On Data-Aware Global Explainability of Graph Neural Networks

Ge Lv
HKUST
glvab@connect.ust.hk

Lei Chen
HKUST
leichen@ust.hk

ABSTRACT

Graph Neural Networks (GNNs) have significantly boosted the performance of many graph-based applications, yet they serve as black-box models. To understand how GNNs make decisions, explainability techniques have been extensively studied. While the majority of existing methods focus on local explainability, we propose DAG-Explainer in this work aiming for global explainability. Specifically, we observe three properties of superior explanations for a pretrained GNN: they should be highly recognized by the model, compliant with the data distribution and discriminative among all the classes. The first property entails an explanation to be faithful to the model, as the other two require the explanation to be convincing regarding the data distribution. Guided by these properties, we design metrics to quantify the quality of each single explanation and formulate the problem of finding data-aware global explanations for a pretrained GNN as an optimizing problem. We prove that the problem is NP-hard and adopt a randomized greedy algorithm to find a near optimal solution. Furthermore, we derive an improved bound of the approximation algorithm in our problem over the state-of-the-art (SOTA) best. Experimental results show that DAG-Explainer can efficiently produce meaningful and trustworthy explanations while preserving comparable quantitative evaluation results to the SOTA methods.

PVLDB Reference Format:
Ge Lv and Lei Chen. On Data-Aware Global Explainability of Graph Neural Networks. PVLDB, 14(1): XXX-XXX, 2023.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/Gori-LV/DAG.

1 INTRODUCTION

Graph Neural Networks (GNNs) have been widely employed in tasks using graph-structure data owing to their outstanding performance. However, the models are not yet fully trusted due to their black-box nature, as users cannot verify if the model is truly reliable. As a result, intensive research efforts have been devoted to understand how GNNs make decisions [15, 31, 43, 54, 65, 72]. Researchers attempt to identify substructures that are critical for GNNs to classify the instances. Such a subgraph, termed an *explanation* of the GNN, shows the focus of the complex model and sheds light on its decision making mechanism by answering the question "*what leads the model to make such a prediction*?". The majority of existing methods target local explainability, *i.e.*, *instance-level* explanation [29, 31, 52, 54, 56, 65, 69], which aims to find one explanation for a given input instance. Yet, the global explainability is under shallow exploration, *i.e.*, *model-level* techniques [32, 66], which produce input-independent explanations to capture the general behavior of the model. Though global methods could be less precise for one specific instance, they provide a higher-level interpretation of the model's decision making mechanism and thus avoid exploring the explanations for a large number of instances before we can trust the models. Hence, before the model takes any real-world service, global explanations can help domain experts examine the network's trustworthiness and capture any possible systematic errors . Not to mention that model-level explanations provide easy generalization to an inductive setting, which is the nature of many GNN applications [31]. In this work, we focus on model-level explanation to understand pretrained GNNs form a global view.

An open issue in explaining black-box models is the absence of a unified evaluation scheme for measuring explainability. Due to the scarcity of natural ground truths, common evaluation metrics such as accuracy, F1 score, and AUC score can no longer be applied to measure effectiveness in a scientific manner. Without golden knowledge, the key challenge lies in quantifying the quality of an explanation. To tackle this problem, we investigate and analyze the interrelation between GNNs and their training data. In some situations, outputs of existing methods cannot serve as superior model-level explanations. We elaborate on them in the following, using two datasets from different domains. The first one is the MUTAG dataset, comprised of molecular structures classified by their mutagenic effect. The second one is a social network dataset named Highschool, where each graph in it is a face-to-face contact network between highschool students, in which either a *high-risk* or *ordinary* epidemic is spreading. Selected examples from the two datasets[1] are shown in Figure 1(b) and (d), respectively.

Case 1. The generated explanation is not (highly) recognized by the GNN. As shown in **Case 1.** of Figure 1(a) is a 6-carbon ring found by SubgraphX [68] as an explanation for the *mutagenic* class in the MUTAG dataset; however its prediction score from the GNN for the underlying label is only 0.0028, which means the model does not recognize this structure as *mutagenic*. In **Case 1.** in Figure 1(c), a pattern of sequential social contacts between four students is found by Glocal [32] as an explanation for the *high-risk epidemic* class, yet its GNN score is only 0.5164. Though the predicted label is correct, the model is not exactly sure about its decision.

Case 2. The generated explanation does not exist in the data. A typical example is the explanation generated by XGNN [66], one of

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

[1] In the MUTAG dataset [24], nodes represent atoms and edges represent chemical bounds. In Highschool dataset, an edge represents a safe contact, or a risky contact without infection, or a risky contact with infection caused. More details see Section 3.1.1.

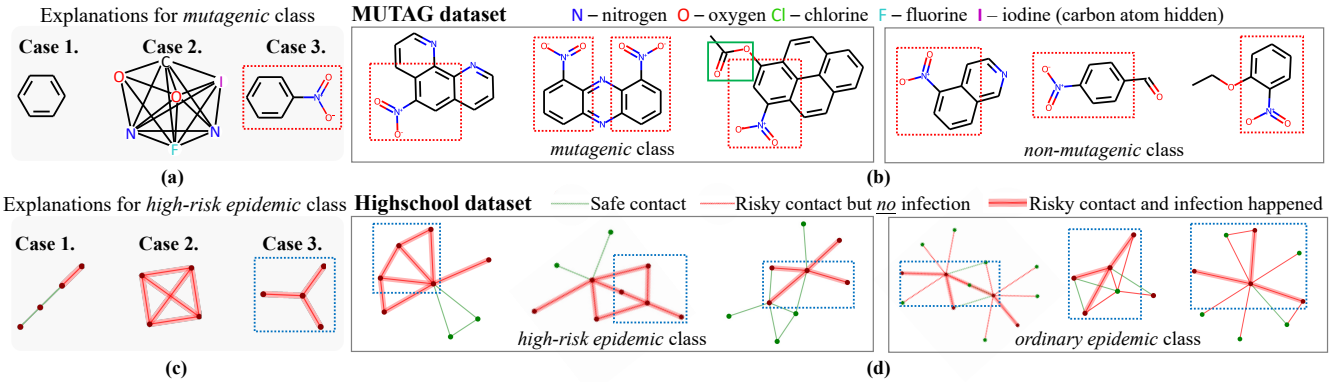

Figure 1: (a) & (c) The generated explanations by existing methods. (b) Examples form the MUTAG dataset, where nodes represent atoms and edges represent chemical bounds. (d) Examples from Highschool dataset, where nodes represent *susceptible* or *infected* students and an edge represents a safe contact, or a risky contact without infection, or a risky contact with infection caused. More details on the datasets are presented in Section 3.1.1.

its output is shown in **Case 2.** in Figure 1(a). Besides, the structure violates the chemical rules: take the oxygen atoms as examples, the maximum absolute value of their valency is two [21], however their degrees in the structure are four and six respectively; the same problem also happens with other atoms. Thus rationality of the explanation structure is doubtful. In **Case 2.** in Figure 1(c) shows a complete network of four infected students, all of their contacts are risky ones with infection caused. The structure is believed to be a traditional high-risk epidemic network [58]. However, this structure does not exist in the data, thus it cannot serve as a superior model-level explanation.

Case 3. The generated explanation does not discriminate among classes. For instance, shown in the red box in Figure 1(a) is a carbon ring attached with an NO_2 structure (**Case 3.**) found by GNNExplainer [65] as an explanation for *mutagenic* class, it receives a GNN score as high as 0.99998, and exists in 91.2% of graphs in the class. The issue is that this structure also presents in 84.1% of instances in the *non-mutagenic* class. Similarly, as shown in the blue box in Figure 1(b) is a windmill-shaped structure found for explaining the *high-risk epidemic* class [32]. It is not only present in 66 graphs in the target class, but also in 37 graphs of the *ordinary epidemic* class. The explainers are then trapped in the predicament where graphs containing an explanation of some class are actually predicted as another class by the GNN.

Essentially, the first case demonstrates that existing explainers may fall short from the model's perspective: the outputs may not be faithful to the GNN; the other two show that current techniques may fail from the data perspective: the outputs may not align with the knowledge of the data. An example of a high-quality explanation is shown in the green box in Figure 1(b): two oxygen atoms connected by one carbon atom. The structure has a GNN score of 1.0, and presents in the *mutagenic* class only, which suggests it is both faithful to the model and truly data-aware. If an explanation does not hold faithfulness to the GNN, it becomes invalid and meaningless; if it does not retain data-awareness, it is not trustworthy or convincing. Thus, our goal is to find superior explanations that can achieve the best of both worlds.

In conclusion, we aim for a set of substructures that overcomes the shortcomings of three discussed cases while preserving effectiveness of the existing methods. This is a challenging task due to the following three reasons. First, without ground truth of the problem, one needs to design persuasive and vital metrics for defining the optimization goal as well as measuring the quality of the final output. Second, for each single explanation, the recipe for solving different cases may not align with each other, thus finding optimal subgraphs that can tackle all the discussed problems concurrently is difficult. Third, the quantifying metrics defined to be optimized do not necessarily preserve properties required for commonly used optimization techniques (*e.g.* monotonicity and submodularity), hence designing an algorithm with a theoretical guarantee is non-trivial.

To address these issues, we propose a framework named DAG-Explainer (Data-Aware Global Explainer), in which we first define a number of new metrics to quantify how superior an explanation is from the perspectives of both the model and the data, and further introduce an objective function that scores the quality of different sets of model-level explanations; then we propose a randomized greedy algorithm with theoretical bound to find a final set of explanations that optimizes the objective function. Our contribution is summarized as follows:

- We propose a unified evaluation scheme to quantify the explainability of a structure, which can be used to measure the quality of explanations in a model-agnostic fashion.
- We formulate the data-aware global explanations generation problem as finding an optimal set that maximizes an objective function. We then show that solving it for the optima is NP-hard by proving the objective is weakly-submodular.
- We propose a framework named DAG-Explainer that adopts a randomized greedy algorithm to find a near-optimal solution to the problem, and derive an improved bound of the approximation algorithm in our problem over the state-of-the-art best.
- We conduct experiments on one synthetic dataset and two real-world datasets to demonstrate that our method outputs meaningful and trustworthy explanations with decent quantitative evaluation results for GNNs.

2 DATA-AWARE GLOBAL EXPLAINABILITY OF GRAPH NEURAL NETWORKS

In this section, we first present related concepts and preliminaries of our problem, then formally introduce our proposed DAG-Explainer.

2.1 Preliminaries

In general, GNNs learn node representations by iteratively aggregating neural messages along edges between neighboring nodes. Consider a graph $G = (V, E)$ consisting of a set of nodes $V = \{v_1, v_2, ..., v_N\}$ and a set of edges $E \subseteq V \times V$, the nodes are associated with d-dimensional node features $\mathcal{X} = \{x_1, x_2, ..., x_N\}, x_i \in \mathbb{R}^d$. Let $\hat{\mathbf{A}} = \bar{\mathbf{D}}^{\frac{1}{2}} \bar{\mathbf{A}} \bar{\mathbf{D}}^{-\frac{1}{2}}$ be the symmetrically normalized adjacency matrix in GCN [26], where $\bar{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is the adjacency matrix of G after adding self-loops and $\bar{\mathbf{D}} = \mathbf{D} + \mathbf{I}_N$ is the diagonal node degree matrix. Without loss of generality, the l-th layer message propagation of GNNs [2, 60] can be formulated in matrix form as below:

$$\mathbf{H}^{(l)} = \sigma\left(\mathbf{H}^{(l-1)}, \hat{\mathbf{A}}; \mathbf{W}^{(l-1)}\right),$$

where σ is the non-linear activation function and $\mathbf{W}^{(\cdot)}$ is a trainable weight matrix. Initially, the node feature matrix $\mathcal{X}$ is used as $\mathbf{H}^{(0)}$. By stacking k layers, the k-hop neighborhood information can be aggregated. For graph classification, a global pooling layer (e.g., max pooling, mean pooling) is needed to combine node representations to give a single representation for the graph; mathematically, it is calculated as $\mathbf{z}_G = \text{POOL}\left(\mathbf{H}^{(k)}\right)$ for a k-layer GNN. Finally, the node or graph representation is input to a classifier (e.g., fully connected layers) to give a prediction on the instance. In summary, GNNs take the adjacency matrix and the node feature matrix as input, then output a predicted class for each instance.

In this work, we consider a GNN $\phi(\cdot)$ pretrained on a dataset, the goal is to find a set of substructures that best explain the model. We denote a candidate explanation as $e = (s, c)$, where s is a subgraph mined from the dataset and c is the predicted class when input s into the GNN. Let $\mathcal{I} = \{g_1, ..., g_N\}$ be the set of all instances in the dataset, and note that instances can be either graphs in graph classification or computational graphs of nodes in node classification. We use $s \sqsubseteq g$ to denote that s is a subgraph of g, the set of candidates of data-aware global explanations of $\phi(\cdot)$ for class c^* is defined as:

$$C(c^*) = \bigcup_{\forall g \in \mathcal{I}} \{e = (s, c) \mid s \sqsubseteq g \wedge c = c^*\} \tag{1}$$

For the convenience of notation, we simply write C in the rest of the paper. We seek for a group of high-quality explanations, denoted by $\mathcal{E}$, that is the optimal set of candidates which can best explain the pretrained GNN, and the desired output should not fail in any case discussed in Section 1. For the sake of guiding the optimization, we first conclude criteria from the perspectives of both the model and the data that global explanations should preserve based on our observation, then further design quantitative metrics to measure to what extend a candidate satisfies the criteria.

2.2 Properties of High-quality Explanations

While analyzing the interrelation between GNNs and their training data, we identify three properties of superior model-level explanations, which form the basis of our method.

Property 1. *A high-quality explanation should be highly recognized by the GNN.* Desired explanations are expected to receive high GNN scores for their respective class so as to ensure they faithfully explain the GNN. High prediction scores are crucial for global explainability since only when the model is decidedly confident about its decision, one can conclude that the input structure is truly critical for the GNN to make decisions in general. As a matter of fact, this property aligns with the input optimization technique for model-level explainability in dealing with image and text data [37, 38, 40], which has recently been introduced to graph-based data [66]. The technique aims at optimizing input that can maximize a certain prediction score while keeping the model fixed. Likewise, we directly employ the prediction score for the respective class when inputting the candidate structure into the GNN as a metric, considering that the score provides a direct and faithful indication of the model's behavior. Formally, it is defined as below.

Definition 2.1. Faithfulness of a candidate $e = (s, c)$ is the prediction score that the GNN gives for class c when input s:

$$faithfulness(e) = \phi(s)[\hat{y} = c]$$

where $\phi(\cdot)$ is the GNN and $[\hat{y} = c]$ denotes that the score is specified for the underlying class c.

Property 2. *A high-quality explanation should be compliant with the data distribution.* Otherwise, it may contradict the knowledge that the GNN learned form the data, or even violate domain rules and consequently decrease users' trust in the model. If the explanation structure is doubtful regarding the data distribution, it will result in human users rejecting the explained GNN even if its performance is decent. For this property, candidate structures are prepared by mining subgraphs from the dataset, which already ensure the candidate explanations are compliant with the data distribution. Furthermore, we consider that superior explanations are also closely related to salient structures in input graphs, *i.e.* the reoccurring subgraphs [43]. Intuitively, frequent patterns with higher support in the class possess stronger evidence to prove themselves representative of the data and qualified as high-quality explanations. Hence, we first introduce *support* of an explanation as below:

Definition 2.2. Support of an explanation $e = (s, c)$ is defined as the set of instances g in the underlying class and s is a subgraph of g, *i.e.*,

$$support(e) = \{g \mid s \sqsubseteq g, \forall g \in \mathcal{I}_c\}$$

where $\mathcal{I}_c = \{g \mid \phi(g) = c, \forall g \in \mathcal{I}\}$.

Naturally, the size of the support set can be directly employed as a metric for measuring *Property 2.*

Property 3. *A high-quality explanation should be discriminative among classes.* In contrast to local explainability, global explanations bear an additional liability to show how the model perceives the difference between two classes. Failing to preserve the property will lead to a predicament that instances containing a global explanation for some class are predicted as another class by the GNN, where the explanation turns out to be invalid due to the lack of discrimination among classes. The property is also termed *contrastive* [17, 18], which aims for similarities to graphs within the same class and differences with graphs in the other class(es). Since we already consider the compliance level of an explanation in the *support* metric, we utilize the amount of its wrong-class presences to evaluate the discrimination level. We define the *denial* of an explanation as below:

Definition 2.3. Denial of an explanation $e = (s, c)$ is defined as the set of instances g, of which s is a subgraph and g does not belong to class c:

$$denial(e) = \{g \mid s \sqsubseteq g, \forall g \in \mathcal{I} / \mathcal{I}_c\}$$

where $\mathcal{I}_c = \{g' \mid \phi(g') = c, \forall g' \in \mathcal{I}\}$, and $\mathcal{I} / \mathcal{I}_c$ denotes the relative complement of $\mathcal{I}_c$ with respect to $\mathcal{I}$.

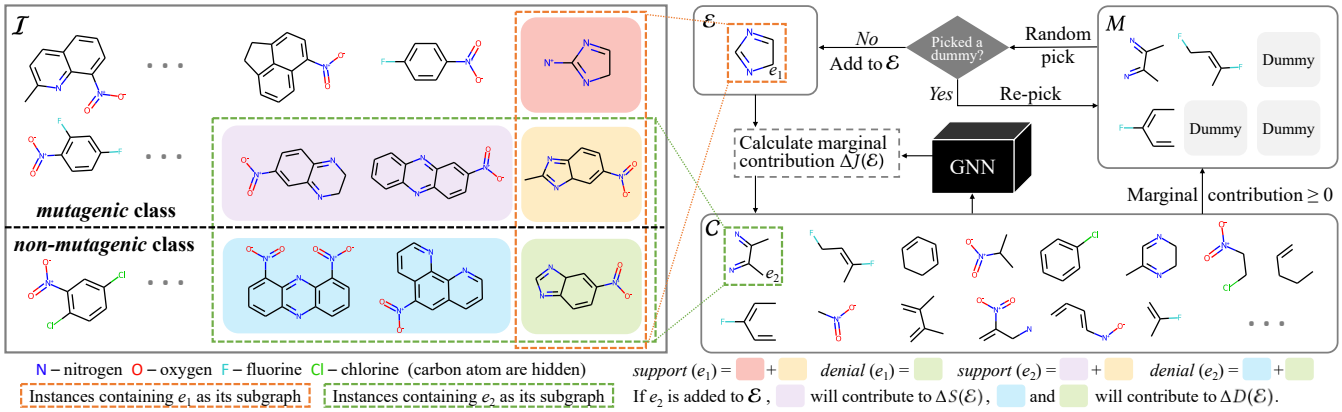

Figure 2: Illustrative examples of $support$ and $denial$ of explanations, which are highlighted in colored background, and workflow of DAG-Explainer.

As a result, the size of the denial set is then considered a metric for measuring *Property 3*. To illustrate the concepts of *support* and *denial*, we show two example explanations e_1 and e_1 in Figure 2 in the MUTAG dataset. In the instance set $\mathcal{I}$ divided by class labels (*mutagenic* class and *non-mutagenic* class), $support(e_1)$ is highlighted in **pink** and **yellow**, $denial(e_1)$ is highlighted in **green**. Meanwhile, $support(e_2)$ is highlighted in **purple** and **yellow**, with $denial(e_2)$ in **blue** and **green**. In graph mining research [12, 23, 62, 63], the union $support(e) \bigcup denial(e)$ of an explanation $e = (s, c)$ is the set of graphs that contains s as its subgraph, and $|support(e) \bigcup denial(e)|$ equals the frequency of s in the dataset $\mathcal{I}$. We equip this idea with class labels in the GNN explanation task to measure the data-awareness of a candidate.

2.3 Objective Function

With the help of all the observed properties and metrics described above, we are now ready to introduce the optimization problem and the underlying objective function for the data-aware global explanation task. The function consists of components that correspond to characteristics desired for the final output. To assist the optimization procedure, we always define the components of the objective to be non-negative. In addition, each of them will be normalized via dividing by the corresponding upper bound, so that their values will fall in the interval between 0 and 1 to match the range of GNN score. The joint objective comprises the following measurements on a set of candidates.

Overall fidelity. Primarily, the level of recognition from the pretrained GNN regarding an explanation set is computed as the average faithfulness of each single candidate in the set:

$$F(\mathcal{E}) = \frac{\sum_{e \in \mathcal{E}} faithfulness(e)}{|\mathcal{E}|}$$

where $|\mathcal{E}|$ is the cardinality of the current explanation set.

Total support. Secondly, we favor a set with higher overall support. Although Property 2 only requires the explanation to conform to the data distribution, it is still expected that more support leads to more reliability; besides, this objective also encourages the diversity of the explanation set. Formally, total support is calculated as:

$$S(\mathcal{E}) = \frac{|\bigcup_{e \in \mathcal{E}} support(e)|}{|\mathcal{I}|}$$

Average denial. Different from *support*, we care more about lower denial, thus we design the corresponding object to minimize the average denial ratio out of all the explanations of the candidate set. The subtrahend is normalized by the product of $|\mathcal{I}|$ and $|C|$, while the former is the upper bound for denial of one candidate, and the latter is the upper bound of possible candidates to be selected. The average denial is computed as:

$$D(\mathcal{E}) = 1 - \frac{\sum_{e \in \mathcal{E}} |denial(e)|}{|\mathcal{I}| \cdot |C|}$$

Size. Finally, we seek explanation sets of a smaller size for the ease of human understanding, thus we encourage a small-size explanation set using the measurement below:

$$Z(\mathcal{E}) = 1 - \frac{|\mathcal{E}|}{|C|}$$

We are now ready to propose our integrated learning objective, mathematically, the function is written as

$$J(\mathcal{E}) = F(\mathcal{E}) + \lambda_1 \cdot S(\mathcal{E}) + \lambda_2 \cdot D(\mathcal{E}) + \lambda_3 \cdot Z(\mathcal{E}), \qquad (2)$$

where λ_i for $i = 1, 2, 3$ are non-negative, which represent the relative importance of the components. They can be set depending on the underlying GNN application, or according to some cross-validation requirements as used in the setting of our experiments. The problem studied in this work is formally defined as below:

Definition 2.4. Given a GNN trained on a group of instances $\mathcal{I}$ and a set of candidates C generated according to Equation (1), the *data-aware global explanation generation problem* for the pretrained GNN is to find a set of substructures that maximizes the objective function (2).

2.4 Optimizing the Objective

As the objective considers different metrics that may not align with each other, optimizing it is non-trivial. Nonetheless, it retains a distinctive structure that can be utilized for approximating the optima with a theoretical guarantee. In the objective function (2), all components are constructed in a way to ensure they are non-negative. $D(\mathcal{E})$ and $Z(\mathcal{E})$ are modular, subsequently, they are submodular. The other component $S(\mathcal{E})$ is also submodular. While the crux lies in $F(\mathcal{E})$, because GNN $\phi(\cdot)$ is a complex black-box function, which is neither submodular nor monotone. Yet we can prove that $F(\mathcal{E})$ is a non-negative non-monotone weakly submodular function, and so is the complete objective function. Weak submodularity was originally introduced on monotone functions by Abhimanyu and Kempe [14], which is further generalized to non-monotone functions by

Santiago and Yoshida [49]. Before presenting more theoretical analysis, we introduce the notation to be used throughout this paper: for a set function $f : 2^E \to \mathbb{R}_+$, the marginal gain of adding the set B to A is denoted as $f_A(B)$, *i.e.* $f(A \cup B) - f(A)$, and $f_A(e)$ is used instead of $f_A(\{e\})$ for simplicity. Formally, the (non-monotone) weakly submodular function is defined as below:

Definition 2.5 (Definition 1.1 [49]). Given a scalar $\gamma \in (0, 1]$, a set (non-monotone) function $f : 2^E \to \mathbb{R}_+$ is ***γ-weakly submodular*** if $\sum_{e \in B} f_A(e) \geq \min\{\gamma f_A(B), \frac{1}{\gamma} f_A(B)\}$ for any pair of disjoint sets $A, B \subseteq E$.

According to this definition, we propose the below lemma:

Lemma 2.6. *$J(\mathcal{E})$ is a non-negative, non-monotone and γ-weakly submodular function with $\gamma = \frac{1}{|C|}$ on the candidate set C.*

As discussed, all components in Equation (2) are non-negative. In addition, $S(\mathcal{E}), D(\mathcal{E}), Z(\mathcal{E})$ are submodular. Note that non-negativity and (weak) submodularity are closed under the operation of addition and multiplication with non-negative constants. Hence to prove Lemma 2.6, it is sufficient to prove $F(\mathcal{E})$ is weakly submodular. We now give the proof below.

Proof. Let $P = \{\rho_i\}$ and $T = \{\tau_j\}$ be two arbitrary non-empty *disjoint* subsets in the domain of $F(\mathcal{E})$, *i.e.*, the subsets of the candidate space C. Thus we have $\rho_i, \tau_j \in (0, 1]$, since C consists of candidates with target predicted class, whose confidence is the highest among all labels. Function $F(\mathcal{E})$ computes the average value of all elements in the input set. Let $p = |P|$ and $t = |T|$, thus $p, t \geq 1$. We further denote

$$\overline{\rho} = F(P) = \frac{\sum_{\rho_i \in P} \rho_i}{p} \quad \text{and} \quad \overline{\tau} = F(T) = \frac{\sum_{\tau_j \in T} \tau_j}{t}.$$

Then $\overline{\rho}, \overline{\tau} \in (0, 1]$. We aim to find $\gamma \in (0, 1]$ *s.t.*

$$\sum_{\tau \in T} F_P(\tau) \geq \min\{\gamma F_P(T), \frac{1}{\gamma} F_P(T)\} \tag{3}$$

for any pair of disjoint sets P and T in the domain.
In *L.H.S.* of Equation (3),

$$\sum_{\tau \in T} F_P(\tau) = \sum_{\tau \in T} \Big(F(P \cup \{\tau\}) - F(P)\Big) = \sum_{\tau \in T} \Big(\frac{p\overline{\rho} + \tau}{p+1} - \overline{\rho}\Big) = \frac{t(\overline{\tau} - \overline{\rho})}{p+1} \tag{4}$$

In *R.H.S.* of Equation (3),

$$F_P(T) = F(P \cup T) - F(P) = \frac{p\overline{\rho} + t\overline{\tau}}{p+t} - \overline{\rho} = \frac{t(\overline{\tau} - \overline{\rho})}{p+t} \tag{5}$$

Consider the two cases below:

Case i. $\overline{\tau} \geq \overline{\rho}$, since $t \geq 1$, *R.H.S.* of Equation (4) $\geq$ *R.H.S.* of Equation (5). Thus Equation (3) holds with $\gamma = 1$.

Case ii. $\overline{\tau} < \overline{\rho}$, *R.H.S.* of Equation (5)< 0, then

$$\min\{\gamma F_P(T), \frac{1}{\gamma} F_P(T)\} = \frac{1}{\gamma} F_P(T).$$

Assume there exists some $\gamma \in (0, 1]$ such that the *R.H.S.* of Equation (4) $\geq \frac{1}{\gamma}$ *R.H.S.* of Equation (5), that is

$$\frac{t(\overline{\tau} - \overline{\rho})}{p+1} \geq \frac{t(\overline{\tau} - \overline{\rho})}{\gamma(p+t)} \quad \Leftrightarrow \quad p+1 \geq \gamma(p+t) \quad \Leftrightarrow \quad (1-\gamma)p + 1 - \gamma t \geq 0 \tag{6}$$

Since $1 - \gamma \geq 0$, we only need $1 - \gamma t \geq 0$; t is the size of a non-empty subset of C, hence $t \leq |C|$. Thus let $\gamma = \frac{1}{|C|}$, we have Equation (6) satisfied in *Case ii.* Combining both cases, we finish the proof. □

Algorithm 1: DAG-Explainer

Input: the pretrained GNN model, candidate set C defined by Equation (1), cardinality constraint k;
Output: generated explanation set $\mathcal{E}$;
Initialization: $\mathcal{E} \leftarrow \emptyset$, $i = 0$;
while $i < k$ **do**
  $M \leftarrow \emptyset$;
  for $e \in C$ **do**
    g = marginal gain of e given $\mathcal{E}$;
    if $g \geq 0$ **then**
      if $|M| = k$ **then**
        if $g >$ *lowest marginal gain for all* $e \in M$ **then**
          Replace the element with minimal marginal gain in M with e;
      else
        $M = M \cup \{e\}$;
  if $|M| = 0$ **then**
    return $\mathcal{E}$;// no more positive marginal gain
  else
    Add $(k - |M|)$ dummy variables to M;
    Uniformly random pick an element $e' \in M$;
    while *a dummy variable is picked and* $i < k$ **do**
      $i = i + 1$; // start a new iteration and pick again
      Pick a new e' uniformly random;
    if $i = k$ **then**
      return $\mathcal{E}$;// used up all iterations
    else
      $\mathcal{E} = \mathcal{E} \cup \{e'\}$;
      $i = i + 1$;
return $\mathcal{E}$;

In the sense of relaxing the diminishing marginal gain property, weak submodularity is equivalent to submodularity [49]. Maximizing a γ-weakly submodular function is NP-hard since maximizing a submodular function, which is a special case of weak submodularity ($\gamma = 1$) is NP-hard [28]. Specifically, the problem defined in Definition 2.4 is an NP-hard problem with a hardness factor γ, where $\gamma = \frac{1}{|C|}$ and C is the candidate space defined in Equation (1). As the problem is non-traceable in polynomial time, and commonly used techniques such as Smooth Local Search [19] no longer provide a theoretical guarantee, we then introduce the `RandomGreedy` algorithm proposed by Buchbinder, *et al.* [8] to optimize the objective function; pseudo code is available in Algorithm 1 and the workflow is shown in Figure 2. In each iteration, the algorithm first constructs a candidate pool M of size k, where every candidate retains a positive marginal gain given the current explanation set. In the case where the number of candidates with positive marginal gain is less than k, dummy variables with a virtual marginal gain as zero are supplied to make sure the pool M is of size k. The algorithm then iteratively picks candidates with non-negative marginal gain with a probability $\frac{1}{k}$. As a result, we ensure the objective function increases in each iteration. In addition, the algorithm is quite efficient as it only queries $O(k|C|)$ times for marginal gain, which is of the same cost as the standard deterministic greedy algorithm.

The `RandomGreedy` algorithm with cardinality constraints has been employed to handle various objective functions and the corresponding approximation factor is well studied. When optimizing submodular functions, if the objective function is additionally monotone, the `RandomGreedy` algorithm retains an approximation ratio of $1 - e^{-1}$, while preserving an approximation of e^{-1} for non-monotone submodular objectives [8]. For a non-negative monotone γ-weakly submodular function, the algorithm guarantees an approximation ratio of at least $(1 + 1/\gamma)^{-2}$ [11]. In our problem, the

objective function is non-negative, non-monotone and γ-weakly submodular, the state-of-the-art approximation ratio proved for the RandomGreedy algorithm is $\gamma \cdot e^{-1/\gamma}$ [49]; that is $\frac{1}{|C|} \cdot e^{-|C|}$ for the data-aware global explanation generation problem studied in this work. The guarantee is rather pessimistic, because the weak submodularity ratio γ is defined universally so that it must hold for all possible cases with any disjoint sets in the domain of an arbitrary objective, and the candidate space C in our problem is generally large. Yet we believe there exists an improved local bound for DAG-Explainer, because our objective function has a particular structure with one component as a weakly submodular function and all the others are submordular. To find a better theoretical guarantee, we present the below theorem proved by Santiago and Yoshida (2020), on which our proposed approximation factor is based.

Theorem 2.7 (Theorem 2.4. [49]). *Let $f : 2^E \to \mathbb{R}_+$ be a set function. Assume there are values $0 \le \bar{\alpha}_i \le \bar{\beta}_i \le k$ and $0 \le \alpha_i \le \beta_i \le k$ such that*

$$\sum_{u \in M_i} f_{S_i-1 \cup \mathtt{OPT}}(u) \ge min\{\bar{\alpha}_i \cdot f_{S_i-1 \cup \mathtt{OPT}}(M_i),\ \bar{\beta}_i \cdot f_{S_i-1 \cup \mathtt{OPT}}(M_i)\} \tag{7}$$

and

$$\sum_{e \in \mathtt{OPT}} f_{S_i-1}(e) \ge min\{\alpha_{i-1} \cdot f_{S_i-1}(\mathtt{OPT}),\ \beta_{i-1} \cdot f_{S_i-1}(\mathtt{OPT})\} \tag{8}$$

is satisfied for any choice of M_i and S_{i-1} throughout the execution of the RandomGreedy *algorithm. Then at any iteration $1 \ge i \ge k$ the algorithm satisfies*

$$\mathbb{E}[f(S_i)] \ge \left(\prod_{j=1}^{i-1} \min\{1 - \frac{\bar{\beta}_j}{k}, 1 - \frac{\alpha_j}{k}\}\right) \cdot \left(\sum_{j=0}^{i-1} \frac{\alpha_j}{k}\right) \cdot f(\mathtt{OPT}).$$

In the respective analysis, the substitution of $\alpha_i, \bar{\alpha}_i$ with γ and $\beta_i, \bar{\beta}_i$ by $\frac{1}{\gamma}$ immediately leads to an approximation of $\gamma(1 - \frac{1}{\gamma k})^{k-1}$ for a non-negative non-monotone γ-weakly submodular function, which is asymptotically $\gamma e^{-1/\gamma}$ as $k \to \infty$. A better bound can be derived using the above theorem for the objective function $J(\mathcal{E})$. Before proposing the guarantee, we introduce the following lemma:

Lemma 2.8. *A submodular function $f : 2^E \to \mathbb{R}_+$ always satisfies*

$$\sum_{e \in B} f_A(e) \ge min\{f_A(B),\ \beta^* f_A(B)\} \tag{9}$$

for an arbitrary real number $\beta^* \ge 1$ and any pair of sets $A, B \subseteq E$.

The proof of this lemma is presented in the Supplementary Material [1]. It is worth noting that by taking $\beta^* = 1$, Lemma 2.8 verifies that submodularity is a special case of γ-weak submodularity with $\gamma = 1$. Now we are ready to propose a useful theorem that provides a theoretical guarantee for execution of the RandomGreedy algorithm to maximize functions with a particular structure.

Theorem 2.9. *Let $\mathcal{U} = \{c \mid c \in \mathbb{R}_+\}$ be a finite set of positive real numbers, a set function $g : \wp^+(\mathcal{U}) \to \mathbb{R}_+$ is defined as*

$$g(\mathcal{S}) = \frac{\sum_{c \in \mathcal{S}} c}{|\mathcal{S}|} + h(\mathcal{S}),$$

where $h(\mathcal{S})$ is an arbitrary non-negative submodular function and $\wp^+(\mathcal{U})$ denotes the set of non-empty subsets of $\mathcal{U}$. Execution of the RandomGreedy *algorithm for maximizing g produces (on expectation) an approximation factor of at least $(1 - \frac{|\mathcal{U}|+1}{2k})^{k-1}$ with a cardinality constraint $k \ge \left\lceil \frac{|\mathcal{U}|+1}{2} \right\rceil$, which is asymptotically $e^{-\frac{|\mathcal{U}|}{2}}$ as $k \to \infty$.*

Proof. To prove the theorem, we denote $\hat{g}(\mathcal{S}) = \frac{\sum_{c \in \mathcal{S}} c}{|\mathcal{S}|}$ and first prove that for any pair of non-empty sets $A, B \in \wp^+(\mathcal{U})$,

$$\sum_{e \in B} \hat{g}_A(e) \ge min\{\hat{g}_A(B),\ \beta^* \hat{g}_A(B)\} \tag{10}$$

is always satisfied with $\beta^* = \frac{|\mathcal{U}|+1}{2}$.

For simplicity, we continue to adopt the notation used in the proof of Lemma 2.6, where two cases are discussed separately. Let $P, T \subseteq \mathcal{U}$ be two arbitrary non-empty *disjoint* sets in the domain of $\hat{g}(\mathcal{S})$, denote $p = |P|$, $t = |T|$, $\overline{\rho} = \hat{g}(P)$, $\overline{\tau} = \hat{g}(T)$ and the we discuss the two cases respectively.

Case i. $\hat{g}_P(T) \ge 0$, Equation (10) $\Leftrightarrow \sum_{e \in T} \hat{g}_P(e) \ge \hat{g}_P(T)$ since $\beta^* \ge 1$ given $\mathcal{U}$ is non-empty. Without the constraint on the range of elements in $\mathcal{U}$, it can still be easily verified that, the condition always holds (refer to *Case i.* in the proof of Lemma 2.6).

Case ii. $\hat{g}_P(T) < 0$, that is $\overline{\tau} < \overline{\rho}$, one needs to verify that $\sum_{e \in T} \hat{g}_P(e) \ge \beta^* \hat{g}_P(T)$, which is

$$\frac{t(\overline{\tau} - \overline{\rho})}{p+1} \ge \frac{\beta^* t(\overline{\tau} - \overline{\rho})}{(p+t)} \quad \Leftrightarrow \quad \beta^* \ge 1 + \frac{t-1}{p+1},$$

(see *Case ii.* in the proof of Lemma 2.6). Since $t, p \in [1, |\mathcal{U}|]$, we see that

$$1 + \frac{t-1}{p+1} \le 1 + \frac{|\mathcal{U}|-1}{2}.$$

Let $\beta^* = \frac{|\mathcal{U}|+1}{2}$, Equation (10) is satisfied in this case.

Without loss of generality, one also needs to consider the situation where P and T are not disjoint. When $T \subseteq P$, Equation (10) is trivial; when $T \backslash P \ne \varnothing$, it can be easily proved by substituting T with $T' = T \backslash P$ in the above discussion. In conclusion, Equation (10) always holds for any pair of non-empty sets $A, B \in \wp^+(\mathcal{U})$.

Combining Equation (10) and Lemma 2.6 for the submodular component $h(\mathcal{S})$, we conclude that

$$\sum_{e \in B} g_A(e) \ge min\{g_A(B),\ \beta^* g_A(B)\},$$

for $\beta^* = \frac{|\mathcal{U}|+1}{2}$. Consider g in the setting of Theorem 2.7, let $\alpha_i = \bar{\alpha}_i = 1$, $\beta_i = \bar{\beta}_i = \beta^*$ and $k \ge \left\lceil \frac{|\mathcal{U}|+1}{2} \right\rceil$, the theorem guarantees that at any iteration $1 \ge i \ge k$ during the execution of the RandomGreedy algorithm, the inequality below is always satisfied:

$$\mathbb{E}[g(S_i)] \ge \left(\prod_{j=1}^{i-1} \left(1 - \frac{\beta^*}{k}\right)\right) \cdot \left(\sum_{j=0}^{i-1} \frac{1}{k}\right) \cdot f(\mathtt{OPT}).$$

Upon finishing the execution, *i.e.*, $i = k$, we have

$$\mathbb{E}[f(S_i)] \ge \left(1 - \frac{\beta^*}{k}\right)^{k-1} \cdot f(\mathtt{OPT}),$$

where $\beta^* = \frac{|\mathcal{U}|+1}{2}$. The proposed theorem is then proved. □

The conclusion directly leads to the below corollary regarding our problem using a direct mapping from $\mathcal{U}$ to C and taking $J(\mathcal{E})$ as the objective function.

Corollary 2.10. *The execution of the* RandomGreedy *algorithm to solve the data-aware global explanation generation problem defined in Definition 2.4 with a cardinality constraint $k \ge \left\lceil \frac{|C|+1}{2} \right\rceil$ produces (on expectation) an approximation factor of at least $(1 - \frac{|C|+1}{2k})^{k-1}$. When $k \to \infty$, the solution has a theoretical guarantee of $O(e^{-\frac{|C|}{2}})$.*

Hereby, we have shown that DAG-Explainer has an improved bound $O(e^{-\frac{|C|}{2}})$ over the state-of-the-art bound $O(\frac{1}{|C|} \cdot e^{-|C|})$ in our problem. In next section, we present our experimental study.

3 EXPERIMENTAL EVALUATION

In this section, we present details of the experimental study of the proposed DAG-Explainer. We intend to answer the questions below through the evaluation:

1) With respect to the quantitative metrics, how does DAG-Explainer perform compared to the existing methods?
2) Does DAG-Explainer output meaningful and convincing explanations compared to the existing methods?
3) What is the difference of the explanations for different GNNs on the same dataset?
4) How does DAG-Explainer perform if edge information (*e.g.*, edge labels) in the graph is given additionally?
5) Compared with existing methods, can DAG-Explainer better help users understand the model?

Dataset	# Nodes	# Edges	GNN Accuracy		
			GCN	GIN	GINE
isAcyclic	30.04	28.46	0.992	0.983	-
MUTAG	17.93	19.79	0.963	0.979	-
Highschool	13.71	15.85	-	-	0.999

Table 1: Properties of the datasets and accuracy of the pretrained GNNs. Numbers of nodes and edges are averaged values.

3.1 Datasets and Competitors

3.1.1 Datasets. The proposed method is evaluated on three datasets, the isAcyclic dataset, the MUTAG dataset and the Highschool dataset. The first two are utilized to study the effectiveness of explainability techniques in a common setting of both datasets and GNN models. Meanwhile, the Highschool dataset is specifically prepared to evaluate the proposed method when edge label information is also available. Statistics of the datasets and corresponding GNN accuracies are shown in Table 1, we give the details below.

isAcyclic *dataset.* This datasets is a synthetic dataset designed by Hao, et al. [66] specifically for evaluating model-level explanations, in which the ground truth of explanations is prepared. Each graph in the dataset is labeled either *Cyclic* or *Acyclic* according to whether it encloses cycle(s). The *Cyclic* class contains grid-like structures, circle structures, circular ladder structures or wheel-like structures; whereas the *Acyclic* class consists of star-like structures, binary tree structures, path-like structures and full rary tree structures [51].

MUTAG *dataset.* This dataset consists of molecule structures of chemical compounds, where node labels represent atom types and edge labels represent chemical bonds. There are in total seven elements in the dataset: carbon, nitrogen, oxygen, chlorine, fluorine, bromine and iodine. Each instance is labeled according to its mutagenic effect on a bacterium [15]. The dataset is widely used in GNN explanation works [65, 66, 68] given its distinct structural feature with respect to the underlying domain (see Figure 1(b)).

Highschool *dataset.* We design this dataset based on the Highschool_ct2 dataset prepared by Oettershagen, et al. [39]. Each graph in the dataset is a face-to-face contact network between students in a Highschool over a period of seven days, on which epidemic spreading processes are simulated using the Susceptible-Infected Model [6]. Each person in a network is either *susceptible* or *infected*, an infected person will disseminate the epidemic to a susceptible person via face-to-face contact with a probability of $0 < p \leq 1$. The simulation involves two kinds of epidemics spreading across the social network, one is an *ordinary* epidemic, whose dissemination probability is set to 0.2, while for the *high-risk* epidemic, the probability is set to be 0.8, this means the latter is significantly more infectious. Simulated spreading stops when a threshold of infected node number is reached, thus classification by simply counting the infected nodes is not feasible. We relabel the edges based on the infection risk of the contact. Note that two students may have multiple face-to-face contacts during the seven days, while they may become infected at a specific time point. For each pair of connected nodes, we label the edge between them as the number of contacts when they hold different labels, *i.e.* one is *infected* and the other is not; such contacts are dangerous since dissemination may happen. Thus the higher the edge label value, the more dangerous it is for the disinfected student. Subsequently, edge labels contain important information on the propagation path of the epidemic since it distinguishes no-risk meets (edge label 0) and risky meets (edge label $\geq$ 1), a GNN model considering both node label and edge label is desirable on this dataset.

3.1.2 Competitors. We compare DAG-Explainer with 3 baselines.

XGNN [66] is a learning-based explainer, we compare DAG-Explainer to XGNN on the isAcyclic and MUTAG datasets. The method requires user-set parameters, including number of nodes in an explanation and the label of the initial node for graph generation. Size of the explanations and diversity of node labels will affect the information contained in the structure and node features, which further affects the model recognition. Hence, for fair comparison, we have the following settings. On the isAcyclic dataset, we follow XGNN to set the number of nodes in candidates for DAG-Expaliner to be a specific value in {3, 4, ..., 7}; correspondingly, we use *isAcyclic-n^** to refer to experiments on the isAcyclic dataset using only candidates with n nodes. On the MUTAG dataset, we set the number of nodes for XGNN as the same number of outputs as DAG-Explainer, and run XGNN multiple times using an initial node with *every* label in the compared explanation generated by DAG-Explainer. Regarding the Highschool dataset, XGNN is not designed to be capable of choosing an edge with a certain label during the graph generation, thus we cannot use XGNN on the Highschool dataset.

Glocal [32] is a graph-mining-based explainer, the method includes a pruning strategy guided by the GNN's behavior when a candidate explanation is occluded from the original instances. We compare our explainer to Glocal on all datasets, including *isAcyclic-n^**.

Optima baseline. We further implement a power set search algorithm to find the optima for the objective function on *isAcyclic-n^**, where the candidate spaces are rather small and exhaustive enumeration is feasible. We denote this baseline as **OPT**.

3.2 GNN Models to Be Explained

In this work, experiments are conducted to explain GNNs in graph classification task. While, without loss of generality, DAG-Explainer can be effortlessly adopted in node-level and edge-level tasks for various graph-based applications by considering the computational graphs of the input instances. We choose two widely employed

models, GCN [26] and GIN [61] to explain. Other GNNs can also be used as explained objects, we simply choose these two for the purpose of demonstration as they are classic models. We also implement a GNN model that additionally consumes edge labels to answer the question "*How does DAG-Explainer perform if edge information in the graph is given additionally?*". We uniformly use two fully connected layers as the final classifier for all the GNNs, which are implemented using Pytorch [42] and trained using the Adam optimizer [25]. All models are trained to a reasonable accuracy to ensure they have learned the knowledge from the datasets. The training accuracies are reported in Table 1. Details of the network structures are introduced in the following.

GCN model. We train two GCNs [26] for the isAcyclic and MUTAG datasets accordingly. For the former, a 2-layer GCN is trained with hidden dimensions as 8 and 16 respectively, the dimension of the fully connected layers is set to 32. For the MUTAG dataset, we train a 3-layer GCN with hidden dimensions all equal to 128 and set the dimension of fully connected layers to 64. In both models, Relu is adopted as the non-linear function and global mean pooling is employed. Node degree is used as node feature for the isAcyclic dataset[2]; as for the MUTAG dataset, we use the one-hot feature to encode the node labels, *i.e.* atom types in the molecular. For fair comparison with the baseline work XGNN, we also ignore the edge label in the MUTAG dataset and investigate the GNN's learned knowledge from graph structures and node labels only, the same settings apply to the GIN model as well.

GIN model. We additionally train two GIN models [61] on the two datasets above with the same structure aiming to inspect possible different explanations for different GNNs on the same dataset. We build 3-layer GIN models with the dimension as 64 for all hidden layers and the final fully connected layers. We use Relu for both datasets as the non-linear activation function.

GINE model. For the Highschool dataset, to make use of the edge label, we employ the GINE model proposed in [35], which is an extension of the GIN model [61] equipped by an additional utility to consume edge features besides node features. In each layer of GINE model, a 2-layer Multi-layer Perceptron (MLP) is used to process the edge features to match the dimension of node features, finally the network sums the two to give an embedding containing both node and edge information. Moreover, global mean pooling is adopted to pool the leaned embeddings. The complete GINE model uses 3 layers described above with the hidden layer dimension equals 32. Node labels and edge labels are encoded with one-hot features.

3.3 Setup of Algorithmic Experiments

Candidates Generation. Before running the proposed algorithm, explanation candidates need to be prepared. In the area of graph theory, there exists a rich body of research works proving preeminent tools for mining subgraphs [23, 62, 63]. In regard of preparing candidate subgraphs, we utilize gSpan [62] for fair comparison as it is the mining algorithm Glocal is based on. We highlight that the candidate generation procedure is not a part of our contribution, and advanced mining techniques developed in the future can be further plugged in to improve the preprocessing practice. However, this is out of the scope of this work. Below we detail the settings of our experiments. We set the number of nodes in the explanation candidates to be between 3 and 7, which is a common setting for finding model-level explanations [66]. Following the conversion [48], the support threshold is set to 1% for the MUTAG dataset and 10% for the other datasets. The same settings apply to Glocal.

Algorithm Setting. The objective function (2) to be optimized contains parameters λ_i. Coordinate ascent has been proved to be effective for tuning the parameters equipped with cross-validation requirements [7, 27]. In practice, depending on the application scenarios, users may prefer model-level explanations with certain characteristics. For example, if users prefer explanations that explore more knowledge models learned from the data, the overall support should be high; else if users want to know more about the difference between classes, the average denial should be low. We introduce a general guideline on setting up DAG-Explainer in our experiment, but users can always adapt the method to meet practical needs. Specifically, we conduct the same excise to sample 20% of candidates of the isAcyclic and MUTAG datasets, 10% of candidates of the Highschool dataset for estimating the parameters using coordinate ascent while preserving the following constrains: the average GNN score of the output explanation set should be greater than 0.95, the self-support ratio must be larger 0.5 and the size of the output set should be smaller than 10% of the total number of instances in the underlying class. We tune the parameters to minimize the average denial ratio. The cardinality constraint k is set to $k = \left\lceil \frac{|C|+1}{2} \right\rceil$ as required by Corollary 2.10.

Results Processing. As the algorithm is randomized, we run the method on each dataset for 1000 iterations and take the mean values for quantitative metrics. Majority voting is used to select the final explanation set, which will also be evaluated. We compute the vote of each single explanation as below:

$$vote(e) = \sum_{e \in \mathcal{E}} \frac{1}{|\mathcal{E}|},$$

where $\mathcal{E}$ are the outputs of all iterations running the algorithm. Then the final explanation set is selected as:

$$\mathcal{E}_{selected} = \underset{\mathcal{E}'s}{\operatorname{argmax}} \frac{\sum_{e \in \mathcal{E}} vote(e)}{|\mathcal{E}|}.$$

3.4 User Study for Model-level Explanations

In addition to algorithm evaluation, we also design a user study to verify whether DAG-Explainer can really help users understand GNNs in practice. Existing user studies for explaining GNNs are limited to asking 10 general users to rate the explanations [54], which can be subjective and lack real measures of "human understanding" of model behavior. Hence, we design a novel evaluation scheme to evaluate global explanations. Specifically, we randomly sample 20 instances from each dataset and present them to the user along with model-level explanations. We then ask the user to predict the GNN's predictions for these instances based on their understanding of the explanations; if the explanations really help users interpret the model, they should be able to predict the model's decisions.

[2]Regarding the comparison with XGNN, official implementation of the method is open to the public for the MUTAG dataset only, it is unclear how XGNN handles node features for the isAcyclic dataset during explanation generation as node degrees change when the structure grows, we use the results reported in the paper for comparative study, which explains a GCN model. For evaluating other methods on the isAcyclic dataset, we build a GCN with the same structure as the one used for XGNN; ordinary p and italic p are then used in Figure 3 to differentiate scores from the two GCN models.

For the isAcyclic dataset and the Highschool dataset, users need to judge the GNN's predicted class for each instance given explanations of two classes. Therefore the task is formulated as a classification task with the GNN's prediction as ground truth; accuracy is used as the evaluation metric. Whereas for the MUTAG dataset, users need to select instances that they believe the GNN will predict as *mutagenic*, given explanations for the class (base class has no explanation). The task is multiple choice, and we measure the accuracy, false positive rate (FPR), false negative rate (FNR), and average number of selected answers. We recruit 40 postgraduate students from the Department of Computer Science and Engineering of the HKUST (35 PhD students and 5 Master students) to join the user study. To avoid users guessing the true label instead of predicting GNN's decision, we use *class A* and *class B* for notation to eliminate the semantic information contained in the class names in the study. The online survey is available here.

3.5 Evaluation and Discussion

To conduct our evaluation, a scheme of quantifying the explainability of the final output needs to be introduced following the guidance of the three properties from both the model and data perspectives. For simplicity of notation, we use $\phi(e)$ to denote $\phi(s)[\hat{y} = c]$ for an explanation candidate $e = (s, c)$, *i.e.* the GNN score for the underlying class; furthermore, the shorthand p is utilized to denote the score in figures. Below we introduce the two metrics for measuring the explainability of the final explanation set.

- *Overall recognizability*. From the model's perspective, GNN recognition for the explanation set is the primary evaluation metric, which is calculated as

$$\phi(\mathcal{E}) = \frac{\sum_{e \in \mathcal{E}} \phi(e)}{|\mathcal{E}|}.$$

  In fact, the GNN score of the target class is used as an evaluation metric in many existing model-level methods [37, 38, 40, 66].
- *Data-awareness*. From the data perspective, the candidate mining procedure has guaranteed that explanations are compliant with the data distribution. Hence we measure the discrimination level for data-aware explanations by a pair of metrics, one is the self-support ratio computed as

$$\text{Self-}sup.(e) = \frac{|support(e)|}{|support(e)| + |denial(e)|},$$

  and the other one is Self-*den*., which simply equals 1 − Self-*sup*. For the outputs by XGNN, we run the subgraph isomorphism checking algorithm VF2 [13] to measure the data-awareness.

Moreover, the number of explanations contained in the final output is also considered for evaluating our method; while XGNN generates one single explanation in each run and gives different outputs in multiple runs, it is meaningless to consider the number of explanations generated as it is a user-set parameter. Additionally, the size of the structure is compared. We use the number of edges instead of nodes as size measurement for distinguishing different subgraphs induced by the same set of nodes. We also test the efficiency of different methods by measuring running time. All tests are conducted in CentOS Linux 7 (Core) (x86-64) on a machine with an 8-core Intel(R) Xeon(R) Silver 4210 CPU@2.20GHz, 8 NVIDIA GPUs (all GeForce RTX 2080 SUPER) and 225 GB of RAM.

3.5.1 Quantitative Results. The quantitative results on the isAcyclic, MUTAG and Highschool datasets are reported in Table 2. Visualization of the outputs with quantitative evaluations on the isAcyclic and *isAcyclic-n** datasets are shown in Figure 4 and Figure 3, respectively. In these figures, we report GNN score and Self-*sup*. (SS.) of each output, while Self-*den*. is omitted for simplicity as it equals 1-Self-*sup*.. **Green** (**orange**) background are used for explanation of GIN (GCN) model. Outputs of DAG-Explainer, Glocal and XGNN are shown with **blue**, **yellow** and **red** nodes, accordingly. For both GIN and GCN, **OPT** and DAG-Explainer give exactly the same outputs, thus **OPT**'s visualization are not shown separately. Additionally, running time on *isAcyclic-n** dataset are reported in Table 3.

Model recognition. Overall DAG-Explainer produces explanations with comparable GNN prediction scores to XGNN using less edges. The final output of our method (DAG-*final*) constantly outperforms Glocal. Moreover, our method gives exactly the same output as the optima baseline on both the isAcyclic (see Figure 4) and *isAcyclic-n** (see Figure 3) datasets.

Error analysis. Notice in Table 2, the average score in multiple runs of DAG-Explainer (DAG-*avg*.) is slightly lower than Glocal in three cases: GCN-*cyclic*, GIN-*acyclic* and GCN-*mutagenic*. The reason is that the GNN scores of candidates in these cases are generally very low (mean: 0.890, 0.877 and 0.901). Note that GNNs are complex models and its prediction score of candidates is not necessarily high. Nonetheless, since users only refer to the final output, we believe the proposed method is decent as DGA-*final* gives superior performance. In Figure 3, all outputs from DAG-Explainer retain a high GNN score with only a few exceptions: the 5-node explanation of the *cyclic* class for the GIN model (Figure 3(a), Row 1, Column 2), 3- and 4- node explanations of the *acyclic* class for GCN^3 (Figure 3(b), Row 1, Column 5-6). Their prediction scores are rather low, yet these are the highest ones in the candidates space. Thus we believe catching the highest score is acceptable. The same reason leads to the absence of candidates with 5- and 7- nodes of the *cyclic* class for the GCN (Figure 3(b), Row 1, Column 2&4), where the mined structures are all predicted as the other class.

Notice that the number of generated explanations in the output set by DAG-Explainer is generally small, except for the GCN in the *mutagenic* class in the MUTAG dataset. However, the final output drawn by the majority voting only contains 4 structures. That is because high-quality explanations have many repeating occurrences in multiple runs, and they offer superior global explainability.

Data-awareness. Overall, our method (both DAG-*avg*. and DAG-*final*) outperforms the two competitors in data-awareness metrics, *i.e.*, DAG-Explainer retains the highest (lowest) Self-*sup*. (Self-*den*.).

Error analysis. We notice that outputs of our explainer give relatively high values in Self-*den*. for the *acyclic* class in the isAcyclic and *isAcyclic-n** datasets explaining both GCN and GIN. It is because typical acyclic patterns are usually subgraphs of cyclic patterns, *e.g.* star-like structures *v.s.* wheel-like structures (see Figure 3). Meanwhile, data-awareness of the explanations also turns out not so well on the Highschool dataset. It is because the epidemic spread is simulated on observed social networks, only node and edge labels are affected while patterns of student's activity remain the same. Thus, the two classes do not differ in topology distribution, but in node and edge labels only. Moreover, social networks are generally

isAcyclic - *candidate generation time: 3290.33s*							
model-class	**explainer**	$\phi(\mathcal{E})$	Self-$sup.$	Self-$den.$	$\lvert edges \rvert$	Time/run	$\lvert\mathcal{E}\rvert$
GCN *cyclic*	Glocal	1.000	1.000	0.000	4.000	1876.88s	1
	DAG-*avg.*	0.954	1.000	0.000	6.516	0.0012s	3.622
	DAG-*final*	1.000	1.000	0.000	6.000	-	1
GCN *acyclic*	Glocal	0.521	0.467	0.531	3.000	2300.81s	1
	DAG-*avg.*	0.992	0.582	0.418	4.000	0.018s	1.475
	DAG-*final*	0.957	0.826	0.174	5.000	-	1
GIN *cyclic*	Glocal	0.746	1.000	0.000	5.500	2289.34s	2
	DAG-*avg.*	0.954	1.000	0.000	6.516	0.005s	3.622
	DAG-*final*	1.000	1.000	0.000	7.000	-	1
GIN *acyclic*	Glocal	0.999	0.511	0.489	3.750	3107.84s	4
	DAG-*avg.*	0.991	0.585	0.415	5.147	0.012s	1.551
	DAG-*final*	1.000	0.826	0.174	5.000	-	1
MUTAG -*candidate generation time: 7.14s*							
model-class	**explainer**	$\phi(\mathcal{E})$	Self-$sup.$	Self-$den.$	$\lvert edges \rvert$	Time/run	$\lvert\mathcal{E}\rvert$
GCN *mutagenic*	Glocal	1.000	0.665	0.335	3.000	6.893s	1
	XGNN	1.000	OOD	OOD	5.900	17.440s	-
	DAG-*avg.*	0.997	0.936	0.064	4.753	0.214s	18.535
	DAG-*final*	1.000	1.000	0.000	5.000	-	4
GIN *mutagenic*	Glocal	0.959	0.333	0.667	5.000	5.515s	1
	XGNN	1.000	OOD	OOD	5.429	15.247s	-
	DAG-*avg.*	1.000	0.917	0.083	4.704	0.011s	3.214
	DAG-*final*	1.000	1.000	0.000	4.500	-	2
Highschool - *candidate generation time: 537.39s*							
model-class	**explainer**	$\phi(\mathcal{E})$	Self-$sup.$	Self-$den.$	$\lvert edges \rvert$	Time/run	$\lvert\mathcal{E}\rvert$
GINE *ordinary*	Glocal	0.584	0.537	0.463	3.000	343.34s	2
	DAG-*avg.*	0.930	0.867	0.133	5.066	23.284s	7.167
	DAG-*final*	0.918	0.804	0.196	5.000	-	4
GINE *high-risk*	Glocal	0.901	0.524	0.476	2.000	324.59s	2
	DAG-*avg.*	0.972	0.789	0.211	7.760	23.093s	3.940
	DAG-*final*	0.979	0.800	0.200	8.000	-	2

Table 2: Quantitative evaluation on the MUTAG, isAcyclic and Highschool datasets. For DAG-Explainer, we report both the averaged metric values among the 1000 runs and evaluation on the final output. Self-*sup.*, Sefl-*den.* and number of edges in individual explanations ($\lvert edges \rvert$) are averaged over the entire output set. OOD stands for Out-of-Distribution, which means the structure does not exist in the data. For results of *isAcyclic-n**, see Figure 3.

complicated, where small substructures tend to be frequent. To conclude, we believe the data-awareness level of DAG-Explainer is reasonable. The main problem with Glocal are the high Self-*den.* values, which indicates that its output is barely discriminative between classes. XGNN suffers from the Out-Of-Distribution (OOD) problem, which means its output does not exist in the graph data. More details are given in Section 3.5.2 Qualitative Analysis.

Running time. We report the candidate generation cost (gSpan mining time) in Table 2 and Table 3. On the MUTAG dataset, DAG-Explainer outperforms XGNN even with candidate generation time included. Glocal is obviously much more costly than our method considering only the explainer running time. Although Glocal adopts pruning strategy so that the substructure mining time is shorter than the candidate generation time in DAG-Explainer, it needs to be rerun every time it explains a new GNN. While candidate generation for DAG-Explainer on a dataset can be used to explain *any* GNN trained on the dataset. This is extremely helpful in a situation where different GNNs need to be compared on the same dataset. For instance, if one intends to facilitate neural architecture search (NAS) for GNNs [22] with explainability, once the candidates are prepared, DAG-Explainer can fully enjoy its high efficiency in repeating explaining searched GNN architectures.

model *class*	**method**	***n*=** 3	4	5	6	7
GIN *cyclic*	**OPT**	NC	0.0003s	0.0003s	0.0019s	0.0020s
	Glocal	NC	4.66s (15533.3x)	19.32s (64400.0x)	137.99s (72626.3x)	794.88s (397440.0x)
	DAG	NC	0.0003s (1.0x)	0.0004s (1.3x)	0.0012s (0.6x)	0.0012s (0.6x)
GIN *acyclic*	**OPT**	0.0004s	0.0010s	0.0011s	0.0283s	2.1137s
	Glocal	2.028s (5070.0x)	5.103s (5103.0x)	17.34s (15763.6x)	161.34s (5701.1x)	933.27s (441.5x)
	DAG	0.0003s (0.7x)	0.0011s (1.1x)	0.0011s (1.0x)	0.0028s (0.1x)	0.0047s (0.0x)
GCN *cyclic*	**OPT**	NC	0.0004s	NC	0.0018s	NC
	Glocal	NC	4.331s (10827.5x)	18.798s -	138.230s (76794.4x)	827.610s -
	DAG	NC	0.0004s (1.0x)	NC	0.0016s (0.9x)	NC
GCN *acyclic*	**OPT**	0.0004s	0.0011s	0.0020s	0.0303s	3.2205s
	Glocal	1.873s (4682.5x)	4.125s (3750.0x)	21.348s (10674.0x)	141.973s (4685.6x)	893.439s (277.4x)
	DAG	0.0004s (1.0x)	0.0011s (1.0x)	0.0015s (0.8x)	0.0031s (0.1x)	0.0042s (0.0x)
DAG cand. gen.		1.12s	3.18s	17.23s	133.93s	801.98s

Table 3: Running time on the *isAcyclic-n** dataset. For DAG-Explainer and Glocal, we report the multiple values relative to the OPT baseline in brackets. Candidate generation time for DAG explainer is shown in the last row.

In Table 3 presents running time on *isAcyclic-n**, the multiple values relative to the **OPT** baseline are also reported in brackets for DAG-Explainer and Glocal. Glocal is too computationally expensive compared to the other two methods. DAG-Explainer takes much less time than **OPT** when $n \geq 6$. In other cases, the time differences are rather small, we run the two methods 10,000 times and conduct Student's t-test [41] with a significance level of 0.05 to validate if the efficiency improvement is significant, and the p-values are presented in the Supplementary Material [1]. The null hypothesis (*the means of the two populations are equal*) is not rejected for GCN and GIN when $n \leq 6$ in the *cyclic* class and $n \leq 5$ in the *acyclic* class. This is because the underlying candidate space is very small (≤ 10) in these cases. Otherwise, the performance gain of DAG-Explainer is shown to be significant. When the candidate space is large ($n = 7$), the efficiency gap between the two methods becomes obvious.

3.5.2 Qualitative Analysis. We further evaluate the explanations qualitatively via visualization and discuss the findings.

isAcyclic dataset. Explanations on isAcyclic are shown in Figure 4. DAG-Explainer and **OPT** output the same results, *i.e.*, a ladder structure for GIN and a ring-like structure for GCN as *cyclic* explanation;

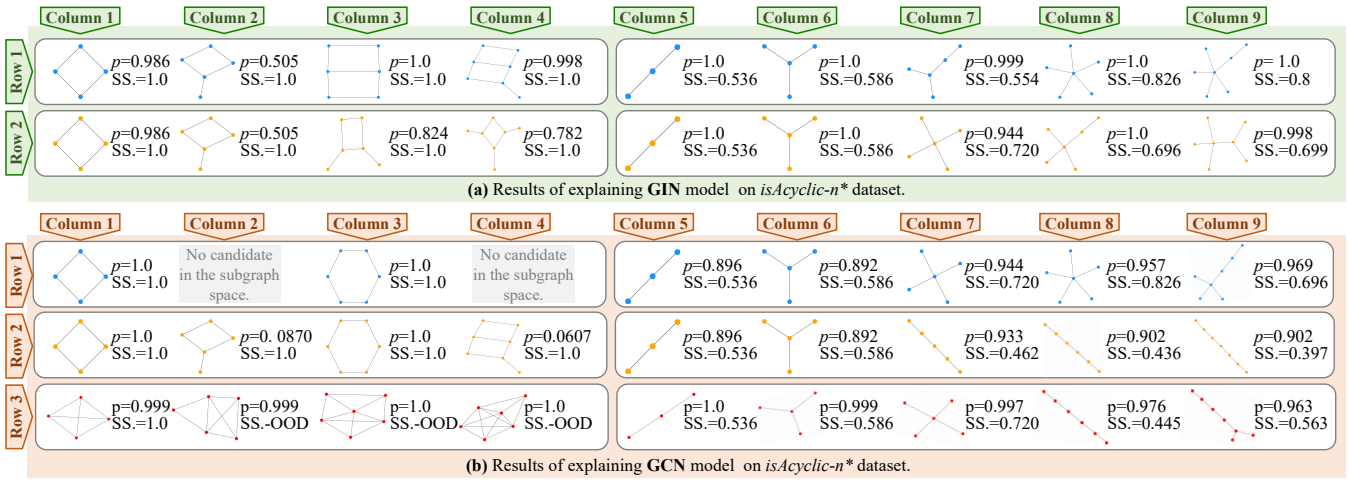

Figure 3: Results on *isAcyclic-n* dataset. Explanations with specified numbers of nodes together with GNN score and Self-*sup.* (SS.) are reported. For XGNN, explanations for a GCN model from the official paper are presented[3]. On the left (right) shows the explanations for the *cyclic* (*acyclic*) class. For both GIN and GCN, OPT and DAG-Explainer give exactly the same outputs, thus OPT's visualization are not shown separately.**

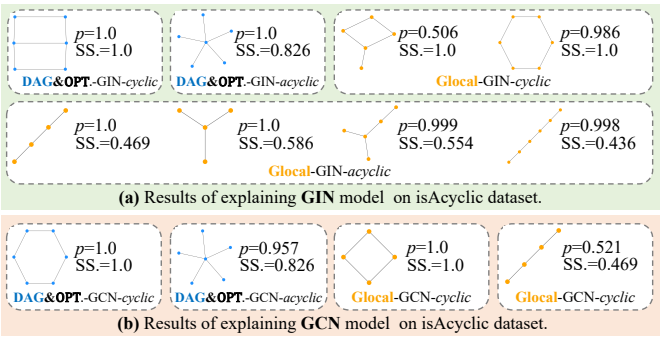

Figure 4: Results on isAcyclic dataset. Visualization of the explanations with GNN score and Self-*sup.* (SS.) are reported. For both GNNs, OPT and DAG-Explainer give exactly the same outputs.

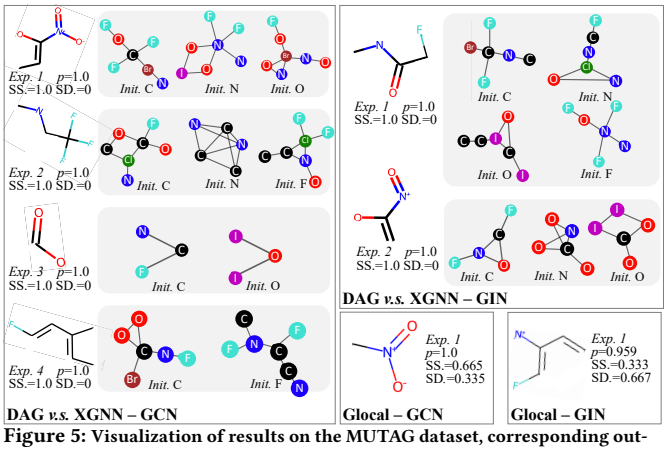

Figure 5: Visualization of results on the MUTAG dataset, corresponding outputs by XGNN are shown in shaded boxes. DAG-Explainer generates for both GNNs faithful and data-aware explanations. Every output by XGNN receives GNN score 1.0, yet they are all Out-Of-Distribution. Glocal produces explanations with high GNN scores, yet they do not discriminate between classes.

meanwhile, the two methods both output a star-like structure as *acyclic* explanation for the two GNNs, which concisely summarizes the synthesizing rule of the dataset. Glocal finds circle structures for the *cyclic* class explaining GIN and GCN, yet it misses ladder-shaped structures. Though Glocal's outputs explaining the *acyclic* class for both models fit the rule of the dataset, its measurements in all quantitative metrics are outperformed by DAG-Explainer.

isAcyclic-n* **dataset**. The results are shown in Figure 3. Outputs of **OPT** are, again, the same as DAG-Explainer's, thus not shown separately. For the *cyclic* class, both DAG-Explainer and Glocal capture circle- and ladder- structures for the two GNNs (Figure 3 (a)&(b), Row 1-2, Column 1-4), which is consistent with the ground truth of the synthetic dataset, yet DAG-Explainer has higher model recognition. In contrast, most of XGNN's outputs (Figure 3(b), Row 3, Column 2-4) are OOD, meaning that they do not exist in the data, hence one cannot argue these explanations are valid. For the *acyclic* class, the three methods find explanations with similar structures. However, DAG-Explainer successfully identifies the star-like structure (Figure 3(a)&(b), Row 1, Column 8) while other baselines fail. Moreover, Glocal (Figure 3(b), Row 2, Column 5, 7-9) and XGNN (Figure 3(b), Row 3, Column 5, 8-9) prefers path structures, which are not discriminative between classes. The results in Self-*den.* are generally not decent due to the reason discussed earlier: acyclic structures are often subgraphs of cyclic structures.

MUTAG dataset. The results for explaining GCN and GIN on the *mutagenic* class are shown in Figure 5. We visualize outputs of DAG-Explainer and Glocal using SMILES encoding [57], while the chemical ball-and-stick model is used for XGNN due to its output structures being illegal in chemical domain and SMILES encoding is not feasible. DAG-Explainer outputs explanantions with perfect data-awareness (Self-*sup.*=1, Self-*den.*=0) with GNN scores all equals 1. Interestingly, these explanations by DAG-Explainer for the GIN model are very similar to some for the GCN model but smaller, *i.e.* GIN-*Exp. 1* to GCN-*Exp. 2* and GIN-*Exp. 2* to GCN-*Exp. 1*. We suppose such results show the similarity and difference between decision making mechanisms of GCN and GIN models on the MUTAG dataset, which can help human users understand both GNNs better. Meanwhile, the outputs of XGNN are all OOD, meaning they do not exist in the input data (compare them to instances in Figure 1(b)). This means the rationality of these explanations are doubtful,

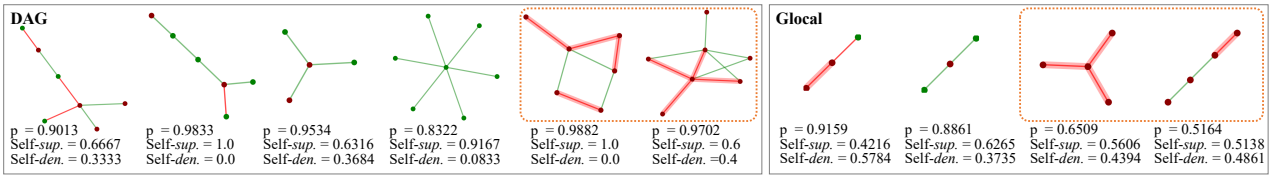

Figure 6: Results on the Highschool dataset. Explanations for the *high-risk* epidemic class are shown in the dotted box, the others are for the *ordinary* epidemic class. Epidemic propagation is indicated by thickened edges in the semi-transparent red.

and they confuse users in our user study (to be discussed later). Glocal successfully finds high-score explanations for both GNNs, yet they are not discriminative between classes (high Self-*den.*); in fact, Glocal's output for GCN exists in *every* instance in the MUTAG dataset regardless of their labels .

Highschool dataset. The results are shown in Figure 6. We utilize a visualization scheme specifically designed for this dataset to facilitate human understanding. *Infected* (*w.r.t. non-infected*) nodes are plotted in red (*w.r.t.* green). Safe contact is shown using green edges while risky contact is represented by the thin red edge; the darker the color, the riskier the contact is (see Section 3.1.1). We thicken an edge in a semi-transparent red color if it is a risky contact and both its ends are infected, which means the infection between the two students was very likely from this contact, hence it is part of the epidemic propagation. With the aid of such visualization, one can easily see the spreading patterns of the disease.

Regarding DAG-Explainer's outputs, dissemination is not captured in explanations of the *ordinary* class as we expected. None of the outputs contain any propagation path; moreover, two out of four explanations contain red edge(s) that are not thickened, which indicates that the epidemic did not successfully spread even when risky contact happened. According to the explanation, GINE model also recognizes structures with safe contacts only as "ordinary". Though such pattern does not necessarily lead to the idea of "not risky" in human reasoning, considering that GNN is a black-box model and its decision making criteria may differ from humans, we suppose such a case can be regarded as an interesting finding that tells the difference between human cognition and machine recognition. As for Glocal's outputs, though a pure-safe-contact explanation is also identified, the other one does not provide any insight as dissemination happens in only one out of its two risky contacts. While in the *high-risk* class, we observe evident patterns of propagation in DAG-Explainer's output, every risky contact commits to an infection; moreover, our method outperforms Glocal in quantitative results. The observation precisely aligns with the ground truth that *high-risk epidemic* is more infectious and dangerous, hence it is demonstrated that DAG-Explainer produces meaningful explanations that verify the trustworthiness of GNN.

3.5.3 User Study Evaluation. Accuracy of users (binary classification) on the isAcyclic and Highschool datasets are reported in Table 4, and the results on the MUTAG dataset (multiple choice) are reported in Table 5. On the isAcyclic and Highschool datasets, users are able to correctly predict the GNN's decision on more than 85% of instances based on explanations by DAG-Explainer. On the MUTAG dataset, the accuracy is generally lower than that on the other two datasets, because the MUTAG dataset requires chemical knowledge. Nonetheless, DAG-Explainer always retains the highest accuracy. On the MUTAG dataset, XGNN has highest false negative rate, because its outputs are all OOD, users can hardly find any instances similar to the explanations. For the same reason, its average number of selected instances is the smallest (< 2 out of 20). Glocal has the highest false positive rate and number of selected instances, because its outputs are not discriminative between classes and users cannot tell the difference between them based on the explanation. Overall, our method outperforms the other two baselines in helping a user anticipate the model's behavior on the three datasets.

	isAcyclic - GCN		isAcyclic - GIN		Highschool - GINE	
	Glocal	DAG	Glocal	DAG	Glocal	DAG
Acc.	0.898	0.910	0.880	0.881	0.818	0.850

Table 4: User study results on the isAcyclic and Highschool dataset.

	GCN				GIN			
	Acc.	FPR	FNR	$\lvert Resp. \rvert$	*Acc.*	FPR	FNR	$\lvert Resp. \rvert$
DAG	0.623	0.163	0.650	5.133	0.607	0.067	0.813	2.533
XGNN	0.552	0.070	0.920	1.500	0.560	0.040	0.937	1.033
Glocal	0.462	0.913	0.193	17.20	0.588	0.097	0.817	2.800

Table 5: User study result on the MUTAG dataset.

Discussion. In general, whether the explanation is helpful depends on a) *if the users can understand the output*, and b) *if it helps the users compare the decision-making mechanism of the model with human cognition.* The degree of incomprehension of the explanation mainly depends on the degree of abstraction of the graph itself. Users can easily understand visual datasets such as social networks. As for graph data that requires domain knowledge, it is generally believed that the GNN users have the domain knowledge of the application. Meanwhile, global explanations can help users understand general behavior of the model. For example, epidemiologists can verify reliability of the model by observing whether the patterns of disease transmission shown by the explanation match typical spreading patterns [5, 55]. For more discussion, see Section 6.3.2.

4 RELATED WORK

Graph Neural Networks. The research on GNN origins from [20, 50]. These models can capture both node features and graph topology, hence becoming a solid tool for handling graph-structure data. Notable GNN variants include Graph Convolution Networks (GCNs) [26], which extend the conversational technique in deep learning to graph setting; Graph Attention Networks (GATs) [53], which introduce a self-attention mechanism to GNN for distinguishing important neighbors; and Graph Isomorphism Network (GIN) [61], in which researchers study the express power of GNNs compared to the Weisfeiler-Lehman graph isomorphism test. The outstanding performance of GNNs leads to a large scale of applications in various downstream tasks [9, 10, 30, 59, 70, 71].

Explainability of GNNs. Explainibility of GNN can be categorized into two groups: *instance-level* methods and *model-level* methods. Given the input data, instance-level techniques [29, 31, 52, 54, 65, 68] have been present main stream of GNN explanation, which aim to

acquire explanations for a target instance. On the contrary, model-level methods are still under shallow exploration, which target at producing general and input-independent explanations that interpret the overall behavior of the model. Such explanations can provide high-level and intrinsic reasoning of the GNN's prediction. In recent works PGExplainer [31] and ReFine [56], researchers claim that the method can generate explanations with global knowledge of the GNN by allowing the explainer to be trained on multiple instances. However the output is still based on one single instance, hence the explanation is not *input-independent* and we do not consider the technique model-level in the strictest sense. In the pioneering work of GNN global explainability is XGNN [66], the authors propose to explain GNNs via graph generation using a reinforcement learning framework. Glocal [32] is a recent work that equips a subgraph mining technique with pruning strategy for finding explanations that are both faithful to the model and frequent in the explained class.

Explainability in Database Intensive research efforts have also been devoted to studying the explainability in database. Methods on explaining query results are developed based on various techniques such as clustering [34], provenance [3, 4, 64], responsibility [33], and so on. Counterfactual-guided causality is employed in data analysis [46]. Explainability has also been considered in building database system [47] and debugging framework [44]. Both the machine learning and database communities show great interest in unveiling the working mechanisms of various models and systems.

Graph mining. There exists a rich body of graph mining research that help users understand the properties of graph data [23]. Classic graph mining tools include gSpan[62] and CloseGraph[63]. Recently, more and more mining techniques have been developed to boost the performance of this task on large scale datasets [36, 45].

5 CONCLUSION

Graph neural networks are widely employed and explanation techniques for GNNs have been desired. In this paper, we propose a model-level explanation technique, called DAG-Explainer. Specifically, we observe three properties of high-quality explanations: they should be highly recognized by the model and compliant to the data distribution, while being discriminative among classes. We design metrics to quantify the degree of an explanation retaining the three properties. We then define an objective function to find explanations that best explain the model. We prove that optimizing the objective is NP-hard, and adopt a randomized greedy algorithm to find a near optimal solution. Furthermore, we prove an improved theoretical guarantee of the approximation algorithm over the state-of-the-art best. Experimental results show that DAG-Explainer outputs meaningful and trustworthy explanations with decent quantitative evaluation results.

REFERENCES

[1] [n.d.]. Supplementary Materials. https://github.com/Gori-LV/DAG.

[2] Sergi Abadal, Akshay Jain, Robert Guirado, Jorge López-Alonso, and Eduard Alarcón. 2021. Computing graph neural networks: A survey from algorithms to accelerators. *ACM Computing Surveys (CSUR)* 54, 9 (2021), 1–38.

[3] Efrat Abramovitz, Daniel Deutch, and Amir Gilad. 2018. Interactive inference of SPARQL queries using provenance. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 581–592.

[4] Omar AlOmeir, Eugenie Yujing Lai, Mostafa Milani, and Rachel Pottinger. 2021. Summarizing Provenance of Aggregate Query Results in Relational Databases. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1955–1960.

[5] Roy M Anderson, Christophe Fraser, Azra C Ghani, Christl A Donnelly, Steven Riley, Neil M Ferguson, Gabriel M Leung, Tai H Lam, and Anthony J Hedley. 2004. Epidemiology, transmission dynamics and control of SARS: the 2002–2003 epidemic. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 359, 1447 (2004), 1091–1105.

[6] Yuan Bai, Bo Yang, Lijuan Lin, Jose L Herrera, Zhanwei Du, and Petter Holme. 2017. Optimizing sentinel surveillance in temporal network epidemiology. *Scientific reports* 7, 1 (2017), 1–10.

[7] Peng Bao, Weihui Hong, and Xuanya Li. 2021. Predicting Paper Acceptance via Interpretable Decision Sets. In *WWW (Companion Volume)*. ACM / IW3C2, 461–467.

[8] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. 2014. Submodular Maximization with Cardinality Constraints. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*. SIAM, 1433–1452.

[9] Leonid A. Bunimovich, Chi-Jen Wang, Seokjoo Chae, and Benjamin Z. Webb. 2018. Uncovering Hierarchical Structure in Social Networks Using Isospectral Reductions. In *IEEE/ACM 2018 International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*. IEEE Computer Society, 1199–1206.

[10] Feihu Che, Dawei Zhang, Jianhua Tao, Mingyue Niu, and Bocheng Zhao. 2020. ParamE: Regarding Neural Network Parameters as Relation Embeddings for Knowledge Graph Completion. In *AAAI*. AAAI Press, 2774–2781.

[11] Lin Chen, Moran Feldman, and Amin Karbasi. 2018. Weakly Submodular Maximization Beyond Cardinality Constraints: Does Randomization Help Greedy?. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research)*, Vol. 80. PMLR, 803–812.

[12] Diane J Cook and Lawrence B Holder. 2006. *Mining graph data*. John Wiley & Sons.

[13] Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. 2004. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence* 26, 10 (2004), 1367–1372.

[14] Abhimanyu Das and David Kempe. 2011. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975* (2011).

[15] Asim Kumar Debnath, Rosa L. Lopez de Compadre, Gargi Debnath, Alan J. Shusterman, and Corwin Hansch. 1991. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry* 34, 2 (1991), 786–797.

[16] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. 1991. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry* 34, 2 (1991), 786–797.

[17] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Pai-Shun Ting, Karthikeyan Shanmugam, and Payel Das. 2018. Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives. In *NeurIPS*. 590–601.

[18] Lukas Faber, Amin K. Moghaddam, and Roger Wattenhofer. 2020. Contrastive Graph Neural Network Explanation. In *Proceedings of the 37th Graph Representation Learning and Beyond Workshop at ICML 2020*.

[19] Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. 2011. Maximizing Non-monotone Submodular Functions. *SIAM J. Comput.* 40, 4 (2011), 1133–1153.

[20] M. Gori, G. Monfardini, and F. Scarselli. 2005. A new model for learning in graph domains. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* 2 (2005), 729–734 vol. 2.

[21] Norman Neill Greenwood and Alan Earnshaw. 2012. *Chemistry of the Elements*. Elsevier.

[22] Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems* 212 (2021), 106622.

[23] Chuntao Jiang, Frans Coenen, and Michele Zito. 2013. A survey of frequent subgraph mining algorithms. *Knowledge Engineering Review* 28, 1 (2013), 75–105.

[24] Kristian Kersting, Nils M. Kriege, Christopher Morris, Petra Mutzel, and Marion Neumann. 2016. Benchmark Data Sets for Graph Kernels. http://graphkernels.cs.tu-dortmund.de

[25] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[26] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net.

[27] Himabindu Lakkaraju and Cynthia Rudin. 2017. Learning cost-effective and interpretable treatment regimes. In *Artificial intelligence and statistics*. PMLR, 166–175.

[28] Heesang Lee, George L Nemhauser, and Yinhua Wang. 1996. Maximizing a submodular function by integer programming: Polyhedral results for the quadratic case. *European Journal of Operational Research* 94, 1 (1996), 154–166.

[29] Wanyu Lin, Hao Lan, and Baochun Li. 2021. Generative Causal Explanations for Graph Neural Networks. In *ICML (Proceedings of Machine Learning Research)*, Vol. 139. PMLR, 6666–6679.

[30] Dongsheng Luo, Yuchen Bian, Yaowei Yan, Xiao Liu, Jun Huan, and Xiang Zhang. 2020. Local Community Detection in Multiple Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, 2020*. ACM, 266–274.

[31] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized Explainer for Graph Neural Network. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*.

[32] Ge Lv, Lei Chen, and Caleb Chen Cao. 2022. On Glocal Explainability of Graph Neural Networks. In *DASFAA (1) (Lecture Notes in Computer Science)*, Vol. 13245. Springer, 648–664.

[33] Alexandra Meliou, Wolfgang Gatterbauer, Katherine F Moore, and Dan Suciu. 2010. The complexity of causality and responsibility for query answers and non-answers. *arXiv preprint arXiv:1009.2021* (2010).

[34] Aurélien Moreau, Olivier Pivert, and Grégory Smits. 2016. A clustering-based approach to the explanation of database query answers. In *Flexible Query Answering Systems 2015: Proceedings of the 11th International Conference FQAS 2015, Cracow, Poland, October 26-28, 2015*. Springer, 307–319.

[35] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. *CoRR* abs/2007.08663 (2020). arXiv:2007.08663

[36] Kazuya Nakagawa, Shinya Suzumura, Masayuki Karasuyama, Koji Tsuda, and Ichiro Takeuchi. 2016. Safe Pattern Pruning: An Efficient Approach for Predictive Pattern Mining. In *KDD*. ACM, 1785–1794.

[37] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. 2017. Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. IEEE Computer Society, 3510–3520.

[38] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*. IEEE Computer Society, 427–436.

[39] Lutz Oettershagen, Nils M. Kriege, Christopher Morris, and Petra Mutzel. 2020. Temporal Graph Kernels for Classifying Dissemination Processes. In *SDM*. SIAM, 496–504.

[40] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature visualization. *Distill* 2, 11 (2017), e7.

[41] Donald B Owen. 1965. The power of Student's t-test. *J. Amer. Statist. Assoc.* 60, 309 (1965), 320–333.

[42] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[43] Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. 2019. Explainability Methods for Graph Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*. Computer Vision Foundation / IEEE, 10772–10781.

[44] Romila Pradhan, Jiongli Zhu, Boris Glavic, and Babak Salimi. 2022. Interpretable data-based explanations for fairness debugging. In *Proceedings of the 2022 International Conference on Management of Data*. 247–261.

[45] Sayan Ranu and Ambuj K. Singh. 2009. GraphSig: A Scalable Approach to Mining Significant Subgraphs in Large Graph Databases. In *ICDE*. IEEE Computer Society, 844–855.

[46] Sudeepa Roy. 2022. Toward interpretable and actionable data analysis with explanations and causality. *Proceedings of the VLDB Endowment* 15, 12 (2022), 3812–3820.

[47] Sudeepa Roy, Laurel Orr, and Dan Suciu. 2015. Explaining query answers with explanation-ready databases. *Proceedings of the VLDB Endowment* 9, 4 (2015), 348–359.

[48] Hiroto Saigo, Sebastian Nowozin, Tadashi Kadowaki, Taku Kudo, and Koji Tsuda. 2009. gBoost: a mathematical programming approach to graph classification and regression. *Mach. Learn.* 75, 1 (2009), 69–89.

[49] Richard Santiago and Yuichi Yoshida. 2020. Weakly Submodular Function Maximization Using Local Submodularity Ratio. In *ISAAC (LIPIcs)*, Vol. 181. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 64:1–64:17.

[50] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80.

[51] James Andrew Storer. 2012. *An introduction to data structures and algorithms*. Springer Science & Business Media.

[52] Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. 2022. Learning and Evaluating Graph Neural Network Explanations based on Counterfactual and Factual Reasoning. In *WWW*. ACM, 1018–1027.

[53] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018*. OpenReview.net.

[54] Minh N. Vu and My T. Thai. 2020. 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*.

[55] Lin Wang, Yan Zhang, Zhen Wang, and Xiang Li. 2013. The impact of human location-specific contact pattern on the SIR epidemic transmission between populations. *International Journal of Bifurcation and Chaos* 23, 05 (2013), 1350095.

[56] Xiang Wang, Ying-Xin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. 2021. Towards Multi-Grained Explainability for Graph Neural Networks. In *NeurIPS*. 18446–18458.

[57] David Weininger. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* 28, 1 (1988), 31–36. https://doi.org/10.1021/ci00057a005 arXiv:https://pubs.acs.org/doi/pdf/10.1021/ci00057a005

[58] G Witten and G Poulter. 2007. Simulations of infectious diseases on networks. *Computers in Biology and Medicine* 37, 2 (2007), 195–205.

[59] Ning Wu, Wayne Xin Zhao, Jingyuan Wang, and Dayan Pan. 2020. Learning Effective Road Network Representation with Hierarchical Graph Neural Networks. In *KDD*. ACM, 6–14.

[60] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[61] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR*.

[62] Xifeng Yan and Jiawei Han. 2002. gSpan: Graph-Based Substructure Pattern Mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining*. IEEE Computer Society, 721–724.

[63] Xifeng Yan and Jiawei Han. 2003. CloseGraph: mining closed frequent graph patterns. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2003*. ACM, 286–295.

[64] Alexander Yao. 2022. Interactive Query Explanations Using Fine Grained Provenance. In *Proceedings of the 2022 International Conference on Management of Data*. 2536–2538.

[65] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating Explanations for Graph Neural Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*.

[66] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, 2020*. ACM, 430–438.

[67] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2020. Explainability in Graph Neural Networks: A Taxonomic Survey. *CoRR* abs/2012.15445 (2020). arXiv:2012.15445

[68] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2020. On Explainability of Graph Neural Networks via Subgraph Explorations. In *ICML (Proceedings of Machine Learning Research)*.

[69] Yue Zhang, David DeFazio, and Arti Ramesh. 2020. RelEx: A Model-Agnostic Relational Model Explainer. *CoRR* abs/2006.00305 (2020). arXiv:2006.00305

[70] Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhi-Ping Shi, Hui Xiong, and Qing He. 2020. Relational Graph Neural Network with Hierarchical Attention for Knowledge Graph Completion. In *AAAI*. AAAI Press, 9612–9619.

[71] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinform.* 34, 13 (2018), i457–i466.

[72] Tanli Zuo, Yukun Qiu, and Weishi Zheng. 2020. Neighbor Combinatorial Attention for Critical Structure Mining. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. ijcai.org, 3299–3305.

6 SUPPLEMENTARY MATERIAL

6.1 Proofs

In below we present the proof of Lemma 2.8 in Section 2.4.

Lemma 2.8. A submodular function $f : 2^E \to \mathbb{R}_+$ always satisfies

$$\sum_{e \in B} f_A(e) \geq min\{f_A(B),\ \beta^* f_A(B)\} \tag{11}$$

for an arbitrary real number $\beta^* \geq 1$ and any pair of sets $A, B \subseteq E$.

Proof. Since f is submodular, for every pair of sets $S, T \subseteq E$, we have

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T). \tag{12}$$

Firstly, assume $B \nsubseteq A$ and $|B \backslash A| = l$, denote $B \backslash A = \{e_i\}$ for $i = 1, 2, ..., l$. According to Equation (12), we have

$$\begin{gathered} f(A \cup \{e_1\}) + f(A \cup \{e_2\}) \geq f(A \cup \{e_1, e_2\}) + f(A) \\ f(A \cup \{e_1, e_2\}) + f(A \cup \{e_3\}) \geq f(A \cup \{e_1, e_2, e_3\}) + f(A) \\ \cdots \\ f(A \cup B \backslash \{e_l\}) + f(A \cup \{e_l\}) \geq f(A \cup B) + f(A) \end{gathered}$$

Summing up the above inequations at both ends, conducting transposition and elimination gives

$$\begin{aligned} \sum_{e_i \in B \backslash A} f(A \cup e_i) - l \cdot f(A) &\geq f(A \cup B) - f(A) \\ \Leftrightarrow \qquad \sum_{e \in B} f_A(e) &\geq f_A(B). \end{aligned} \tag{13}$$

Without loss of generality, one can verify that, when $B \subseteq A$, Equation (13) also holds.

In Equation (11), if $f_A(B) \geq 0$, then

$$min\{f_A(B),\ \beta^* f_A(B)\} = f_A(B),$$

Equation (13) implies Equation (11); else if $f_A(B) < 0$, then

$$min\{f_A(B),\ \beta^* f_A(B)\} = \beta^* f_A(B),$$

it also satisfies that $\sum_{e \in B} f_A(e) \geq f_A(B) \geq \beta^* f_A(B)$. □

6.2 Results of Student's *t*-test on Running Time

Regarding running time results on *isAcyclic-n** dataset in Table 3, we run the DAG-Explainer and the optima baseline for 10,000 times and conduct Student's t-test [41] with significance level as 0.05 to validate if the means of running time of our method and the optima baseline are significantly different, p-values are reported in Table 6.

	cyclic class				*acyclic* class				
n	4	5	6	7	3	4	5	6	7
p	10.33	9.132	5.654	0.042	0.935	0.771	1.239	0.022	0.019

Table 6: p-values of Student's t-test on running time.

6.3 Discussion

6.3.1 Transferability of DAG-Explainer. DAG-Explainer is transferable to general dataset. We discuss this from the two perspective: *Effectiveness* perspective: as demonstrated by the new examples from Highschool dataset added to *Section 1. Introduction*, the three spotted issues can appear in general datasets other than molecular structure dataset:

(1) Explanations generated by current methods may not be (highly) recognized by the model. In general, explanation candidates may receive varying scores from the GNN, hence there is always a need to optimize model recognition in global explainability of GNNs. This goal actually corresponds to the input optimization technique for model-level explainability in dealing with not only the graph data [66] but also image and text data [37, 38, 40], which indicates that DGA-Explainer tackles a general challenge for various model-level explanation tasks.
(2) Generated explanations by current methods may not exist in the input data. Though this Out-Of-Distribution (OOD) problem can be avoid by finding subgraphs in input instance using techniques such as masking [31, 65] and graph traversal [68], it happens to most graph generation explainers [66?], because it is impossible to enforce a complete set of constraints to ensure the generated graph perfectly fit in the original input data. On the contrary, DAG-Explainer can always be used to avoid such an OOD problem.
(3) Generated explanations may not be discriminative among different classes. This is a general issue because explanations as subgraphs of instances in the input can naturally appear in any classes, especially when the graph data is complex (e.g., social networks). Hence our method is useful in general datasets because it can produce data-aware (discriminative) explanations for GNNs.

Efficiency perspective: both the algorithm of DAG-Explainer is efficient and the candidate generation is affordable:

(1) The RandomGreedy algorithm is quite efficient as it only queries $O(k|C|)$ times for the values of marginal gain, which is of the same cost as the standard deterministic greedy algorithm.
(2) A natural concern would be the cost of candidate generation for mining the subgraphs. First of all, explanation serves human understanding, thus should be compact (commonly 3-7 nodes [65–67]), mining large subgraphs is avoided. Besides, for graph classification, input graphs are usually small (mean of average number of nodes in each graph is 53.93 among 109 datasets in TU Benchmark [35]); for node or edge -level tasks, explainers only need to consider the L-hop neighborhood of a node or ends of an edge (L is the number of GNN layers), which is also typically small. Regarding candidate generation tool, graph mining is always intensively studied and new solutions are constantly proposed to tackle scaling challenges in real-life application [23, 36, 45].

In conclusion, our method is transferable to general datasets.

6.3.2 Will typical "users" find the explanation helpful? We discuss this question from below two perspectives.

- *Can typical users understand the explanation?* The degree of incomprehension of the explanation mainly depends on the degree of abstraction of the graph itself. Users can easily understand visual datasets such as social networks, but it is relatively difficult to understand graph data that requires domain knowledge, for example, molecular structure graphs. Explainers serve the GNN user, and it is generally believed that the GNN user has the domain knowledge of the application. In our

user study, we hide the semantic meaning information including the dataset name and label name, and simply examine the user's understanding of the graph. Some interviewed study participants said that isAcyclic dataset (abstract topological structure) and highschool dataset (social network) are easy to understand, and it is very simple to think of how the instances are classified (labeled). While MUTAG dataset is quite difficult to understand or guess the meaning of the labels and explanations.

- *Will typical users find the explanation useful?* The usefulness of explanations is to reveal the reasoning logic of the GNN and help users compare the decision-making mechanism of the model with human cognition. For example, chemical experts found that whether a compound is mutagenic depends on whether its molecular structure contains the structure of nitrogen dioxide [16], then if a reliable explainer shows that the GNN makes predictions based on the nitrogen dioxide substructure, users of this GNN (chemical experts or interdisciplinary scholars) can confirm that the working mechanism of the model is reliable. Similarly, epidemiologists can verify the reliability of the model by observing whether the patterns of disease transmission shown by the explanation match the typical spreading patterns in domain knowledge [5, 55]. In particular, global explanation helps users understand general behavior of the model. In our user study, given explanation output by DAG-Explainer, the accuracy of users predicting GNN's predictions on fresh instances is as high as 91% (see Section 3.5.3), which means that real users have already understood the decision-making mechanism of the model. In fact, the interviewed users clearly stated that on isAcyclic dataset, the criterion for the model is if the instance contains circle(s). Although whether typical users really find explanation useful is a question that can only be verified in real scenarios, based on our qualitative analysis and user study results, we believe that explanation output by DAG-Explainer will be helpful.