GNES: Learning to Explain Graph Neural Networks

Yuyang Gao*, Tong Sun†, Rishab Bhatt*, Dazhou Yu*, Sungsoo Hong†, Liang Zhao*
*Emory University
{yuyang.gao, rhbhatt, dazhou.yu, liang.zhao}@emory.edu
†George Mason University
{tsun8, shong31}@gmu.edu

***Abstract*—In recent years, graph neural networks (GNNs) and the research on their explainability are experiencing rapid developments and achieving significant progress. Many methods are proposed to explain the predictions of GNNs, focusing on "how to generate explanations". However, research questions like "whether the GNN explanations are inaccurate", "what if the explanations are inaccurate", and "how to adjust the model to generate more accurate explanations" have not been well explored. To address the above questions, this paper proposes a GNN Explanation Supervision (GNES)[1] framework to adaptively learn how to explain GNNs more correctly. Specifically, our framework jointly optimizes both model prediction and model explanation by enforcing both whole graph regularization and weak supervision on model explanations. For the graph regularization, we propose a unified explanation formulation for both node-level and edge-level explanations by enforcing the consistency between them. The node- and edge-level explanation techniques we propose are also generic and rigorously demonstrated to cover several existing major explainers as special cases. Extensive experiments on five real-world datasets across two application domains demonstrate the effectiveness of the proposed model on improving the reasonability of the explanation while still keep or even improve the backbone GNNs model performance.**

***Index Terms*—Graph Neural Networks, Explainability, Human-in-the-loop**

I. Introduction

As Deep Neural Networks (DNNs) are widely deployed in sensitive application areas, recent years have seen an explosion of research in understanding how DNNs work under the hood (e.g., explainable AI, or XAI) [1], [2] and more importantly, how to improve DNNs using human knowledge [3]. In particular, Graph Neural Networks (GNNs) have been increasingly grabbed attention in several research fields, including computer vision [4], [5], natural language processing [6], medical domain [7], and beyond. Such trend is attributed to the practical implication of graphs data—many real-world data, such as social networks [8], chemical molecules [9], and financial data [10], are represented as graphs.

However, similar to other DNNs' architectures, GNNs also offer only limited transparency, imposing significant challenges in observing when GNNs make successful/unsuccessful predictions [3], [11]. This issue motivates a surge of recent research on GNN explanation techniques, including gradients-based methods, where the gradients are used to indicate the importance of different input features [4], [12]; perturbation-based methods, where an additional optimization step is typically used to find the important input that influences the model output the most with input perturbations [13]–[15]; response-based methods, where the output response signal is backpropagated as an importance score layer by layer until the input space [4], [12], [16]; surrogate-based methods, where the explanation obtained from an interpretable surrogate model that is trained to fit the original prediction is used to explain the original model [17]–[19]; and global explanation methods, where graph patterns are generated to maximize the predicted probability for a certain class and use such graph patterns to explain the class [20].

Despite the recent fast progress on GNN explanation techniques, the existing research body focuses on "how to generate GNN explanations" instead of "whether the GNN explanations are inaccurate", "what if the explanations are inaccurate", and "how to adjust the model to generate more accurate explanations". Answering the above questions is highly beneficial to the model developers and the users of GNN explanation techniques, but are also extremely difficult due to several challenges: **1) Lack of an automatic learning framework for identifying and adjusting unreasonable explanations on GNNs.** Although there are plenty of existing works on GNN explanations, they are not able to ensure the correctness of explanations, not able to identify the incorrect explanations, nor able to adjust the unreasonable explanations. The technique that can enable this has not been well explored yet and is technically challenging due to the additional involvement of another backpropagation originated from explanation error. **2) Difficulty in aligning the node and edge explanations.** Existing GNN explanation works usually focus on either node and edge explanation while the interplay and consistency between the explanations of nodes and edges are extremely challenging to maintain and jointly adjusted. **3) Difficulty in jointly improving model performance and explainability with limited explanation supervision.** Due to the high cost for human annotation, it can be impractical to assume the full accessibility to the human explanation label during model training. Thus designing an effective framework that can best leverage a partially labeled dataset is on-demand yet challenging.

To address the above challenges, beyond merely generating GNN explanations, this paper focuses on a generic GNN explanation supervision framework for correcting the unreasonable explanations and learning how to explain GNNs correctly. Specifically, we first propose a unified explanation method

[1]Code available at: https://github.com/YuyangGao/GNES.

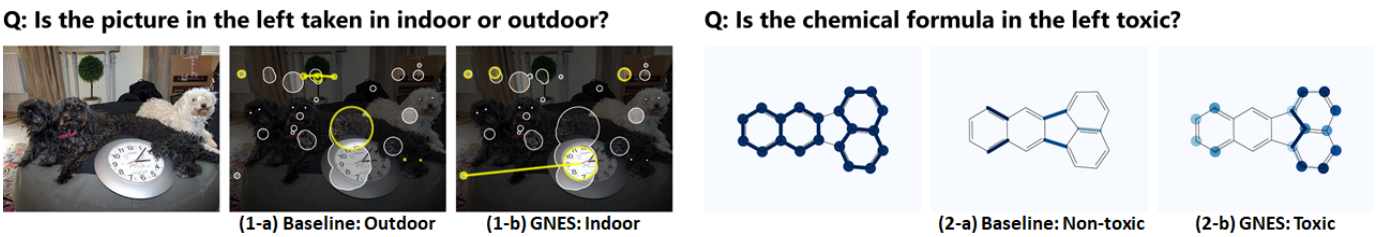

Fig. 1. Cases for adjusting model explanation to improve Graph Neural Networks (GNNs). Scene Graph (left three): From the left, an input image, explanation before adjustment (1-a, inaccurate), and explanation after the adjustment (1-b, accurate). Note that the model explanation has been shifted from puppy eyes and back, rods, and an artificial tree to curtains, a clock, and a rug. Molecular formula (right three): From the left, an input formula, explanation before the adjustment (2-a, inaccurate), and explanation after the adjustment (2-b, accurate). Reactivity for this molecule is mostly affected by benzene ring sub-components in the overall molecular structure. 2-b highlights the main benzene rings of the molecule more effectively than 2-a.

for GNNs that can generate node and edge explanations with consistency regularization among them. The generality of the proposed method over existing node-explanation methods is rigorously demonstrated. Finally, we develop a learning objective that jointly optimizes model prediction and explanation with weak supervision from human explanation annotations.

Specifically, the main contributions of our study are as follows:

1) **Developing a generic framework for adaptively learning how to explain GNNs with weak explanation supervision.** We present a new learning objective for joint optimization among the model prediction loss, the explanation loss, and the graph regularization loss on regulating the model explanation. In addition, our framework can treat the explanation loss as an optional term and thus work effectively in scenarios where the human annotation on explanation is limited.
2) **Developing a unified graph-based explanation framework for calculating both node-level and edge-level explanation of GNNs.** We proposed a unified framework for both node-level and edge-level explanations that is suitable for explanation supervision and generalizable to the existing differentiable explanation methods.
3) **Proposing a model that can regularize both the node-level and edge-level explanations to form a better graph-level explanation.** We propose to apply novel explanation regularizations (i.e., explanation consistency and sparsity) onto the model-generated explanation to inject general graph principles and prior knowledge about the explanation that enhance the quality and consistency among the multiple levels of explanations.
4) **Conducting comprehensive experiments to validate the effectiveness of the proposed model.** Extensive experiments on five real-world datasets in two domains, chemical (molecular graphs) and vision (scene graphs), demonstrate that the proposed models improved the backbone GNN model both in terms of prediction power and explainability across different application domains. In addition, qualitative analyses, including case studies and user studies of the model explanation, are provided to demonstrate the effectiveness of the proposed framework.

II. Related Work

Our work draws inspiration from the research fields of graph neural network explanations that provide the model generated explanations, and explanation supervision on DNNs which enables the design of pipelines for the human-in-the-loop adjustment on the DNNs based on their explanations.

A. Graph Neural Networks Explanations

Most of the existing GNN explanation methods are instance-level methods, where the methods explain the models by identifying important input features for its prediction [21]. The first category is gradients-based methods, where the gradients are used to indicate the importance of different input features. Existing methods are SA [12], Guided BP [12], CAM [4], and GradCAM [4]. The second category is perturbation-based methods, where an additional optimization step is typically used to find the important input that influences the model output the most with input perturbations. Existing methods are GNNExplainer [13], PGExplainer [14], GraphMask [15]. The third category is the response-based method, where the output response signal is backpropagated as an importance score layer by layer until the input space. Existing methods in this category including LRP [12], Excitation BP [4] and GNN-LRP [16]. The last category is surrogate-based methods, where the explanation obtained from an interpretable surrogate model that is trained to fit the original prediction is used to explain the original model. The surrogate methods include GraphLime [17], RelEx [18], and PGM-Explainer [19]. Besides instance-level explanation methods, very recently, the global explanation of the GNN model has also been explored by XGNN [20]. Please see Yuan et. al. [21] for a survey of explainability in Graph Neural Networks.

Even though there are plenty of existing explanation methods for GNNs, most of the methods above can not be applied to explanation supervision mechanism, as the goal is to apply supervision on the generated explanation such that the backbone GNN model itself can be fine-tuned accordingly to generate better explanations as well as keep or even improve the model performance. To enable this fine-tuning process over the explanation, the explanation itself needs to be differentiable to the backbone GNN model's parameters. In other words, only the explanation that is directly calculated from the computational

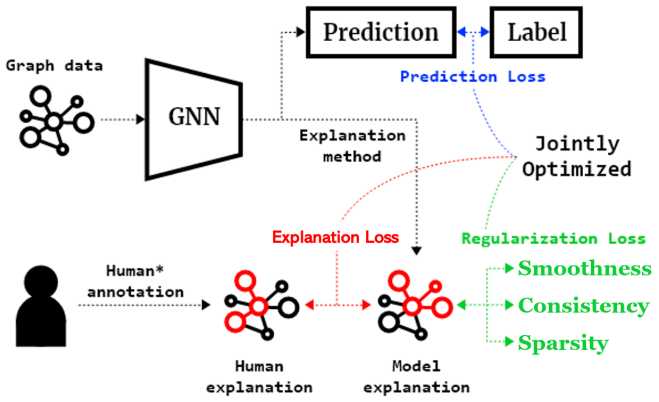

Fig. 2. The proposed GNN Explanation Supervision (GNES) framework that jointly optimized the GNN models based on 1) a prediction loss, 2) an explanation loss on the human annotation and model explanation, and 3) a graph regularization loss to inject high-level principles of the graph-structured explanation. Notice that we only assume limited accessibility to the human annotation for only a small set of samples (10% in our experiments).

pipeline (such as gradients-based and response-based methods) can be used to apply this additional explanation supervision to fine-tune the backbone GNN models explanation. The perturbation-based and surrogate-based methods all require additional optimization steps to obtain the explanation and thus are unable to be end-to-end trained with the explanation supervision on the backbone GNNs.

B. Explanation Supervision on DNNs

The potential of using *explanation*–methods devised for understanding which sub-parts in an instance are important for making a prediction–in improving DNNs has been studied in many domains across different applications. In fact, explanation supervision has been widely studied on image data by the computer vision community [22]–[28]. Linsey et al. [22] have demonstrated that the benefit of using stronger supervisory signals by teaching networks where to attend, which looks similar to the proposed approach. Moreover, Mitsuhara et al. [23] have proposed a post hoc fine-tuning strategy where an end-user is asked to manually edit the model's explanation to interactively adjust its output. Such edited explanations are then used as ground-truth explanations (from humans) to further fine-tune the model. In addition, several works in the Visual Question Answering (VQA) domain have proposed to use explanation supervision to obtain improved explanation on both the text data and the image data [24], [26]–[28]. Besides image data, the explanation supervision has also been studied on other data types, such as texts [29], [30], attributed data [31], and more. However, to our best knowledge, explanation supervision on graph-structured data with graph neural networks has not been explored before, and we are the first to propose a framework to handle this open research problem.

III. Model

In this section, we first introduce the proposed GNES framework that boosts the model explainability via explanation supervision and the novel explanation regularizations (i.e., explanation consistency and sparsity) that enhance the quality and consistency among the multiple levels of explanations. We then move on to introduce the proposed unified formulations for both node-level and edge-level explanation that are suitable for explanation supervision.

Problem formulation: Let $\mathcal{G} = (X, A)$ denote a attributed graph with N nodes be defined with its node attributes $X \in \mathbb{R}^{N \times d_{in}}$ and its adjacency matrix $A \in \mathbb{R}^{N \times N}$ (weighted or binary), where d_{in} denotes the dimension of input feature. Let y be the class label for graph $\mathcal{G}$, the general goal for a GNN model is to learn the mapping function f for each graph $\mathcal{G}$ to its corresponding label, $f : \mathcal{G} \rightarrow y$.

Following the definition of Graph Convolutional Networks (GCN) [32], a graph convolutional layer can be defined as:

$$F^{(l)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}F^{(l-1)}W^{(l)}) \tag{1}$$

Where $F^{(l)}$ denotes the activations at layer l, and $F^{(0)} = X$; $\tilde{A} = A + I_N$ is the adjacency matrix with added self connections where $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix; $\tilde{D}$ is the degree matrix of $\tilde{A}$, where $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$; The trainable weight matrix for layer l is denoted as $W^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l+1)}}$; $\sigma(\cdot)$ is the element-wise nonlinear activation function.

In addition, to deal with variable size graphs in the dataset where the number of nodes can be different among graph samples, we adopt a similar design as in [4] to our backbone GNN model using several layers of graph convolutional layers followed by a global average pooling (GAP) layer over the graph nodes (e.g., atoms for the molecular graph and objects for the scene graph).

A. GNES Framework

The general goal for the GNES framework is to boost the model explainability via explanation supervision such that the model performance could also benefit from assigning more importance to the right features. Specifically, for graph data, the explanation supervision can be done in two main ways: 1) by applying some high-level graph-structured rules to the explanation, and 2) by adding human annotation samples as additional guidance. Thus, we present the learning objective of the GNES framework to be a joint optimization among the model prediction loss, the explanation loss, and graph regularizations on regulating the model explanation, as shown in Figure 2. Concretely, we propose the objective function as:

$$\min \ \mathcal{L}_{\text{Pred}} + \underbrace{\mathcal{L}_{\text{Att}}(\langle M, M' \rangle, \langle E, E' \rangle)}_{\text{explanation loss}} + \underbrace{\Omega(A, M, E)}_{\text{regularization}} \tag{2}$$

where $M \in \mathbb{R}^{N \times 1}$ and $E \in \mathbb{R}^{N \times N}$ denote the model generated node-level and edge-level explanations using a given explanation method. and M', E' are the corresponding ground-truth explanations marked by the human annotators. $\mathcal{L}_{\text{Pred}}$ is the typically prediction loss (such as the cross-entropy loss) on the training set. The proposed explanation loss $\mathcal{L}_{\text{Att}}$ measures

the discrepancies between model and human explanations on both node-level and edge-level, as:

$$\mathcal{L}_{\text{Att}}(\langle M, M' \rangle, \langle E, E' \rangle) = \underbrace{\alpha_n \text{dist}(M, M')}_{\text{node-level loss}} + \underbrace{\alpha_e \text{dist}(E, E')}_{\text{edge-level loss}} \tag{3}$$

Where α_n and α_e are the scale factors for balancing node-level and edge-level loss; the function $\text{dist}(x, y)$ measures the mean element-wise distance between the inputs x and y, a common choose can be absolute difference or squared difference. In practice, we found that the absolute difference is more robust to the labeling noise from the annotator.

However, due to the high cost of human annotation on the explanations, obtaining the human explanations for the whole dataset can be prohibitive in practice. To deal with this issue, we propose to only apply the explanation loss to the samples that have the ground-truth labels for the human explanations, and apply the high-level graph rules to regulate the model explanation for each sample even if the human annotation is unavailable. Specifically, we propose a novel explanation consistency regularization term that regulates the node and edge explanation simultaneously so that the model is more likely to generate a globally consistent and smooth explanation over nodes and edges. Besides, we use sparsity regularization to regulate the model to only focus on a few important nodes and edges for the explanations. Thus, we propose the following graph regularizations to obtain more reasonable model explanations:

$$\Omega(A, M, E) = \underbrace{\beta \Omega_c(A, M, E)}_{\text{explanation consistency}} + \underbrace{\gamma \Omega_s(M, E)}_{\text{sparsity}} \tag{4}$$

Where β is the scaling factor for the explanation consistency between node and edge explanations, γ is the scaling factor for the sparsity constraints on both node and edge explanations. Concretely, each regularization and its desirable effects for regulating the graph explanation is described in more detail below:

Explanation consistency regularization. The node explanation and edge explanation are not independent, but rather highly correlated with each other. One natural assumption about the node explanation smoothness is that the adjacent nodes should share similar importance. However, this assumption can be too strong and sometimes lead to over-smoothing of the node explanation and tend to yield indistinguishable patterns for the explanation. In addition, it ignored the connection between the node and edge explanations, which can be a crucial factor for the explanation model to generate a global consistent explanation.

Here, we propose to take one step further regarding the smoothness assumption about the explanation by considering both node and edge explanations and making them more consistent with each other. Concretely, instead of treating all pairs of adjacent nodes equally important when enforcing the smoothness constraint, we propose to weight them by the corresponding edge importance such that the explanation consistency is better enforced on those nodes and edges that are deemed important. Mathematically, the explanation consistency can be measured by:

$$\Omega_c(A, M, E) = \frac{1}{2N^2} \sum_{i,j} E_{i,j} A_{i,j} \|M_i - M_j\|^2 \tag{5}$$

The above regularization can be interpreted as follows: given a pair of nodes i and j that is adjacent (i.e., $A_{i,j} = 1$), if the edge that connects the two nodes is important (i.e., $E_{i,j}$ is high), then the nodes it connects also tend to be consistent.

Sparsity regularization. As sparsity is a common practice for the model explanation, we apply the ℓ_1 norm to regulate both the node-level and the edge-level explanations, as:

$$\Omega_s(M, E) = \frac{1}{N} \|M\|_1 + \frac{1}{N^2} \|E\|_1 \tag{6}$$

Overall, the benefits of applying the proposed regularization terms are threefold. First, the regularization terms do not rely on the specific human labels on the explanation, which can be very limited and hard to acquire in practice. Thus they can be very crucial in the scenarios where the explanation labels are scarce. Second, since the explanation for the node and edge can be highly relevant, the proposed explanation consistency regularization can be critical for enforcing the model to generate more reasonable and consistent results that better align with the human explanation. Lastly, our overall framework is very flexible such that the regularization terms are not affected by changing the specification of the node and edge explanation formulation in Equation (7) and Equation (10), respectively, making the proposed framework easily applicable to give explanation and apply explanation supervision on any downstream applications with little to no overhead.

B. Node Explanation Formulation for Explanation Supervision

Although the node-level explanation is the most studied topic in the instance-based graph explanation domain, there are still several challenges to apply the node explanation supervision: First, most existing methods do not apply to the explanation supervision as the generated explanations are no longer differentiable to the backbone GNN model's parameters. Moreover, there is no unified formulation for the node-level explanation supervision.

To handle those challenges, we propose the first unified node explanation formulation for node-level explanation supervision. Concretely, we first identify that the gradient and the response/activation can be the major information that can produce the model-generated explanation that remains differentiable to the backbone GNN model's parameters so that the explanation supervision can be performed to affect the model during training. We then propose to integrate both aspects to form a general formulation for the node explanation. Mathematically, given the output y_c on class c, the explanation for node n at layer l can be computed as:

$$M_n^{(l)} = \|\text{ReLU}(g(\frac{\partial y_c}{\partial F_n^{(l)}}) \cdot h(F_n^{(l)}))\| \tag{7}$$

Where $\frac{\partial y_c}{\partial F_n^{(l)}}$ represents the gradient of the features of node n at layer l given class c, and $F_n^{(l)}$ denotes the node activation at layer l, $g(\cdot)$ and $h(\cdot)$ are the functions that can be further defined to cover more complicated computation over the gradient as well as the activation, respectively.

The formulation above is a generic framework that covers as special cases major existing works where the gradient of the node features and the activation of the node are used to calculate the node explanation or the node importance, as shown in the following theorem.

Theorem 1 (Generality of Equation (7))**.** *The proposed generic node-level explanation formulation in Equation* (7) *covers a broad range of important existing works on node-level explanation as special cases with specification of* $h(\cdot)$ *and* $g(\cdot)$*, such as the gradient-based saliency maps (GRAD), GradCAM [4], [33], Layer-wise Relevance Propagation (LRP) [12], [34], and Excitation Backpropagation (EB) [4], [35].*

Proof. The specification for the function $g(\cdot)$ and function $h(\cdot)$ for each existing methods are listed in detail below:

Simple gradient-based saliency maps (GRAD): For simple GRAD, only the function $g(\cdot)$ is active, and it is simply the identity function, i.e. $g(\frac{\partial y_c}{\partial F_n^{(l)}}) = \frac{\partial y_c}{\partial F_n^{(l)}}$; the function $h(\cdot)$ will trivially return 1 (i.e. $h(F_n^{(l)}) = 1$) as the activation is not used in simple GRAD situation.

GradCAM: For the GradCAM [4], [33], since it uses both gradient information and node activation, both functions will be non-trivial. Specifically, the function $g(\cdot)$ can be defined as $g(\frac{\partial y_c}{\partial F_n^{(l)}}) = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial y_c}{\partial F_n^{(l)}}$; and the function $h(\cdot)$ is the identity function (i.e. $h(F_n^{(l)}) = F_n^{(l)}$).

Layer-wise Relevance Propagation (LRP): For LRP [12], [34], gradient information is ignored and only the node activation is used. Concretely, the function $g(\cdot)$ will trivially return 1 (i.e. $g(\frac{\partial y_c}{\partial F_n^{(l)}}) = 1$); the function $h(F_n^{(l)}) = \frac{1}{d_l}\sum_{k=1}^{d_l}\hat{h}(F_{k,n}^{(l)})$ where $\hat{h}(F_{k,n}^{(l)})$ can be calculated via a relevance propagation as shown below.

For notational simplicity, we first decompose a graph convolutional operator into:

$$\begin{cases}\hat{F}_{k,n}^{(l)} = \sum_m V_{n,m}F_{k,m}^{(l)} \\ F_{k',n}^{(l+1)} = \sigma(\sum_{k'}\hat{F}_{k,n}^{(l)}W_{k,k'}^{(l)}),\end{cases} \tag{8}$$

where $V = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is the normalized graph Laplacian; the first equation is a local averaging of nodes, and the second equation is a fixed perceptron applied to each node (analogous to one-by-one convolutions in CNNs).

To capture both activatory and inhibitory parts of the forward pass, the $\alpha\beta-$rule is applied in RP, and the corresponding backward passes for these two functions can be defined as:

$$\begin{cases}\hat{h}(F_{k,n}^{(l)}) = \sum_m \frac{V_{n,m}F_{k,n}^{(l)}}{\sum_n V_{n,m}F_{k,m}^{(l)}}\hat{h}(\hat{F}_{k,m}^{(l)}) \\ \hat{h}(\hat{F}_{k,n}^{(l)}) = \sum_{k'}(\alpha\frac{\hat{F}_{k,n}^{(l)}W_{k,k'}^{(l)+}}{\sum_k \hat{F}_{k,n}^{(l)}W_{k,k'}^{(l)+}} + \beta\frac{\hat{F}_{k,n}^{(l)}W_{k,k'}^{(l)-}}{\sum_k \hat{F}_{k,n}^{(l)}W_{k,k'}^{(l)-}})\hat{h}(F_{k',n}^{(l+1)}),\end{cases} \tag{9}$$

where $W_{k,k'}^{(l)+} = max(0, W_{k,k'}^{(l)})$, and $W_{k,k'}^{(l)-} = min(0, W_{k,k'}^{(l)})$, and typically $\alpha + \beta = 1$ in order to uphold conservativity of relevance between layers.

Excitation Backpropagation (EB): For EB [4], [35], it follows the same setting as in LRP, except the parameter $\alpha = 1, \beta = 0$ in Equation (9), which only focus on the activatory or excitation part of the forward pass when calculating $h(F_n^{(l)})$. □

Here we have demonstrated the broad coverage of the proposed node-level explanation formulation for enabling the unified node explanation supervision. Other existing gradient-based methods and response-based methods can also be easily derived and fitted into this framework by specifying the functions $g(\cdot)$ and $h(\cdot)$ respectively.

C. Edge Explanation Formulation for Explanation Supervision

Besides node-level explanation, the edge-level explanation can also be very crucial in many applications to highlight the important relationships between nodes. Unfortunately, most existing methods that focus on edge-level or subgraph-level explanations such as GNNExplainer [13], PGExplainer [14], and GraphMask [15] can not be used under the explanation supervision framework, as those explanations typically require additional objectives and optimization steps, making it not differentiable to the backbone model's parameters. Existing gradients-based methods and response-based methods typically focused only on node-level explanation, while little to no work has explored the edge-level explanation. Very recently, GNN-LRP [16] explored the higher-order edge-level explanation based on LRP. However, the multiple levels/orders of explanations on the edges are generally very hard to interpret and align with human annotations.

To enable edge-level explanation supervision, we propose the first unified edge-level explanation formulation following a similar path from node-level explanation. Concretely, using the chain rule, we identify that the gradient of the adjacency matrix, as well as the response/activation of the pairs of nodes that are associated with the edges can be the major information that can produce the model generated explanation that remains differentiable to the backbone GNN model's parameters. We then propose to integrate both aspects together to form a general formulation for the edge-level explanation. Concretely, given the output y_c on class c, the edge explanation between node n and node m at layer l can be computed as:

$$E_{n,m}^{(l)} = \|\text{ReLU}(g(\frac{\partial y_c}{\partial F^{(l)}}\cdot\frac{\partial F^{(l)}}{\partial A_{n,m}})\cdot h(F_n^{(l)}, F_m^{(l)}))\| \tag{10}$$

Where $\frac{\partial y_c}{\partial F^{(l)}}\cdot\frac{\partial F^{(l)}}{\partial A_{n,m}}$ represents the gradient of the edge that connects node n and node m at layer l given class c; $F_n^{(l)}$ and $F_m^{(l)}$ denote the activation of node n and node m at layer l, respectively; again $g(\cdot)$ and $h(\cdot)$ are the two functions that can be further defined to cover more complicated computation over the gradient as well as the activation, respectively.

Again, the formulation above should be able to generalize to most cases where the gradient of the edge and the activation of the pair of nodes are used to calculate the edge explanation. Although there is not yet any existing work that falls under this umbrella, we propose two possible specifications of the edge-level explanation from the above formulation as shown below.

Gradient-based: This can be seen as the extension from GRAD to edge-level explanation. Specifically, only the gradient information is used, as $g(\frac{\partial y_c}{\partial F^{(l)}} \cdot \frac{\partial F^{(l)}}{\partial A_{n,m}}) = \frac{\partial y_c}{\partial F^{(l)}} \cdot \frac{\partial F^{(l)}}{\partial A_{n,m}}$, and the node activation information is ignored, i.e. $h(F_n^{(l)}, F_m^{(l)}) = 1$.

Response-based: This can be seen as the extension from LRP to edge-level explanation. In this specification, the gradient information is ignored, i.e. $g(\cdot) = 1$, and the function $h(\cdot)$ is defined as:

$$h(F_n^{(l)}, F_m^{(l)}) = V_{n,m} \sum_{k=1}^{d_l} (\hat{h}(\hat{F}_{k,m}^{(l)}) + \hat{h}(\hat{F}_{k,n}^{(l)})) \tag{11}$$

where $\hat{h}(\hat{F}_{k,n}^{(l)})$ can be computed by Equation (8) and Equation (9).

IV. Experiments

We test our GNES framework on two application domains, visual scene graphs and molecules. We first describe the detailed settings for the experiments and then present the quantitative studies on both model prediction as well as the explanation. In addition, we include several qualitative studies, including case studies and user studies, to make a qualitative assessment of how the proposed model has enhanced the explainability of the GNNs.

A. Experimental Settings

Molecular Graphs: We study three binary classification molecular datasets[2], BBBP, BACE, and task NR-ER from TOX21 [36], where the general goal for the classification task is identifying functional groups on organic molecules for biological molecular properties. Each dataset contains binary classifications of small organic molecules as determined by the experiment. The details of each dataset are listed below:

1) *BBBP*: The Blood-brain barrier penetration (BBBP) dataset comes from a recent study [37] on the modeling and prediction of barrier permeability. As a membrane separating circulating blood and brain extracellular fluid, the blood-brain barrier blocks most drugs, hormones, and neurotransmitters. Thus penetration of the barrier forms a long-standing issue in the development of drugs targeting the central nervous system. This dataset includes binary labels for 2053 compounds (graphs) on their permeability properties.
2) *BACE*: The BACE dataset provides quantitative (IC50) and qualitative (binary label) binding results for a set of inhibitors of human b-secretase 1 (BACE-1) [38]. This dataset contains a collection of 1522 compounds (graphs) with their 2D structures and binary labels.
3) *TOX21*: The "Toxicology in the 21st Century" (TOX21) initiative created a public database measuring the toxicity of compounds. The original dataset contains qualitative toxicity measurements for 8014 compounds (graphs) on 12 different tasks, here we selected the NR-ER task, which is concerned with the activation of the estrogen receptor [39].

Following the existing works on molecule classification [36], we split the dataset into train/validation/test with an 80/10/10 split ratio. In addition, we use the "scaffold" split algorithm for BBBP and BACE, where structurally similar molecules are partitioned in the same split. For TOX21, the random split is used.

Scene Graphs: We obtain the scene graphs from the Visual Genome dataset[3] [40]. The Visual Genome dataset consists of images and a corresponding scene graph where the nodes are objects in the scene and edges are relationships between objects. Objects and relationships are of many types and the data is collected from free-text responses obtained from crowdsourced workers. Objects have an associated region of the image, defined by a bounding box. Following the previous work by [4], we construct two binary classification tasks: country vs. urban, and indoor vs. outdoor. The data samples for the two tasks are selected based on a set of pre-defined keywords which are used to query the Visual Genome data for matches in any attribute of an image. Specifically, the keywords used to define each class are listed below:

- *country*: countryside, farm, rural, cow, crops, sheep
- *urban*: urban, city, downtown, building
- *indoor*: indoor, room, office, bedroom, bathroom
- *outdoor*: outdoor, nature, outside

Notice that the keywords are non-comprehensive and the generated datasets are just for the purpose of studying the explanation on graphs. We balanced the sample size for each class by randomly selecting 1000 samples out of the image pools from the Keyword match. Again, we randomly split the dataset into train/validation/test with an 80/10/10 split ratio.

To convert the visual genome data to the graph input data, we treat each object as a unique node in the graph and the edge will be the corresponding relationship between a pair of objects. For the node feature for each object, we use a pre-trained InceptionV3 [41] network to extract the deep features from the image region defined by the bounding box associated with each object. The feature dimension for all visual genome nodes is of size $d = 2048$.

Evaluation Metrics: We evaluate the model in terms of performance as well as in terms of explainability. Specifically, for model performance assessment, we use accuracy (ACC) and Area Under the Curve (AUC) scores to measure the prediction power of the GNNs on the prediction tasks for sense graph datasets, and only AUC scores for molecular graph datasets as the sample size can be imbalanced. Besides,

[2]Available online at: http://moleculenet.ai/datasets-1

[3]Available online at: https://visualgenome.org/

we leverage the human-labeled explanation on the test set to quantitatively assess the goodness of the model explanation. Specifically, for both node-level and edge-level explanations, we treat the human explanation as the gold standard, and compute the distance between human and model explanation via Mean Square Error (MSE) and Mean Absolute Error (MAE). To match with human annotation, both node-level and edge-level explanations are normalized in the range of $(0, 1]$ by dividing the corresponding max values.

Comparison Methods: Since there is no existing work on explanation supervision on GNNs and graph data, we demonstrate the effectiveness of our model in the following two aspects: First, we compare the explanation obtained by the proposed model with the explanation generated by the existing explanation methods on the backbone GNN as baselines to assess the improvement in terms of the model explainability. Concretely, we compare the explanation generated by Grad-CAM as the gradient propagation-based explanation, and EB as the relevance propagation-based explanation on a GNN with the same architecture as used in the proposed framework. Next, we conduct the ablation study of the proposed GNES framework to assess the effect of each proposed component. Specifically, we studied the following variations of GNES:

- **GNES**$^{-human}_{+reg}$: The variation where we ablate the human annotation and use graph regularization only to regulate the model explanation.
- **GNES**$^{+human}_{-reg}$ The variation where we ablate the regularization and only use the human annotation to supervise the model explanation.
- **GNES**$^{+human}_{+reg}$ The complete pipeline where we leverage both human annotation as well as graph regularization to supervise the model explanation.

Implementation details. Following the previous work on the explainability method on GNNs, we used a 3 layer GCN as our backbone GNN model. More specifically, the hidden dimension size for the three graph convolutional layers are of size 512, 256, and 128, respectively, followed by a global average pooling (GAP) layer, and a softmax classifier. Models were trained for 100 epochs using the ADAM optimizer [42] with a learning rate of 0.001. The models were implemented in Keras with Tensorflow backend [43] and the newly proposed explanation loss and regularization loss were implemented by the custom loss function in Keras. We studied the node and edge explanation at the last GCN layer (i.e. $l = 3$). The node-level explanation for the GNES was specified following the GradCAM formulation, and the edge-level explanation is specified following the gradient-based formulation accordingly. The scale factors α_n and α_e for balancing node-level and edge-level loss in (3) were set to 1 by default; and the scale factors β and γ for the regularization in Equation (4) were grid researched via the AUC score on the validation set. Notice that for the human explanation annotation, we only used 10% of the human annotation for the training data for every dataset to simulate a more piratical situation where we only have partial human label data available. The samples in the test set are all labeled for evaluation purposes.

B. Performance

Table I shows the model performance and model generated explanation quality for the three molecular datasets. The results are obtained from 5 individual runs for every setting. The best results for each dataset are highlighted with boldface font and the second bests are underlined. For the models with human annotation (i.e., GNES$^{+human}_{-reg}$ and GNES$^{+human}_{+reg}$), we only assume 10% of the training sample has the explanation label for the node-level and edge-level explanations while all the remaining are treated as unlabeled samples. In general, our proposed GNES model variations outperformed the explanations from the backbone GNN model in terms of both prediction power as well as explainability on all 3 molecular datasets. More specifically, the ablation study of the model variations suggested that both the human annotation and graph regularization can have positive effects in different scenarios, and the full GNES model (i.e., GNES$^{+human}_{+reg}$) achieved the best performance, out-performing baseline GNN by 1% - 4% on AUC score. In addition, the full GNES model also significantly enhanced the explainability of the backbone GNNs by a great margin, both on node-level explanation (outperformed baselines by 20% - 37% and 6% - 16% on MSE and MAE, respectively) and on edge-level explanation (outperformed baselines by 9% - 36% and 1% - 13% on MSE and MAE, respectively). Those results demonstrated the effectiveness of the proposed framework not only on enhancing the model to pay correct explanation to the critical nodes and edges, but also consequently improved the model performance and prediction power on the prediction tasks.

Next, we examine the model performance and explanation quality on the two scene graph tasks. As shown in Table II, all the setting are the same as in molecular graph tasks, except this time we also studied the accuracy (ACC) as the sample size for each class are balanced. We continue to see that the proposed GNES model achieved the best performance in terms of both ACC and AUC, and largely improved the GNN model's explainability on both node-level and edge-level explanations. Specifically, we observed a 5%-22% improvement on node-level explanation, and a 7% - 30% improvement on edge-level explanation. All the above results have further demonstrated the general effectiveness of the proposed framework across different application domains.

C. Qualitative Analysis of the Explanation

1) Case Studies: Here we provide some case studies about the model explanation for both molecular graphs and scene graphs, as illustrated in Figure 3.

Molecular graphs: As shown in the bottom 3 rows of Figure 3, nodes and edges for molecular graphs were marked as important if they presented unique characteristics of significant reactivity or stability. For reactivity, special importance and annotations were provided if the atoms (nodes) and bonds (edges) were included in functional groups, highly polar

TABLE I

THE PERFORMANCE AND MODEL GENERATED EXPLANATION EVALUATION AMONG THE PROPOSED MODELS AND THE BASELINES ON 3 MOLECULAR GRAPH DATASETS. THE RESULTS ARE OBTAINED FROM 5 INDIVIDUAL RUNS FOR EVERY SETTING. THE BEST RESULTS FOR EACH DATASET ARE HIGHLIGHTED WITH BOLDFACE FONT AND THE SECOND BESTS ARE UNDERLINED.

Dataset	Exp_Method	AUC	Node MSE	Node MAE	Edge MSE	Edge MAE
BBBP	EB	0.659 ± 0.011	0.572 ± 0.010	0.590 ± 0.009	0.050 ± 0.003	0.051 ± 0.002
	GradCAM	0.659 ± 0.011	0.460 ± 0.008	0.545 ± 0.004	0.042 ± 0.001	0.050 ± 0.001
	$\text{GNES}^{-human}_{+reg}$	0.662 ± 0.012	0.375 ± 0.018	0.514 ± 0.008	**0.029 ± 0.001**	**0.047 ± 0.001**
	$\text{GNES}^{+human}_{-reg}$	0.665 ± 0.009	0.449 ± 0.005	0.540 ± 0.006	0.041 ± 0.001	0.049 ± 0.001
	$\text{GNES}^{+human}_{+reg}$	**0.676 ± 0.007**	**0.358 ± 0.007**	**0.504 ± 0.007**	0.032 ± 0.001	0.048 ± 0.001
BACE	EB	0.703 ± 0.030	0.517 ± 0.008	0.548 ± 0.003	0.033 ± 0.001	**0.035 ± 0.000**
	GradCAM	0.703 ± 0.030	0.483 ± 0.006	0.544 ± 0.002	0.032 ± 0.000	0.036 ± 0.000
	$\text{GNES}^{-human}_{+reg}$	0.729 ± 0.009	0.427 ± 0.004	0.525 ± 0.002	0.026 ± 0.000	0.036 ± 0.000
	$\text{GNES}^{+human}_{-reg}$	0.732 ± 0.020	0.421 ± 0.004	0.522 ± 0.003	0.024 ± 0.001	**0.035 ± 0.000**
	$\text{GNES}^{+human}_{+reg}$	**0.733 ± 0.010**	**0.391 ± 0.005**	**0.519 ± 0.003**	**0.023 ± 0.001**	**0.035 ± 0.000**
TOX21	EB	0.788 ± 0.010	0.560 ± 0.028	0.622 ± 0.007	0.081 ± 0.006	0.091 ± 0.004
	GradCAM	0.788 ± 0.010	0.466 ± 0.018	0.566 ± 0.005	0.071 ± 0.002	0.084 ± 0.003
	$\text{GNES}^{-human}_{+reg}$	0.789 ± 0.020	0.460 ± 0.024	0.562 ± 0.004	0.068 ± 0.004	0.081 ± 0.001
	$\text{GNES}^{+human}_{-reg}$	0.789 ± 0.008	0.393 ± 0.009	0.537 ± 0.008	**0.065 ± 0.003**	0.083 ± 0.001
	$\text{GNES}^{+human}_{+reg}$	**0.794 ± 0.012**	**0.392 ± 0.008**	**0.523 ± 0.004**	**0.065 ± 0.002**	**0.079 ± 0.001**

TABLE II

THE PERFORMANCE AND MODEL GENERATED EXPLANATION EVALUATION AMONG THE PROPOSED MODELS AND THE BASELINES ON 2 SCENE GRAPH CLASSIFICATION TASKS. THE RESULTS ARE OBTAINED FROM 5 INDIVIDUAL RUNS FOR EVERY SETTING. THE BEST RESULTS FOR EACH TASK ARE HIGHLIGHTED WITH BOLDFACE FONT AND THE SECOND BESTS ARE UNDERLINED.

Dataset	Exp_Method	ACC	AUC	Node MSE	Node MAE	Edge MSE	Edge MAE
Indoor vs. Outdoor	EB	0.922 ± 0.009	**0.965 ± 0.001**	0.304 ± 0.002	0.361 ± 0.001	0.013 ± 0.000	0.016 ± 0.000
	GradCAM	0.922 ± 0.009	**0.965 ± 0.001**	0.280 ± 0.002	0.439 ± 0.006	0.010 ± 0.000	0.016 ± 0.000
	$\text{GNES}^{-human}_{+reg}$	0.927 ± 0.003	0.964 ± 0.002	0.274 ± 0.004	0.420 ± 0.007	0.010 ± 0.000	0.016 ± 0.000
	$\text{GNES}^{+human}_{-reg}$	0.925 ± 0.004	**0.965 ± 0.001**	0.270 ± 0.002	0.419 ± 0.005	0.010 ± 0.000	0.015 ± 0.000
	$\text{GNES}^{+human}_{+reg}$	**0.930 ± 0.005**	**0.965 ± 0.002**	**0.267 ± 0.003**	**0.406 ± 0.005**	**0.009 ± 0.000**	**0.014 ± 0.000**
Country vs. Urban	EB	0.991 ± 0.000	0.965 ± 0.003	0.271 ± 0.006	0.373 ± 0.008	0.015 ± 0.000	0.022 ± 0.000
	GradCAM	0.991 ± 0.000	0.965 ± 0.003	0.257 ± 0.006	0.433 ± 0.008	0.016 ± 0.000	0.023 ± 0.000
	$\text{GNES}^{-human}_{+reg}$	0.992 ± 0.000	0.965 ± 0.004	0.243 ± 0.001	0.414 ± 0.003	0.015 ± 0.000	0.022 ± 0.001
	$\text{GNES}^{+human}_{-reg}$	0.993 ± 0.000	0.969 ± 0.004	0.217 ± 0.008	0.347 ± 0.022	**0.014 ± 0.001**	**0.020 ± 0.001**
	$\text{GNES}^{+human}_{+reg}$	**0.994 ± 0.001**	**0.975 ± 0.005**	**0.212 ± 0.010**	**0.343 ± 0.020**	**0.014 ± 0.000**	**0.020 ± 0.001**

bonds, and or groups with electron-donating and/or electron-withdrawing groups. Likewise, nodes and edges involved in resonance or conjugated systems that provide substantial electron delocalization (which are often attributes of highly stable compounds) were also indicated with high priority. Considering the examples from the TOX21 dataset at the last row of Figure 3, GNES is more accurate than Grad-CAM baseline in assessing the importance of the sulfonyl functional group and the corresponding resonance stabilization it experiences from the connected ring. Likewise, in the BACE example shown in the 4th row of Figure 3, GNES has a better focus in highlighting functional groups and reducing priorities for irrelevant regions compared to the baselines models.

Scene graphs: As shown in the top 4 rows in Figure 3, for scene graph data, the size of the circle denotes the size of the bounding box of the object, and the importance of the nodes and the edges are marked by the lightness of the circles and lines, respectively. As can be seen, in general, the GNES model can more accurately focus on the importance of objects (nodes) and relationships (edges) than the Grad-CAM baselines. For example, as shown in the first row in Figure 3, the GNES explanation successfully found it is important to highlight not only the giraffe itself, but also the background (such as the fields) and the relationship between the giraffe and the fields. In contrast, the Grad-CAM baseline, however, only focused on the giraffe itself. Another example can be the indoor example at the 3rd row in Figure 3, and we can see that GNES gave more importance to the background objects and relationships, which are more accurate explanation and decisive factor for classifying this sample as the "indoor" scene.

2) User Study Results on Scene Graphs: To further assess the quality and interpretability of the model generated explanation, we conducted a user study on scene graph datasets. The annotators were asked to give an overall evaluation of each of the model explanations, specifically focus on the quality and interpretability of the given explanation, including both node-level and edge-level explanation, as well as the consistency between the two as an overall explanation. The final results were obtained by a joint work of 3 annotators. Specifically, the process is as follows: the first annotator gives the initial assessment to all the samples considering only the

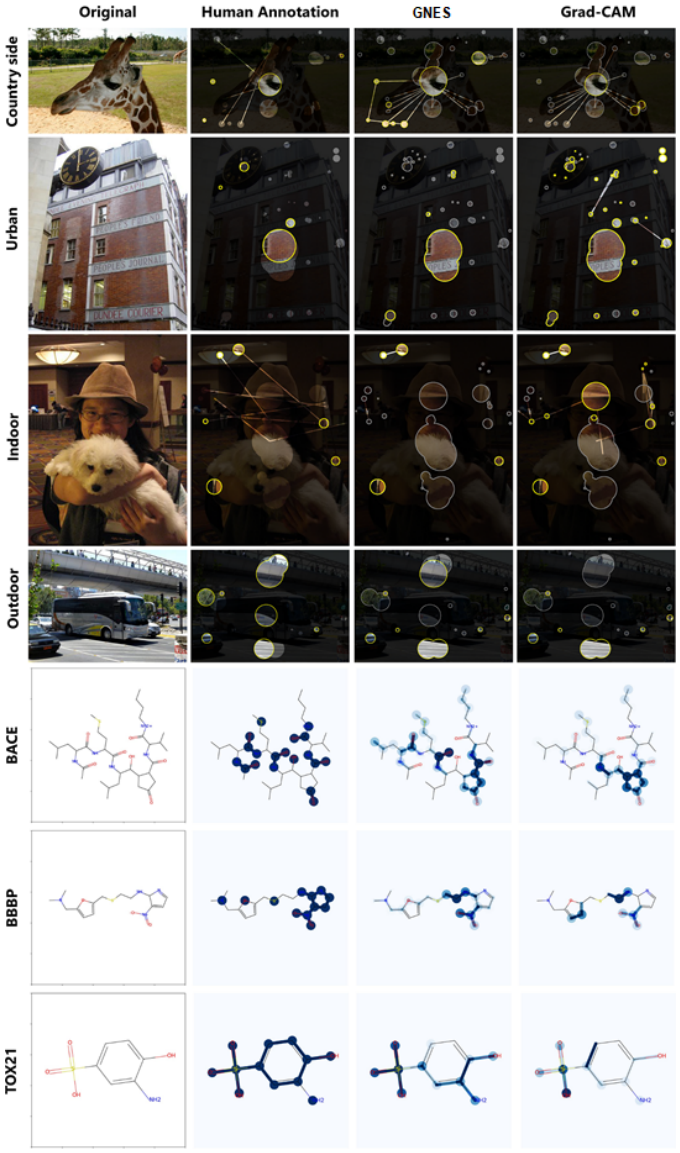

Fig. 3. Selected explanation results for Scene graph dataset (top 4 rows) and molecule datasets (bottom 3 rows). For scene graph data, the size of the circle denotes the size of the bounding box of the object and the importance is marked by the lightness of the circle and the yellow boundaries. For molecule graphs, the importance is marked by the darkness of blue circles on nodes and blue lines on edges. Darker color means more importance is given.

graph explanation itself; then, after the first annotator finished labeling the dataset, the second annotator is asked to review the initial assessment and provide a list of samples he/she disagrees with the first annotator; finally, the third annotator will look into the list of samples where the first two have a disagreement on the label and make a final decision for those samples.

As shown in Table III, we studied the quality for the two baseline explanations and our full framework (i.e., with both human annotation and graph regularization). As can be seen, our user study results further demonstrated that the proposed framework enhanced the GNN model's explainability by a huge margin. More specifically, our GNES model improved the quality of explanation on more than 40 (20%) samples in the test set of Indoor vs. outdoor datasets, and similarity turned more than 25 (13%) samples' explanation from bad to good quality. We argue that these results may have suggested that the GNES framework can have a big impact on the domains and applications, where the explainability of the machine learning model is crucial, and the data can be naturally presented in graphs/networks.

TABLE III
USER STUDY ON SCENE GRAPH DATASETS. THE ANNOTATORS WERE ASKED TO GIVE AN OVERALL EVALUATION SPECIFICALLY ON THE QUALITY OF THE GRAPH EXPLANATION (INCLUDING BOTH NODE-LEVEL AND EDGE-LEVEL EXPLANATIONS). THE FINAL RESULTS WERE OBTAINED BY A JOINT WORK OF 3 ANNOTATORS.

Dataset	Exp_Method	# good	# bad	Positive rate
Indoor vs. Outdoor	EB	100	100	50.0%
	GradCAM	140	60	70.0%
	GNES	181	19	**90.5%**
Country vs. Urban	EB	96	94	50.5%
	GradCAM	140	50	73.7%
	GNES	165	25	**85.8%**

V. CONCLUSIONS

This paper proposes a GNN Explanation Supervision (GNES) framework to adaptively learn how to explain GNNs more correctly. Specifically, our framework jointly optimizes both model prediction and model explanation by enforcing both whole graph regularization and weak supervision on model explanations. For the graph regularization, we propose a unified explanation formulation for both node-level and edge-level explanations by enforcing the consistency between them. The node- and edge-level explanation techniques we propose are also generic and rigorously demonstrated to cover several existing major explainers as special cases. Extensive experiments on five real-world datasets across two application domains demonstrate the effectiveness of the proposed model on improving the reasonability of the explanation while still keep or even improve the backbone GNNs model performance.

However, the improvement of the model explainability and the model performance does not come for free, as we have leveraged additional inputs from human explanation labels which may not be easily accessible. Although in our study we have demonstrated the effectiveness of the proposed GNES framework by only leveraging 10% of the training samples with human annotation on the explanations, this can still be prohibitive and unrealistic in practice for large-scale applications. To mitigate this limitation, our experimental studies suggest that proposing effective regularization terms that enforce some general rules can make the explanation more reasonable without the need for additional human annotation. In addition, designing effective unsupervised learning algorithms based on the model explanation might be one of the promising future directions to further overcome this limitation.

VI. Acknowledgments

This work was supported by the NSF Grant No. 1755850, No. 1841520, No. 2007716, No. 2007976, No. 1942594, No. 1907805, a Jeffress Memorial Trust Award, Amazon Research Award, NVIDIA GPU Grant, and Design Knowledge Company (subcontract number: 10827.002.120.04).

References

[1] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[2] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[3] S. R. Hong, J. Hullman, and E. Bertini, "Human factors in model interpretability: Industry practices, challenges, and needs," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, pp. 1–26, 2020.

[4] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability methods for graph convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 772–10 781.

[5] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 705–10 714.

[6] K. Annervaz, S. B. R. Chowdhury, and A. Dukkipati, "Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing," *arXiv preprint arXiv:1802.05930*, 2018.

[7] W. De Haan, Y. A. Pijnenburg, R. L. Strijers, Y. van der Made, W. M. van der Flier, P. Scheltens, and C. J. Stam, "Functional neural network analysis in frontotemporal dementia and alzheimer's disease using eeg and graph theory," *BMC neuroscience*, vol. 10, no. 1, pp. 1–12, 2009.

[8] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The World Wide Web Conference*, 2019, pp. 417–426.

[9] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[10] D. Matsunaga, T. Suzumura, and T. Takahashi, "Exploring graph neural networks for stock market predictions with rolling window analysis," *arXiv preprint arXiv:1909.10660*, 2019.

[11] L. Wu, P. Cui, J. Pei, and L. Zhao, *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer, 2021.

[12] F. Baldassarre and H. Azizpour, "Explainability techniques for graph convolutional networks," *arXiv preprint arXiv:1905.13686*, 2019.

[13] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," *Advances in neural information processing systems*, vol. 32, p. 9240, 2019.

[14] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, "Parameterized explainer for graph neural network," *arXiv preprint arXiv:2011.04573*, 2020.

[15] M. S. Schlichtkrull, N. De Cao, and I. Titov, "Interpreting graph neural networks for nlp with differentiable edge masking," *arXiv preprint arXiv:2010.00577*, 2020.

[16] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schütt, K.-R. Müller, and G. Montavon, "Higher-order explanations of graph neural networks via relevant walks," 2020.

[17] Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang, "Graphlime: Local interpretable model explanations for graph neural networks," *arXiv preprint arXiv:2001.06216*, 2020.

[18] Y. Zhang, D. Defazio, and A. Ramesh, "Relex: A model-agnostic relational model explainer," *arXiv preprint arXiv:2006.00305*, 2020.

[19] M. N. Vu and M. T. Thai, "Pgm-explainer: Probabilistic graphical model explanations for graph neural networks," *arXiv preprint arXiv:2010.05788*, 2020.

[20] H. Yuan, J. Tang, X. Hu, and S. Ji, "Xgnn: Towards model-level explanations of graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 430–438.

[21] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," *arXiv preprint arXiv:2012.15445*, 2020.

[22] D. Linsley, D. Shiebler, S. Eberhardt, and T. Serre, "Learning what and where to attend," *arXiv preprint arXiv:1805.08819*, 2018.

[23] M. Mitsuhara, H. Fukui, Y. Sakashita, T. Ogata, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Embedding human knowledge into deep neural network via attention map," *arXiv preprint arXiv:1905.03540*, 2019.

[24] T. Qiao, J. Dong, and D. Xu, "Exploring human-like attention supervision in visual question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[25] S. Chen, M. Jiang, J. Yang, and Q. Zhao, "Air: Attention with reasoning capability," in *European Conference on Computer Vision*. Springer, 2020, pp. 91–107.

[26] B. Patro, V. Namboodiri *et al.*, "Explanation vs attention: A two-player game to obtain attention for vqa," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 848–11 855.

[27] Y. Zhang, J. C. Niebles, and A. Soto, "Interpretable visual question answering by visual grounding from attention supervision mining," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 349–357.

[28] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?" *Computer Vision and Image Understanding*, vol. 163, pp. 90–100, 2017.

[29] A. Jacovi and Y. Goldberg, "Aligning faithful interpretations with their social attribution," *arXiv preprint arXiv:2006.01067*, 2020.

[30] A. S. Ross, M. C. Hughes, and F. Doshi-Velez, "Right for the right reasons: Training differentiable models by constraining their explanations," *arXiv preprint arXiv:1703.03717*, 2017.

[31] R. Visotsky, Y. Atzmon, and G. Chechik, "Few-shot learning with per-sample rich supervision," *arXiv preprint arXiv:1906.03859*, 2019.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[34] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[35] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," *International Journal of Computer Vision*, vol. 126, no. 10, pp. 1084–1102, 2018.

[36] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "Moleculenet: a benchmark for molecular machine learning," *Chemical science*, vol. 9, no. 2, pp. 513–530, 2018.

[37] I. F. Martins, A. L. Teixeira, L. Pinheiro, and A. O. Falcao, "A bayesian approach to in silico blood-brain barrier penetration modeling," *Journal of chemical information and modeling*, vol. 52, no. 6, pp. 1686–1697, 2012.

[38] G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny, "Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches," *Journal of chemical information and modeling*, vol. 56, no. 10, pp. 1936–1949, 2016.

[39] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, "Deeptox: toxicity prediction using deep learning," *Frontiers in Environmental Science*, vol. 3, p. 80, 2016.

[40] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, no. 1, pp. 32–73, 2017.

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.