldsymbol{\theta}^{(L)}$ of layer L, which is the last layer of GNN. We measure
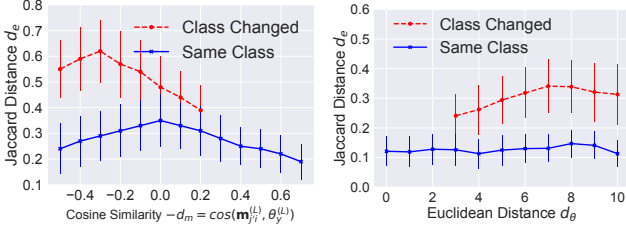
Fig. 10: Robustness and sanity check. *Left*: Jaccard distance d_e changes due to perturbed messages. *Right*:Jaccard distance d_e changes due to perturbed GNN parameters.

the perturbation strength using Euclidean distance between the original parameters $\boldsymbol{\theta}^{(L)}$ and the perturbed parameters $\tilde{\boldsymbol{\theta}}^{(L)}$

$$d_\theta(\boldsymbol{\theta}^{(L)}, \tilde{\boldsymbol{\theta}}^{(L)}) = \|\boldsymbol{\theta}^{(L)} - \tilde{\boldsymbol{\theta}}^{(L)}\|_2. \quad (13)$$

Given a perturbation, we need to measure the change in the explaining subgraph G_i^1 of $\mathbf{y}_i$. Let the explaining subgraphs after the perturbation be denoted by $\tilde{G}_i^1$. We measure the average distance between two explaining subgraph $d_e(G_i^1, \tilde{G}_i^1)$, where d_e is Jaccard distance between two vertex sets.

From Fig. 10, we can observe that the subgraph explanations found by our method pass the sanity check. In particular, the Jaccard distance between two sets of optimal explaining subgraphs becomes larger as the perturbing message $\mathbf{m}_{j'i}^{(L)}$ deviates from the vector $\boldsymbol{\theta}_y^{(L)}$ with a decreasing cosine similarity in the left of Fig. 10, and as the the perturbed parameters $\tilde{\boldsymbol{\theta}}^{(L)}$ deviates from the target model parameters $\boldsymbol{\theta}^{(L)}$ with an increasing euclidean distance in the right of Fig. 10. Also, there is no change in the predicted class by the target GNN when the perturbing message $\mathbf{m}_{j'i}^{(L)}$ is aligned with $\theta_y^{(L)}$ (high cosine similarity) or the perturbing distance is small, and predictions start to change when the perturbations are sufficiently strong. Importantly, d_e is larger when the original GNN predictions are changed under either perturbation. Interestingly, even when there is no change in the predicted class, d_e also increases as the cosine similarity d_m decreases, while d_e stays around 0.13 as the Euclidean distance d_θ increases. Robustness in the predicted class and the explanations is clearly demonstrated when the perturbations are sufficiently small, represented by a large cosine similarity $-d_m$ or small Euclidean distance d_θ. In such cases, on average less than two edges are perturbed in the explaining subgraphs.

D. Reproducibility checklist

We adopt a Graph Convolutional Network (GCN) model [20] to be explained, with two hidden layers ($L = 2$) each with dimension 16. The dimension of the input layer is equal to the feature dimension of the nodes, and the dimension of the output layer is set to the number of classification categories. We adopt the cross-entropy loss function and the Adam optimizer for training the GNN, while the learning rate is set to be 0.01. We set the maximal training iterations to 500, and apply the early-stop strategy when training.

As for the proposed GNN-MOExp, there are two hyper-parameters to be designed. We set the maximum search distance $D = 2$, which is equal to the depth of GCN, and we

set the maximum subgraph complexity $C = 4$, considering both the effectiveness and the calculation costs.

VI. RELATED WORK

Explainable ML. The simulatability and counterfactual relevance are two major metrics for evaluating explanations, but their interactions and how humans perceive them are not clear. In [29] and [41], they provide a prediction explanation framework based on Shapley values which encompasses LIME as a special case. Two algorithms with linear complexity for feature importance scoring are developed in [7]. In [12] and [2], they approximate Shapley values for deep networks via sampling. The methods proposed in [9], [6] use gradients to find salient subgraphs to explain the inference on PGM, but not for GNNs [20], [17], [45]. [49] explains arbitrary graph neural networks using a simplified model. [3] studies the influence of the change of inputs on outputs of GNN models with gradient-based and decomposition-based methods. *Stochastic* explaining subgraph search have been proposed [53], [46], [54] using reinforcement learning and hill-climbing. In [54], Monte Carlo search is used for exploration.

Causal Inference and Counterfactual Reasoning. [15] introduces both traditional and advanced methods in learning causal effect and causal relations. In [16], they discover the unknown confounders from observed data, by learning representations of confounders using GNN. We identify confounders on the computational graph of GNN.

Robustness and sensitivity. Explanation robustness and sensitivity are two desired properties and have been mostly studied on images [1], [11], [55], [48], [34] and texts [34], but none on graphs. The differential geometry formulation of manipulability of gradient-based explanations in [1] assumes that the input is a vector (image) that lies on a low-dimensional manifold. For GNN, a decision of a node depends not only on its feature vectors, but also on the messages from neighboring nodes. On graphs, the only relevant study is [47], and the proposed method differs from [47] in explanation generation (subgraph search vs. gradient-based) and evaluation metrics (output explanation changes vs. attribution accuracy changes).

VII. CONCLUSION AND FUTURE WORK

We proposed to find multi-objective explanations for Graph Neural Networks, with two objectives, simulatability and counterfactual relevance, to be satisfied. The human study showed that the two explanation objectives can represent the perceived quality of explanations based on two different cognitive processes (quick screening *vs*. effortful deliberation), and they jointly influence and predict explanation acceptance by humans. We proposed to maximize the two objectives by subgraph enumeration and ranking-based optimization to produce Pareto optimal explanations that fulfill both objectives. We showed that gradient-based GNN explanations are not robust against the rotation of incoming messages to the target nodes, while GNN-MOExp can reliably output quality explanations. Extensive experiments on 9 graph datasets from 4 applications demonstrated superior performance in simulatability, counterfactual relevance, robustness, and sensitivity.

ACKNOWLEDGEMENT

REFERENCES

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. In *NeurIPS*, 2018.

[2] Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley values approximation. In *ICML*, 2019.

[3] Federico Baldassarre and Hossein Azizpour. Explainability techniques for graph convolutional networks. In *ICML Workshop*, 2019.

[4] Solon Barocas, Andrew D Selbst, and Manish Raghavan. The Hidden Assumptions behind Counterfactual Explanations and Principal Reasons. In *FAT*, 2020.

[5] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. "It's Reducing a Human Being to a Percentage": Perceptions of Justice in Algorithmic Decisions. In *CHI*, 2018.

[6] Hei Chan and Adnan Darwiche. Sensitivity analysis in markov networks. In *IJCAI*, 2005.

[7] Jianbo Chen, Le Song, Martin J. Wainwrightand, and Michael I. Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. In *ICLR*, 2019.

[8] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. In *ICLR*, 2018.

[9] Adnan Darwiche. A differential approach to inference in bayesian networks. *Journal of the ACM*, pages 280–305, 2003.

[10] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Commun. ACM*, 63(1):68–77, 2019.

[11] Amirata Ghorbani, Abubakar Abid, and James Y Zou. Interpretation of Neural Networks is Fragile. In *AAAI*, 2017.

[12] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *ICML*, 2019.

[13] Bryce Goodman and Seth Flaxman. European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation". *AI Magazine*, 38:50–57, 2017.

[14] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51:93:1–93:42, 2018.

[15] Ruocheng Guo, Lu Cheng, Jundong Li, P Richard Hahn, and Huan Liu. A survey of learning causality with data: Problems and methods. *ACM Computing Surveys*, 2020.

[16] Ruocheng Guo, Jundong Li, and Huan Liu. Learning individual causal effects from networked observational data. In *WSDM*, 2020.

[17] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[18] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, Dawei Yin, and Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. *arXiv preprint arXiv:2001.06216*, 2020.

[19] Daniel Kahneman. *Thinking, fast and slow*. 2011.

[20] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[21] David K. Lewis. Causal explanation. In David Lewis, editor, *Philosophical Papers Vol. Ii*, pages 214–240. Oxford University Press, 1986.

[22] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 2018.

[23] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In *NeurIPS*, 2020.

[24] Christopher W Lynn and Danielle S Bassett. How humans learn and represent networks. *Proceedings of the National Academy of Sciences*, 117, 2020.

[25] Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *ICML*, 2020.

[26] K. Miettinen. *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science. 1998.

[27] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[28] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[29] Scott M.Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, pages 4765–4774, 2017.

[30] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In *FAT*, 2020.

[31] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009.

[32] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, 2014.

[33] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *CVPR*, 2019.

[34] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary Chase Lipton. Learning to Deceive with Attention-Based Explanations. *ArXiv*, abs/1909.0, 2019.

[35] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *SIGKDD*, 2015.

[36] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *arXiv preprint arXiv:1909.13021*, 2019.

[37] C. Rudin and Y. Shaposhnik. Globally-consistent rule-based summary-explanations for machine learning models: Application to credit-risk evaluation. *Human-Computer Interaction eJournal*, 2019.

[38] Chris Russell. Efficient Search for Diverse Coherent Explanations. In *FAT*, 2019.

[39] L Shapley. A value for n-person games. *Contributions to the Theory of Games*, pages 31–40, 1953.

[40] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[41] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017.

[42] Henri Jacques Suermondt. *Explanation in Bayesian Belief Networks*. PhD thesis, 1992.

[43] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *SIGKDD*, 2009.

[44] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable Recourse in Linear Classification. In *FAT*, 2019.

[45] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[46] Minh N Vu and M Thai. PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks. In *NeurIPS*, 2020.

[47] Alexander B Wiltschko, Benjamin Sanchez-Lengeling, Brian Lee, Emily Reif, Jennifer Wei, Kevin James McCloskey, Lucy Colwell, Wesley Qian, and Yiliu Wang. Evaluating Attribution for Graph Neural Networks. In *NeurIPS*, 2020.

[48] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David W Inouye, and Pradeep D Ravikumar. On the (In)fidelity and Sensitivity of Explanations. In *NeurIPS*, 2019.

[49] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnn explainer: A tool for post-hoc explanation of graph neural networks. In *NeurIPS*, 2019.

[50] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *SIGKDD*, 2018.

[51] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *SIGKDD*, 2018.

[52] Tomoki Yoshida, Ichiro Takeuchi, and Masayuki Karasuyama. Learning Interpretable Metric between Graphs: Convex Formulation and Computation with Graph Mining. In *SIGKDD*, 2019.

[53] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. Xgnn: Towards model-level explanations of graph neural networks. 2020.

[54] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. *ICML*, 2021.

[55] Xinyang Zhang, Ningfei Wang, Shouling Ji, Hua Shen, and Ting Wang. Interpretable Deep Learning under Fire. *ArXiv*, abs/1812.0, 2018.

[56] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 2017.