# **Cooperative Explanations of Graph Neural Networks**

Junfeng Fang fjf@mail.ustc.edu.cn University of Science and Technology of China

Zemin Liu liu.zemin@hotmail.com

National University of Singapore

# ABSTRACT

With the growing success of graph neural networks (GNNs), the explainability of GNN is attracting considerable attention. Current explainers mostly leverage feature attribution and selection to explain a prediction. By tracing the importance of input features, they select the salient subgraph as the explanation. However, their explainability is at the granularity of input features only, and cannot reveal the usefulness of hidden neurons. This inherent limitation makes the explainers fail to scrutinize the model behavior thoroughly, resulting in unfaithful explanations.

In this work, we explore the explainability of GNNs at the granularity of both input features and hidden neurons. To this end, we propose an explainer-agnostic framework, Cooperative GNN Explanation (CGE) to generate the explanatory subgraph and subnetwork simultaneously, which jointly explain how the GNN model arrived at its prediction. Specifically, it first initializes the importance scores of input features and hidden neurons with masking networks. Then it iteratively retrains the importance scores, refining the salient subgraph and subnetwork by discarding low-scored features and neurons in each iteration. Through such cooperative learning, CGE not only generates faithful and concise explanations, but also exhibits how the salient information flows by activating and deactivating neurons. We conduct extensive experiments on both synthetic and real-world datasets, validating the superiority of CGE over state-of-the-art approaches like GNNExplainer and PGExplainer. Code is available at https://anonymous.4open.science/ r/CGE demo-2AD0.

# **CCS CONCEPTS**

• Security and privacy  $\rightarrow$  Web application security; • Computing methodologies  $\rightarrow$  Neural networks.

Conference'17, July 2017, Washington, DC, USA

© 2018 Association for Computing Machinery.

https://doi.org/XXXXXXXXXXXXXXX

Xiang Wang\* xiangwang1223@gmail.com University of Science and Technology of China

An Zhang an zhang@nus.edu.sg National University of Singapore

Tat-Seng Chua

dcscts@nus.edu.sg

National University of Singapore

Xiangnan He xiangnanhe@gmail.com University of Science and Technology of China

**KEYWORDS** 

Explainability, Graph Neural Networks, Lottery Ticket Hypothesis

#### **1 INTRODUCTION**

Graph neural networks (GNNs) [7, 31] have achieved promising performance in a variety of domains, where graph-structured data are involved, such as biochemistry [18, 39], social networking [11, 16], and e-commerce [14, 41]. The success comes mainly from the powerful expressiveness of GNNs, which incorporates the graph structure into representation learning. However, it usually comes at the cost of model opacity - that is, GNNs work as a black box, making the decision-making process obscure and hard to interpret [23]. Hence, researchers raised the question of post-hoc explainability: "What knowledge does the GNN use to make a certain prediction?".

Most prior studies [1, 15, 17, 32] realize post-hoc explainability from answering "Which fractions of input graph are most influential to the GNN's prediction?", thus generating explanations at the granularity of input features [3]. Scrutinizing the explainers proposed in these studies, we summarize the common scheme of feature attribution and selection. Specifically, given an input graph and its prediction, an explainer distributes the prediction to the input features, traces the importance of each feature, and selects the salient subgraph (e.g., a subset of edges with top importance) as the explanation of the input graph. For example, in the toxicity classification of molecule graph, some functional groups (e.g., cyano group) contain rich saliency information related to the model outcome and align well with human cognition. Such an explanatory subgraph can be viewed as the prototypical knowledge memorized by the GNN classifiers.

However, there are two intrinsic limitations in this scheme:

- The explainers focus solely on the explainability of input features, while leaving the explainability of hidden neurons (or units) [2, 38] unexplored. That is, they only highlight the contribution of each feature to the prediction, without probing into the role of each neuron in making the decision. Considering the GNN classifier that predicts a molecule graph as toxic, even if the explainer can latch on a cyano group as the explanatory subgraph to toxicity, it is unable to answer "Which fractions of classifier neurons are responsible for capturing the cyano group?". Hence, limiting the explanations to input features fails to scrutinize the model behavior thoroughly.
- Most explainers suffer from the out-of-distribution (OOD) issue, which is caused by the distribution shift between original input graphs and explanatory subgraphs. As shown in recent works

<sup>\*</sup>Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00



Figure 1: The framework of CGE, which incorporates the LTH with the criterion of MMI to achieve cooperative explanation. Best viewed in color.

[19, 25, 30], OOD makes the post-hoc explanations less faithful and reliable to reveal the decision-making process. Hence, these works instead incorporate the feature attribution and selection scheme into the GNN model, so as to make the model intrinsically interpretable [25]. However, such intrinsic interpretability poses hurdle to explainers from explaining the trained models.

In this work, we explore the explainability of GNN models at the granularity of both input features and model neurons, aiming to remedy the aforementioned limitations. To *hit two birds with one stone*, we propose a new scheme, <u>C</u>ooperative <u>GNN Explanation</u> (CGE), which incorporates the lottery ticket hypothesis (LTH) [8, 9] with the criterion of maximum mutual information (MMI) [3, 32]. Intuitively, LTH strives to specify a sparse subnetwork from the GNN via iterative pruning, which can be trained independently to maintain the original performance; meanwhile, MMI refines a subgraph from the original input, which maximizes the mutual information between the subgraph and the label.

Targeting on the integration of LTH and MMI, we systemize CGE as a combination of two components: subgraph detector and subnetwork detector. The subgraph detector can be instantiated with one of the current input-aware explainer (e.g., GNNExplainer [32], SA [1], PGExplainer [17], Gem [15], GSAT [19]), while the subnetwork detector can be implemented with the winning ticket [8, 9] of LTH. They play a cooperative game and update iteratively, as shown in Figure 1. Specifically, in each pruning iteration, the subgraph detector first identifies a subgraph based on MMI. Then, the subgraph is fed into the target model to score the importance of neurons. Hereafter, inspired by LTH, the subnetwork detector discards the neurons with the lowest importance scores, and then selectively rewinds them to the initial states to get the winning ticket (i.e., the subnetwork). In view of interdependent and iterative training between two detectors, we exploit the Expectation-Maximization (EM) algorithm [5, 10, 21] to guarantee convergence. This iterative pruning leads to the cooperative explanations, which not only exhibits how salient information flows in the GNN model by activating neurons, but also helps delineate the class-aware semantics of neurons.

Our main contributions can be summarized as:

• We highlight the importance of cooperative explanation at the granularity of both input features and model neurons.

- We propose a general framework, CGE, which integrates LTH and MMI to generate the explanatory subgraph and subnetwork simultaneously.
- Extensive experiments showcase the superiority of CGE over current explanation methods with better explainability.

#### 2 PRELIMINARY AND RELATED WORK

In this section, we first introduce the backgrounds of GNNs, and then present the conventional formulation and existing methods of GNN explainability *w.r.t.* input features.

# 2.1 Graph Neural Networks (GNNs)

Let  $\mathcal{G}$  be a undirected input graph, which involves the node set  $\mathcal{V}$  and the edge set  $\mathcal{E}$ . For node  $v_i \in \mathcal{V}$ , we denote its pre-existing features by a *d*-dimensional attribute vector  $\mathbf{x}_i \in \mathbb{R}^d$ , and collect the features of all nodes into  $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ . For the structural features describing the graph topology, we define an adjacency matrix  $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ , where  $A_{ij} = 1$  if the edge connecting nodes  $v_i$  and  $v_j$  exists (*i.e.*,  $(v_i, v_j) \in \mathcal{E}$ ), otherwise  $A_{ij} = 0$ . In a nutshell,  $\mathcal{G}$  can be alternatively represented as  $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ .

Upon the input graph, graph neural networks (GNNs) [7, 13, 27] are excellent at incorporating the graph structure into the representation learning by propagating and aggregating neural information along with the structure. Benefiting from high-quality representations, GNN models have achieved remarkable success in various tasks, including node classification [26, 40], graph classification [4, 37], and link prediction [33, 34]. In this work, we focus on the scenario of graph classification. Formally, we can systematize the GNN model f as a combination of two modules: the GNN encoder  $f_1$  and the classifier  $f_2$ , *i.e.*,  $f = f_2 \circ f_1$ . Typically, the GNN encoder  $f_1 : \mathbb{G} \to \mathbb{R}^{d'}$  generates the graph representation for the input graph G via three core stages: (1) recursively distill the neural information from neighboring nodes (or edges), (2) aggregate the information to update the representation of each ego node, and (3) garner all node representations as the d'-dimensional representation of the holistic graph. Subsequently, the classifier  $f_2: \mathbb{R}^{d'} \to \mathbb{R}^C$  maps the graph representation into the probability contribution over C classes. This process can be summarized as  $y = f_2(f_1(\mathcal{G})) = f(\mathcal{G}|\Theta)$ , where  $\Theta$  is the set of model parameters (i.e., neurons or units).

#### 2.2 Post-hoc Explainability of GNNs

The scheme of feature attribution and selection [1, 17, 28, 29, 32] prevails towards post-hoc explanations of GNN models. It focuses on explainability *w.r.t.* input features, aiming to answer "Which fractions of input graph contribute most to the model prediction?". Towards this end, it usually employs an additional explainer method to trace contributions of individual features, and then selects the salient part (*e.g.*, a subset of nodes or edges with top contributions) as the explanatory subgraph  $\mathcal{G}_s$ . We formulate the input-aware explainer as  $s_{inp}$ , which yields the explanatory subgraph  $\mathcal{G}_s$ , *i.e.*,  $\mathcal{G}_s = s_{inp}(\mathcal{G}|\Theta)$ .

To obtain the subgraph that best supports the predictive label Y, a prevailing criterion of maximum mutual information (MMI) [3, 32] identifies the explanatory subgraph that solely maximizes the information amount about Y. More formally, MMI is presented Cooperative Explanations of Graph Neural Networks

as:

$$\max_{\mathcal{G}_s} I(\mathbf{Y}, \mathcal{G}_s) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathcal{G}_s), \quad \text{s.t.} \quad \mathcal{G}_s = s_{\text{inp}}(\mathcal{G}), \quad (1)$$

where  $I(\cdot, \cdot)$  measures the mutual information between two variables. Following previous studies [1, 17, 28], we focus on the structural features (*i.e.*, the presence of an edge and its nodes), leaving the node features in future work.

Feature Attribution. The mainstream explainers to attribute the prediction to input features can be roughly categorized into three groups: gradient-, attnetion-, and perturbation-based research lines. Specifically, gradient-based line [1, 22] backpropagates gradients of the target prediction w.r.t. the input features. Such gradient-like signals are viewed as the approximations of feature importance. For example, SA [1] directly calculates the squared values of gradients to represent the importance scores of nodes. Perturbation-based line [15, 32, 36] studies the output variations in response to different input perturbations. This line is inspired by the intuition that the downstream predictions are likely to significantly change if sensitive features are perturbed. For example, GNNExplainer [32] adds soft masks to the input graph's adjacency matrix and node features, and trains them by maximizing the mutual information between the masked outcome and target prediction. Attention-based line [17, 19, 29] focuses on training an attention function for edge attribution according to input features. For example, GSAT (in its post-hoc working mode) [19] trains a parameterized predictor to generate the stochastic attention for each edge in the input graph.

**Feature Selection.** In the process of feature selection, subgraph candidates are guided by diverse constraints (*e.g.*, sparsity constraints, connective constraints, information bottleneck constraints). In more detail, sparsity constraints [28, 32] typically leverage the  $l_1$  norm to guarantee that the selected subgraph remains within a prescribed size. As the determinant subgraphs are expected to be connected, connective constraints [17, 36] are used to allocate more selective probabilities to the edges, which connect with the part selected already. More recently, information bottleneck constraints [19] are proposed to squeeze the mutual information between the selected subgraph and the input graph.

# **3 ANALYSIS ON POST-HOC EXPLAINABILITY**

Aside the success of the post-hoc explainability, considerable attention has recently been paid to the inherent issues of this trajectory [19, 30]. In this section, we first reveal the fundamental limitations (*i.e.*, OOD issue) of the post-hoc explainability; then we employ EM framework [5, 21] to derive a novel shortcut to reveal this limitation from the iterative optimization perspective. Furthermore, we demonstrate the importance of cooperative explainability and illustrate the advantages of the cooperative explainer.

#### 3.1 Revealing the OOD Limitation

We first formalize the date distribution of target model as  $P(\mathbb{G}, \mathbb{Y})$ . To approximate MI between  $\mathcal{G}$  and  $\mathbf{Y}$ , the target model  $f(\cdot|\Theta)$  in function space:  $f : \mathbb{G} \to \mathbb{R}^C$  is optimized via maximizing MI between  $f(\mathcal{G}|\Theta)$  and  $\mathbf{Y}$ :

$$\Theta := \arg \max_{\theta} I\left(f(\mathcal{G}|\theta); \mathbf{Y}\right),\tag{2}$$



Figure 2: Illustration of limitations of post-hoc explanation methods. Best viewed in color.

where  $(\mathcal{G}, \mathbf{Y})$  is independent and identically distributed (IID) from  $P(\mathbb{G}, \mathbb{Y})$ . Now consider the latent distribution of subgraphs with their labels formulated as  $P(\mathbb{G}_s, \mathbb{Y})$ . Analogously, to approximate MI between  $\mathcal{G}_s$  and  $\mathbf{Y}$ ,  $f(\cdot | \hat{\Theta})$  is indispensable which conforms to:

$$\hat{\Theta} := \arg \max_{\theta} I\left(f(\mathcal{G}_{s}|\theta); \mathbf{Y}\right),\tag{3}$$

where  $(\mathcal{G}_s, \mathbf{Y})$  is IID sampled from  $P(\mathbb{G}_s, \mathbb{Y})$ .

As shown in recent works [19, 30], the distribution of full graphs differs from that of subgraphs, which poses an out-of-distribution (OOD) issue in the data space. Worse still, comparing Equations (2) and (3), we argue that OOD between  $P(\mathbb{G}, \mathbb{Y})$  and  $P(\mathbb{G}_s, \mathbb{Y})$  in the data space consequently leads to OOD between  $P(f(\cdot|\Theta))$  and  $P(f(\cdot|\hat{\Theta}))$  in the function space, as shown in Figure 2. Therefore,  $I(f(\mathcal{G}_s|\Theta); Y)$  is not strictly proportional to  $I(\mathcal{G}_s; Y)$ . As a result, treating  $f(\cdot|\Theta)$  as the proxy of  $f(\cdot|\hat{\Theta})$  is an inherent limitation of post-hoc explainability. That is, unaware of this limitation, most previous explainers [17, 19, 29, 32] simply feed subgraph  $f(\cdot|\Theta)$  into the original network  $\Theta$ , rather than  $f(\cdot|\hat{\Theta})$ , and then use  $I(f(\mathcal{G}_s|\Theta); Y)$ as the main part of the loss function to optimize the explainer.

# 3.2 Remedying the OOD limitation

According to Equation (3), the learning of explanatory subgraph  $G_s$ and the approximation of  $f(\cdot|\hat{\Theta})$  are mutually and interdependently promoted. Hence, the conventional one-shot training stops short for optimizing  $G_s$  and  $\hat{\Theta}$ . To get around this dilemma, the existing explainers take  $f(\cdot|\Theta)$  as the proxy of  $f(\cdot|\hat{\Theta})$ , thereby suffering from the OOD limitation and making the explanations less faithful. In this work, we aim to confront the problem and employ the idea of Expectation-Maximization (EM) [5, 10, 21] to alleviate it. Generally, EM algorithm alternates between expectation step (E-step) and maximization step (M-step):

- *E-step:* Estimate parameters from observed data and existing models, then use this estimated parameter value to calculate the conditional probability expectation.
- *M-step:* Find the corresponding parameter via maximizing the likelihood function.

We then concrete these two steps to remedy the OOD limitation of post-hoc explainability. Concisely, in E-step, we fix the parameters  $\theta$  and extract subgraph  $G_s$ ; in M-step, we fix the subgraph  $G_s$ and optimize the parameters  $\theta$ . These two phases can be formulated as following, where *t* is the epoch of training:



Figure 3: Comparison between existing explainers and cooperative explainer. The graph example from the BA-3motif dataset is predicted as "Cycle" because of the insertion of cycle motif. The first line exhibits the forward propagation of input graph; The second line demonstrates the latent process of existing explainer; The third line shows the process of cooperative explainer, which follows the form of feature-neuron cooperative explanation. Best viewed in color.

*E-step*: Use the estimated function *f*(·|*θ*<sup>(t)</sup>) and label Y to calculate the conditional probability expectation of subgraph *G<sub>s</sub>*<sup>(t)</sup>:

$$P\left(\mathcal{G}_{s}^{(t)} \mid \mathbf{Y}, \theta^{(t)}\right) \to E_{\mathcal{G}_{s}^{(t)} \mid \mathbf{Y}, \theta^{(t)}}[\log P(\mathcal{G}_{s}^{(t)}, \mathbf{Y} \mid \theta)].$$
(4)

*M*-step: Find the corresponding f(·|θ<sup>(t+1)</sup>) when the likelihood function P(G<sub>s</sub><sup>(t)</sup>, Y|θ<sup>(t)</sup>) is maximized:

$$\theta^{(t+1)} := \arg \max_{\theta} E_{\mathcal{G}_{s}^{(t)} \mid \mathbf{Y}, \theta^{(t)}} [\log P(\mathcal{G}_{s}^{(t)}, \mathbf{Y} \mid \theta)].$$
(5)

By alternating between these two phases, we can explanatory subgraph  $\mathcal{G}_s$  and target function  $f(\cdot|\hat{\Theta})$  can be gained in a finite number of iterations.

#### 3.3 Advantages of Cooperative Explainability

Besides the distribution shifts in function space, limiting the explanations to input features solely is another issue of the current post-hoc explainers. Distinct from prior studies, cooperative explainability performs explanation on both subgraph and subnetwork. Scrutinizing "Which fractions of input features and model neurons are most influential to the target prediction?" allows us to interpret the input-neuron relationships during the decision-making process. Moreover, different groups of neurons might be activated for different explanatory subgraphs, especially in distinct predictive classes, thus working as experts [2, 38] to latch on the class-wise knowledge. Finding such meaningful abstractions (*i.e.*, the pair of subgraph and subnetwork) is one of the main goals of explainability.

Cooperative explainability can also boost the accuracy of posthoc explainers. Since explanation methods focus on interpreting a certain graph merely, generating explanation on the full network inevitably extracts various class-wise features, which introduces noise into explanations (See Figure 3). As cooperative explainability is able to outscore the salient neurons that focus on the certain class-wise pattern, it strongly squeezes the tightness of input edges and model neurons and shields the negative influence from the irrelevant or redundant neurons. In conclusion, cooperative explainability can not only endow the post-hoc explainers with better performance but also delineates the class-aware semantics of neurons. We will verify these advantages in Section 5.

# 4 COOPERATIVE GNN EXPLANATION

In this section, we first introduce the tractable objective for EM framework in Section 4.1; then we detail the implementation of Cooperative GNN Explanation (CGE), where the EM framework is concreted via combining MMI and LTH (Section 4.2 and Section 4.3). Moreover, we show how the CGE exhibits information flow through the model in Section 4.4.

# 4.1 A Tractable Objective for EM framework

Following Section 3, Our CGE aims to: (1) search explanatory subgraph  $\mathcal{G}_s$  and latent function  $f(\cdot|\hat{\Theta})$  to remedy the OOD limitation and (2) extract explanatory subnetwork to erasure the noisy features in explanations. To kill two birds with one stone and accelerate convergence, CGE constraints the initial states and searching domain of  $f(\cdot|\hat{\Theta})$  to the original network  $f(\cdot|\Theta)$  and the set of subnetwork of  $f(\cdot|\Theta)$ . CGE then employs two cooperative players to implement E-step and M-step alternately: (1) the subgraph detector  $s_1$ , which functions similarly to the feature-aware explainer (*cf.* Equation (1)) and extracts the sparse but critical subgraph from the full input graph; and (2) the subnetwork detector  $s_2$ , which captures the crucial neurons activated by the explanatory subgraph. These two detectors have the shared goal of recovering the target prediction, so as to exhibit the input-neuron-output relationships inherent in the model, as shown in Figure 1.

Since the conditional probability expectation in Equation (2) and Equation (3) is intractable, cooperative MMI is introduced to make the computation traversing from probability estimation to mutual information. More specially, we formulate cooperative MMI as:

$$\max_{\mathcal{G}_{s},\Theta_{s}} I(\mathbf{Y};\mathcal{G}_{s},\Theta_{s}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathcal{G}_{s},\Theta_{s}),$$
  
s.t.  $\mathcal{G}_{s} = s_{1}(\mathcal{G}), \quad \Theta_{s} = s_{2}(\Theta),$  (6)

where  $\Theta_s$  is the parameters of explanatory subnetwork that takes the explanatory subgraph  $\mathcal{G}_s$  as the input and yields the prediction  $f(\mathcal{G}_s|\Theta_s)$ ;  $H(\mathbf{Y}|\mathcal{G},\Theta_s)$  is the cross entropy between the recovered prediction  $f(\mathcal{G}_s|\Theta_s)$  and the label Y. Both subgraph and subnetwork detectors are forced to select the edge set and neuron set with predefined sizes or sparsity levels. When observing the explanatory subgraph, the subnetwork detector decides which neurons will be activated or deactivated to maintain the target prediction.

These detectors essentially indicate contributions to the input edges and model neurons. Hence, it seeks to find two masks,  $M_1$  and  $M_2$ , and frames Equation (6) as:

$$\max_{\mathbf{M}_1,\mathbf{M}_2} I(\mathbf{Y};\mathbf{M}_1\odot\mathbf{A},\mathbf{M}_2\odot\Theta), \tag{7}$$

where  $M_1$  and  $M_2$  are the masks with the sparsity constraint, which have identical shapes to the adjacency matrix A and the parameter set  $\Theta$ , respectively;  $\odot$  is the element-wise product.

Cooperative MMI criterion can identify the explanatory subgraph and subnetwork that collaborates together and maximizes the predictive performance. In this case, we reframe E-step (*i.e.*, subgraph detector) and M-step (*i.e.*, subnetwork detector) as:

*E-step:* Given estimated function  $f(\cdot|\theta^{(t)})$  and label Y, extracting explanatory subgraph  $\mathcal{G}_s^{(t)}$  via maximizing the mutual information between output and Y:

$$\mathcal{G}_{s}^{(t)} \coloneqq \arg \max_{\mathcal{G}_{s}} I(f(\mathcal{G}_{s} \mid \theta^{(t)}); \mathbf{Y}).$$
(8)

*M-step:* Taking subgraph  $\mathcal{G}_s^{(t)}$  as input to extract explanatory subnetwork  $f(\cdot|\theta^{(t+1)})$  via maximizing the mutual information between output and Y:

$$\theta^{(t+1)} := \arg \max_{\theta} I(f(\mathcal{G}_s^{(t)} \mid \theta); \mathbf{Y}).$$
<sup>(9)</sup>

Note that according to the properties of *Maximum Likelihood Estimation (MLE)* and *Jensen's inequality*, EM algorithm is proved to be deterministically convergent [5, 21]. Similarly, while merging *MLE* and *Jensen's inequality* into information theory, CGE can be proved to be convergent in a finite number of iterations.

#### 4.2 Implementation of Subgraph Detector

To extract the subgraph following Equation (8), CGE generalizes current feature-aware explainers *h* to the subgraph detector. Given graph  $\mathcal{G}_s$  and network  $f(\cdot|\Theta_s)$ , explainer *h* first attributes the prediction  $f(\mathcal{G}_s|\Theta_s)$  to the features in  $\mathcal{G}_s$  to extract an explanatory subgraph; then *h* yields the mask  $\mathbf{M}_1$  according to the explanatory subgraph. Specially, for the edges selected by the explainers *h*, we set their masks in  $\mathbf{M}_1$  to 1; for the edges unselected, we set their masks in  $\mathbf{M}_1$  to 0. We formulate this procedure as:

$$\mathbf{M}_{1}^{(t)} = h(\mathbf{M}_{1}^{(t-1)} \odot \mathbf{A}, f(\cdot \mid \mathbf{M}_{2}^{(t)} \odot \Theta)),$$
(10)

where *t* is the cycle of iteration. Note that this procedure is explaineragnostic, namely we can subsume arbitrary explainers under subgraph detector in CGE.

#### 4.3 Implementation of Subnetwork Detector

Although studies [17, 28, 32] have been extensively conducted on identifying the salient subgraph, network dissection of GNNs remains largely unexplored. To this end, we exploit the idea of LTH [8]. Specifically, LTH strives to specify a highly-spare subnetwork from the dense model via iterative pruning, such that independently training the subnetwork can achieve the matching performance to the dense model. Coincidentally, this idea is consistent with our aim for explanatory subnetwork. Hence, we incorporate LTH with MMI to implement subnetwork detector (*cf.* Equation 9), which consists of three key steps during one iteration: (1) distinguishable initialization of masks, (2) iterative selection of neurons, and (3) selective rewinding. We next elaborate these steps one by one.

4.3.1 **Distinguishable Initialization.** To learn the masks  $M_2$ , a straightforward solution is to randomly initialize them and optimize them via Equation (7). To understand the role of neurons, we go beyond the random initialization of  $M_2$  and leverage the magnitudes of neurons to guide the optimization of  $M_2$ . Intuitively, larger magnitudes indicate that the neurons are more crucial during making decisions, and vice versa. This distinguished initialization at the beginning of each iteration is able to accelerate the convergence of optimization and improve the identification of salient subnetwork. Formally, we summarize it as:

$$\mathbf{M}_{2}^{(0)} = \mathbf{R} + \sigma(\Theta/\bar{\Theta}), \quad \mathbf{M}_{2}^{(t)} = \mathbf{M}_{2}^{(t-1)} (\mathbf{R} + \sigma(\Theta/\bar{\Theta})), \quad (11)$$

where **R** is the randomly initialized scores; *t* is the cycle of iteration;  $\sigma$  is the sigmoid function that maps the scores to [0, 1];  $\Theta$  represents the magnitudes of neurons, with  $\overline{\Theta}$  as its median magnitude.

*4.3.2* **Iterative Selection.** Having established the modeling of neuron importance, we now optimize  $M_2$ . Specifically, our goal is to narrow the gap between the target prediction and the prediction upon the cooperative explanations. As such, we reframe Equation (7) as minimizing the following objective function:

$$\mathcal{L}_{\text{CGE}} = l(\mathbf{Y}, f(\mathbf{M}_1^{(t)} \odot \mathbf{A} \mid \mathbf{M}_2^{(t)} \odot \Theta)) + \gamma ||\mathbf{M}_2^{(t)}||_1, \quad (12)$$

where *l* measures the prediction gap;  $\gamma$  is the hyperparameter to control the *l*<sub>1</sub> sparsity of M<sub>2</sub>. In each iteration, subnetwork detector first receives M<sub>1</sub> from subgraph detector and distinguished initialize M<sub>2</sub>. When the training stage of M<sub>2</sub> is over, we can rank all neurons based on M<sub>2</sub>, and then select the crucial neurons with top importance scores. To be more specific, neurons with large scores are garnered to separately compose subnetwork  $f(\cdot|\Theta_s)$ , while the rest are not invited to take part in the following procedures.

4.3.3 **Selective Rewinding.** At the end of each iteration, we rewind the importance scores of neurons and transform  $\mathbf{M}_2^{(t)}$  to a binarization matrix as following:

- For the selected neurons, we rewind their masks in  $M_2^{(t)}$  to 1;
- For the unselected neurons, we isolate them from the following procedures and rewind their masks to 0.

CGE alternates between the phase of two detectors until the selection ratio reaches the prescribed threshold. The inessential correlations between edges and neurons are gradually eliminated during these iterations. After training, the final selections are considered as the most significant parts which contribute most to the target prediction. The achievement of CGE can be summarized as: (1) explanatory subgraph  $\mathcal{G}_s$  extracted by CGE contains the set of most label-relevant features and few noisy features; (2) explanatory subnetwork  $f(\cdot|\Theta_s)$  extracted by CGE can not only delineate the class-aware semantics of neurons, but also be the optimal approximation of latent function  $f(\cdot|\hat{\Theta})$ . In conclusion, CGE can unleash

the full potential of post-hoc explainers by remedying their inherent limitations and achieve cooperative explanation simultaneously.

#### 4.4 Exhibiting Information Flow via CGE

Besides the boosting accuracy of post-hoc explainers and achieving cooperative explanations, CGE can also exhibit how the salient subgraph is passed forward by activating and deactivating neurons. Specially, through inspection on the cooperative explanations (*i.e.*, the pair of explanatory subgraph and subnetwork), we observe that such abstractions somehow delineate the class-aware semantics:

- *Difference among different classes.* For input graphs belonging to different classes, a tiny ratio (about 30%) of neurons are activated simultaneously in their explanatory subnetworks. That is, the model parameters can be split into multiple experts [2, 38], each of which holds a class-wise view.
- *Consistency within identical classes.* A couple of input graphs from the same class activate a high ratio (more than 80%) of neurons simultaneously, indicating that the class-aware prototypical patterns are carried by similar explanatory subgraphs and memorized by shared subnetworks.

We present such neurons as the class-aware information flow. Formally, for class *c*, we first build a subset  $\mathcal{D}_c$  to collect graphs labeled with *c*, and hire CGE to generate the cooperative subgraph and subnetwork per graph. Then we set an enumerator T, which has the identical shapes to  $\Theta$ , to traverse the subnetworks collectively and count the activated frequency of each neuron. Frequently activated neurons explicitly show how the class-specific information flows in the model and arrives at the target prediction.

# **5 EXPERIMENT**

We present empirical results to demonstrate the effectiveness of our proposed CGE. The experiments aim to investigate the following research questions:

- **RQ1**: Can CGE boost the performance of existing methods when explainability is focused on input features solely?
- **RQ2:** How does the CGE perform while input features and model neurons are explained simultaneously?
- **RQ3**: Can CGE exhibit the information flow and delineate the class-aware semantics of neurons?

# 5.1 Experimental Settings

**Datasets and Target GNNs.** To evaluate the effectiveness of CGE, we adopt three benchmark datasets: BA3-motif [32], Mutagenicity [12], and MNIST [6], which are publicly accessible and vary in terms of domain, size, and sparsity. Three popular GNN models are trained to perform graph classification. Table 1 summarizes the statistics of datasets and the configurations of GNN models.

• Motif graph classification. We construct a synthetic dataset: BA-3motifs which contains 3,000 graphs. Following previous works [17, 32], we adopt the Barabasi-Albert (BA) graphs as the base and attach each graph with one of three motif types: house, cycle, and grid.

	BA3-motif	MNIST	Mutagenicity
Graphs#	3,000	70,000	4,337
Classes#	3	10	2
Avg.Nodes#	31.44	66.87	30.32
Avg.Edges#	31.24	725.39	30.77
Target GNNs	ASAP	GCN	GIN
Layers#	2	6	2
Testing Accuracy	0.942	0.886	0.823

- **Molecule graph classification.** We consider a real-world dataset, Mutagenicity [12, 24], where 4,337 molecule graphs are categorized into two classes based on their mutagenic effect on the Gram-negative bacterium.
- Handwriting graph classification. We use the MNIST superpixel dataset [20], which contains 70,000 graphs labeled as one of ten digit classes. In this dataset, the original MNIST images [6] are converted to graphs using super-pixels, which represent small regions of homogeneous intensity in images.

**Evaluation Metrics.** It is challenging to quantitatively evaluate the quality of explanations since human evaluations are highly dependent on their subjective understanding. Prior studies have proposed some metrics to quantitatively assess the explanations[7, 35], we select the following metrics:

- **Predictive Accuracy (ACC**@*p*) [3]. This metric measures the performance of the explanatory subgraphs by feeding it solely into the target model and auditing how well it recovers the target prediction. Where, *p* is the selection ratio;  $p \cdot ||\mathbf{A}||_0$  is the size of explanatory subgraph and  $p \cdot ||\Theta||_0$  is the size of explanatory subnetwork. We report the ACC-AUC as the area under the ACC curve over different selection ratios.
- **Precision**@*N* [32]. This metric measures the consistency between the explanatory subgraph and ground-truths subgraph. Specifically, the edges within the ground-truth subgraph are positive in an explanatory subgraph, while the remains are negative. In this case, precision can be adopted as the evaluation protocol. More formally:

Precision@N = 
$$\mathbb{E}_{\mathcal{G}}\left[\frac{|\mathcal{G}_{s} \cap \mathcal{G}_{*}|}{|\mathcal{G}_{*}|}\right]$$
 (13)

where  $\mathcal{G}_s$  is composed of the top-*N* edges and  $\mathcal{G}_*$  is the ground-truth subgraph.

 Fidelity@p [35]. The Fidelity metric studies the prediction change by removing important input features identified by explanation methods. Formally, fidelity can be defined as:

Fidelity @
$$p = \mathbb{E}_{\mathcal{G}}\left[f\left(\mathcal{G}\right)_{y} - f\left(\mathcal{G} - \mathcal{G}_{s}\right)_{y}\right]$$
 (14)

where y is the original prediction of  $\mathcal{G}$ ;  $\mathcal{G}_s$  is composed of the top  $p \cdot ||\mathbf{A}||_0$  edges;  $\mathcal{G} - \mathcal{G}_s$  is the subgraph created by removing  $\mathcal{G}_s$  from  $\mathcal{G}$ .

Alternative Baseline Approaches. To evaluate the quality of explanatory subgraphs, we compare our method with the stateof-the-art methods, covering the gradient-based methods (SA [1]), perturbation-based methods (GNNExplainer [32], Gem [15], SubgraphX [36]) and attention-based methods (PGExplainer [17], GSAT

	Mutagenicity BA3-motif			MNIST				
	ACC-AUC	Fidelity@0.5	ACC-AUC	Fidelity@0.5	Precision@5	ACC-AUC	Fidelity@0.5	Precision@10
SA	0.742	0.313	0.489	0.202	0.301	0.524	0.282	0.578
+CGE	$0.757 \pm 0.026$	$0.342 \pm 0.015$	$0.524 \pm 0.021$	$0.241 \pm 0.010$	$0.321 \pm 0.017$	$0.563 \pm 0.029$	$0.288 \pm 0.007$	$0.591 \pm 0.012$
GNNExplainer	0.845±0.082	$0.504 \pm 0.046$	$0.579 \pm 0.045$	$0.414 \pm 0.061$	$0.550 \pm 0.035$	$0.527 \pm 0.057$	0.277±0.018	$0.615 \pm 0.066$
+CGE	$0.875 \pm 0.057$	$0.530 \pm 0.038$	$0.592 \pm 0.028$	$0.429 \pm 0.031$	$0.589 \pm 0.020$	$0.542 \pm 0.044$	$0.296 \pm 0.018$	0.623±0.056
PGExplainer	0.862±0.056	0.557±0.029	$0.568 \pm 0.040$	$0.400 \pm 0.031$	$0.675 \pm 0.052$	$0.494 \pm 0.037$	$0.297 \pm 0.014$	0.513±0.022
+CGE	0.881±0.046	0.576±0.029	0.577±0.030	$0.413 \pm 0.023$	$0.684 \pm 0.036$	$0.524 \pm 0.020$	$0.312 \pm 0.013$	$0.547 \pm 0.010$
Gem	0.890±0.027	0.569±0.019	$0.591 \pm 0.014$	0.512±0.023	$0.724 \pm 0.030$	$\underline{0.616} \pm 0.026$	0.331±0.016	$0.564 \pm 0.025$
+CGE	0.892±0.037	$0.581 \pm 0.011$	$0.606 \pm 0.032$	0.525±0.025	$0.727 \pm 0.033$	0.631±0.049	$0.339 \pm 0.013$	$0.611 \pm 0.062$
SubgraphX	0.902±0.087	$0.599 \pm 0.064$	0.603±0.050	0.469±0.037	0.702±0.049	0.602±0.071	0.312±0.023	$0.526 \pm 0.031$
+CGE	0.913±0.077	0.612±0.019	0.621±0.022	$0.490 \pm 0.062$	$0.725 \pm 0.030$	$0.623 \pm 0.041$	$0.334 \pm 0.015$	$0.566 \pm 0.027$
GSAT	0.877±0.043	0.581±0.025	$0.614 \pm 0.037$	$\underline{0.478} {\pm 0.019}$	0.713±0.034	$0.547 \pm 0.024$	$\underline{0.342} \pm 0.018$	0.550±0.066
+CGE	$0.890 \pm 0.046$	$0.605 \pm 0.027$	0.624±0.034	$0.482 \pm 0.019$	0.731±0.031	$0.569 \pm 0.017$	0.353±0.030	$0.602 \pm 0.052$
Relative Impro.	4.7%	6.8%	4.6%	6.1%	5.2%	5.5%	7.2%	8.0%

Table 2: Quantitative analyses for explanation methods w.r.t. accuracy, fidelity and precious metrics. The best performing methods are bold with blue line, and the strongest baselines are underlined.

Table 3: Aveage ACC-AUC for cooperative explanations. The best performing methods are bold with blue line, and the strongest baselines are underlined.

	BA3-motif	MNIST	Mutagenicity
Random+LTH	0.364±0.094	0.146±0.083	0.589±0.76
SA+LTH	$0.428 \pm 0.013$	0.263±0.019	0.621±0.043
SA+CGE	$0.468 \pm 0.018$	$0.292 \pm 0.031$	0.632±0.036
GNNExplainer+LTH	$0.459 \pm 0.058$	0.306±0.035	0.667±0.082
GNNExplainer+CGE	0.491±0.029	$0.328 \pm 0.054$	0.752±0.050
PGExplainer+LTH	0.504±0.030	$0.284 \pm 0.040$	0.645±0.071
PGExplainer+CGE	0.562±0.033	$0.320 \pm 0.029$	0.704±0.039
Gem+LTH	0.491±0.053	0.315±0.077	0.701±0.044
Gem+CGE	0.545±0.019	0.352±0.065	0.738±0.049
SubgraphX+LTH	0.533±0.026	$0.323 \pm 0.048$	0.679±0.057
SubgraphX+CGE	0.572±0.023	0.350±0.073	0.714±0.060
GSAT+LTH	$0.541 \pm 0.045$	0.295±0.072	0.704±0.095
GSAT+CGE	0.584±0.035	$0.344 \pm 0.088$	0.743±0.056
Relative Inpro.	7.3%	11.4%	9.7%

in its post-hoc working mode[19]). Note that since IB regularization used in GSAT can not beforehand assign the certain sparsity tailored for numbers of iterations, for fair comparison, we employ  $l_1$  sparsity to replace it in experiments.

## 5.2 Evaluation of Explanatory Subgraph (RQ1)

**Qualitative evaluation.** Table 2 shows the accuracy, fidelity and precision of different post-hoc explanation methods. For simplicity, the baseline explainers enhanced by CGE are named as "explainers + CGE". According to Table 2 we find that:

• The current explainers enhanced by CGE outperform themselves in all cases. To be more specific, CGE achieves significant improvements over the strongest baselines *w.r.t.* fidelity by 6.8% and 7.2% in MNIST and BA3-motif, respectively. It demonstrates the effectiveness and universality of CGE, and verifies that CGE can be leveraged to boost the accuracy of post-hoc explainers. We attribute these improvements to the implementation of LTH and EM framework: (1) LTH allows CGE to filter the irrelevant and redundant information in model neurons by iteratively selecting and rewinding, which baseline explainers can not distinguish. (2) EM framework alternately propels meticulous selection to remedy the OOD limitation, which forces CGE to focus on the latent important features and leave obviously inessential features out in the early training stage.

- Compared with the performance in BA3-motif and Mutagenicity, CGE consistently performs better in MNIST across diverse metrics and baseline explainers. These improvements verify the theoretical analysis in Section 3.3. Specially, since cooperative explainability is able to outscore the salient neurons that focus on the certain class-wise pattern, CGE can shield the negative influence from the irrelevant or redundant neurons. As the consequence, CGE can achieve more significant enhancements while the number of classes in dataset  $\mathcal{D}$  is large.
- CGE provides much stabler explanation than the baselines as for the much smaller variance. More specially, STD of CGE outperforms baseline by a larger margin (24.5% ↓) on average. This phenomenon verifies that CGE can remedy the limitations of over-reliance on the quality of the target model.

**Qualitative evaluation.** We randomly choose the graph instances from class house, cycle, and grid in the synthetic BA3-motif dataset and present visual inspections of their explanatory subgraphs given by different explainers in Figure 4. For each explainer, we highlight the edges which have the top-K importance scores by red lines, where K=6. The ground-truth nodes are highlighted in green, while the turbulence nodes *w.r.t.* nodes in BA-motif are distinguished in blue. According to Figure 4 we can observe that:

- The important edges selected by the CGE framework largely conform to the ground-truth of their graphs, which might result from the class-wise patterns captured by the explanatory subnetwork. On the other hand, some edges not belonging to the ground-truth are selected by the baseline explainers.
- Compared with subgraphs selected by baseline explainers, subgraphs generated by CGE have better connectivity. We attribute

Conference'17, July 2017, Washington, DC, USA



Figure 4: Selected explanations in BA3-motif, where the top-6 of directed edges are highlighted by red lines. The ground-truth nodes are highlighted in green while the turbulence nodes are distinguished in blue. Best viewed in color.

these differences to the alternately and iterative selecting, which baseline explainers can not carry out.

• For blue nodes in BA motif, some nodes connecting to the green nodes might cause interference to the procedure of generating explanations. Our CGE can avoid these traps, while the subgraphs generated by baseline methods contain these turbulence nodes. This demonstrates the robustness and the reliability of the CGE.

# 5.3 Evaluation of Cooperative Explanation (RQ2)

We now focus on the explanations provided by CGE, and qualitative evaluation whether the CGE performs while input feature and model neurons are explained simultaneously. For fair comparison, we employ baseline explainers and the scheme of LTH to achieve one-shot explanations at the granularity of input features and neurons. The average ACC-AUC in different network sparsity are presented in Table 3. We find that:

• The subgraphs and subnetworks generated by the framework of CGE get the highest accuracy on the original graph classification task. Specifically, CGE achieves significant improvements over the strongest baselines *w.r.t.* ACC-AUC by 9.7% and 11.4% in Mutagenicity and MNIST, respectively. These improvements verify the reliability and effectiveness of the CGE. Since all baseline explainers employ the form of cooperative explanation and the LTH, we contribute these improvements mainly to the advantages of alternately extraction. Specifically, by leveraging alternative extraction, CGE iteratively filters the irrelevant and redundant mutual information between model neurons and target predictions, and generates faithful and concise cooperative explanations.

• Similar to the task in Section 5.2 *w.r.t* explainability of input features solely, CGE achieves more significant improvements while the number of classes in dataset  $\mathcal{D}$  is large. This phenomenon further verifies the theoretical analysis in Section 3.3.

# 5.4 Exhibition of Information Flow (RQ3)

We now focus on the class-specific information flow generated by CGE according to Section 4.4. Taking BA3-motif as an example, each layer in the relevant model *w.r.t* ASAP is divided into four parts according to explanatory subnetworks. The visual inspection of information flow is exhibited in Figure 5, where the numbers in network represent the ratios of neurons which class-wise inputs flow through to all neurons. To verify the fidelity and the robustness of information flow, we feed the class-specific subset  $\mathcal{D}_c$  into the relevant neurons (*i.e.*, class-specific information flow) and show the predictive accuracy in Figure 5.

We can find that for each class in BA3-motif, the accuracy of relevant information flow is not less than 84%. While the information flow is exhibited, CGE explicitly shows how the class-specific information put forward in the model by activating different neurons. Meanwhile, these abstractions also somehow delineate the class-aware semantics of neurons.

# 6 CONCLUSION

In this paper, we explored the cooperative explainability of graph neural networks, and proposed a general framework, Cooperative GNN Explanation (CGE), which incorporates LTH and MMI to generate the explanatory subgraph and subnetwork simultaneously. Cooperative Explanations of Graph Neural Networks



Figure 5: Class-specific Information Flow in ASAP model. Best viewed in color.

Focusing on cooperative explanations allows CGE to endow the conventional feature-based methods with better explainability, and exhibits how the salient information flows by activating and deactivating neurons. Extensive experiments in three datasets show that our method indeed improves the quality of explanatory subgraphs and subnetworks. This work represents an initial attempt to exploit cooperative explainability in GNN's explanations. In the future, we would like to consider more fine-grained relevancy between input features and model neurons, to analyze the key features which make predictions different.

# 7 ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2020AAA0106000), the National Natural Science Foundation of China (U19A2079, 62121002), the CCCD Key Lab of Ministry of Culture and Tourism, and Sea-NExT Joint Lab.

#### REFERENCES

- Federico Baldassarre and Hossein Azizpour. 2019. Explainability Techniques for Graph Convolutional Networks. CoRR abs/1905.13686 (2019).
- [2] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. 2020. Understanding the role of individual units in a deep neural network. Proc. Natl. Acad. Sci. USA 117, 48 (2020), 30071–30078.
- [3] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In *ICML*, Vol. 80. 882–891.
- [4] Ting Chen, Song Bian, and Yizhou Sun. 2019. Are Powerful Graph Neural Nets Necessary? A Dissection on Graph Classification. CoRR abs/1905.04579 (2019).
- [5] A. P. Dempster. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39 (1977).
- [6] Li Deng. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Process. Mag.* 29, 6 (2012), 141–142.
- [7] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2020. Benchmarking Graph Neural Networks. CoRR abs/2003.00982 (2020).
- [8] Jonathan Frankle and Michael Carbin. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *ICLR*.
- [9] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. 2020. Linear Mode Connectivity and the Lottery Ticket Hypothesis. In *ICML*, Vol. 119. 3259–3269.
- [10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. *CoRR* abs/1406.2661 (2014).
- [11] Zhiwei Guo and Heng Wang. 2021. A Deep Graph Neural Network-Based Mechanism for Social Recommendations. *IEEE Trans. Ind. Informatics* 17, 4 (2021), 2776–2783.
- [12] Jeroen Kazius, Ross McGuire, and Roberta Bursi. 2005. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of medicinal chemistry* 48, 1 (2005), 312–320.

- [13] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*.
- [14] Zhao Li, Xin Shen, Yuhang Jiao, Xuming Pan, Pengcheng Zou, Xianling Meng, Chengwei Yao, and Jiajun Bu. 2020. Hierarchical Bipartite Graph Neural Networks: Towards Large-Scale E-commerce Applications. In *ICDE*. 1677–1688.
- [15] Wanyu Lin, Hao Lan, and Baochun Li. 2021. Generative Causal Explanations for Graph Neural Networks. In ICML, Vol. 139. 6666–6679.
- [16] Yujia Liu, Kang Zeng, Haiyang Wang, Xin Song, and Bin Zhou. 2021. Content Matters: A GNN-Based Model Combined with Text Semantics for Social Network Cascade Prediction. In PAKDD (1) (Lecture Notes in Computer Science, Vol. 12712). 728–740.
- [17] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized Explainer for Graph Neural Network. In *NeurIPS*.
- [18] Mufti Mahmud, M. Shamim Kaiser, T. Martin McGinnity, and Amir Hussain. 2021. Deep Learning in Mining Biological Data. Cogn. Comput. 13, 1 (2021), 1–33.
- [19] Siqi Miao, Mia Liu, and Pan Li. 2022. Interpretable and Generalizable Graph Learning via Stochastic Attention Mechanism. In *ICML*. 15524–15543.
- [20] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. 2017. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In CVPR. 5425-5434.
- [21] Todd K. Moon. 1996. The expectation-maximization algorithm. IEEE Signal Process. Mag. 13 (1996), 47–60.
- [22] Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. 2019. Explainability Methods for Graph Convolutional Neural Networks. In CVPR. 10772–10781.
- [23] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In KDD. 1135–1144.
- [24] Kaspar Riesen and Horst Bunke. 2008. IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. In SSPR/SPR. 287–297.
- [25] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1, 5 (2019), 206-215.
- [26] Aravind Sankar, Xinyang Zhang, and Kevin Chen-Chuan Chang. 2019. Meta-GNN: metagraph neural network for semi-supervised learning in attributed heterogeneous information networks. In ASONAM. 137–144.
- [27] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *CoRR* abs/1710.10903 (2017).
- [28] Minh N. Vu and My T. Thai. 2020. PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks. In *NeurIPS*.
- [29] 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.
- [30] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. 2022. Discovering Invariant Rationales for Graph Neural Networks. *CoRR* abs/2201.12872 (2022).
- [31] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2019. A Comprehensive Survey on Graph Neural Networks. *CoRR* abs/1901.00596 (2019).
- [32] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating Explanations for Graph Neural Networks. In *NeurIPS*. 9240–9251.
- [33] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. Advances in neural information processing systems 31 (2018).
- [34] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. In NeurIPS.
- [35] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2020. Explainability in Graph Neural Networks: A Taxonomic Survey. CoRR abs/2012.15445 (2020).
- [36] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On Explainability of Graph Neural Networks via Subgraph Explorations. In *ICML*, Vol. 139. PMLR, 12241–12252.
- [37] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An End-to-End Deep Learning Architecture for Graph Classification. In AAAI. 4438– 4445.
- [38] Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2021. MoEfication: Conditional Computation of Transformer Models for Efficient Inference. CoRR abs/2110.01786 (2021).
- [39] Tianyi Zhao, Yang Hu, Linda R. Valsdottir, Tianyi Zang, and Jiajie Peng. 2021. Identifying drug-target interactions based on graph convolutional network and deep neural network. *Briefings Bioinform.* 22, 2 (2021), 2141–2150.
- [40] Fan Zhou, Chengtai Cao, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Ji Geng. 2019. Meta-GNN: On Few-shot Node Classification in Graph Meta-learning. In CIKM. 2357–2360.
- [41] Rong Zhu, Kun Zhao, Hongxia Yang, Wei Lin, Chang Zhou, Baole Ai, Yong Li, and Jingren Zhou. 2019. AliGraph: A Comprehensive Graph Neural Network Platform. Proc. VLDB Endow. 12, 12 (2019), 2094–2105.