

# Node classification in networks via simplicial interactions

Eunho Koo and Tongseok Lim

**Abstract**—In the node classification task, it is natural to presume that densely connected nodes tend to exhibit similar attributes. Given this, it is crucial to first define what constitutes a dense connection and to develop a reliable mathematical tool for assessing node cohesiveness. In this paper, we propose a probability-based objective function for semi-supervised node classification that takes advantage of higher-order networks’ capabilities. The proposed function reflects the philosophy aligned with the intuition behind classifying within higher order networks, as it is designed to reduce the likelihood of nodes interconnected through higher-order networks bearing different labels. Additionally, we propose the Stochastic Block Tensor Model (SBTM) as a graph generation model designed specifically to address a significant limitation of the traditional stochastic block model, which does not adequately represent the distribution of higher-order structures in real networks. We evaluate the objective function using networks generated by the SBTM, which include both balanced and imbalanced scenarios. Furthermore, we present an approach that integrates the objective function with graph neural network (GNN)-based semi-supervised node classification methodologies, aiming for additional performance gains. Our results demonstrate that in challenging classification scenarios—characterized by a low probability of homo-connections, a high probability of hetero-connections, and limited prior node information—models based on the higher-order network outperform pairwise interaction-based models. Furthermore, experimental results suggest that integrating our proposed objective function with existing GNN-based node classification approaches enhances classification performance by efficiently learning higher-order structures distributed in the network.

**Index Terms**—Node classification, semi-supervised, simplex, clique, node interaction, higher-order networks, hypergraph, probabilistic objective function.

## I. INTRODUCTION

Networks represented by graphs consist of nodes representing entities of the system, and edges depicting their interactions. Such graphical representations facilitate insights into the system’s modular structure or its inherent communities [1], [2]. While traditional graph analysis methods only considered pairwise interaction between nodes, recent researches, including those in social sciences [3] and biochemical systems [4], have experimentally demonstrated that networks in real systems often rely on interactions involving more than two nodes or agents. As a result, to analyze the attributes of a network, it is essential to illuminate the causal interactions of the network using higher-order networks (or hypergraphs) beyond pairwise relationships [5]. There are various approaches to

address this point of view, and recent studies are elucidating the relationships between cliques (a subset of nodes such that every two distinct nodes in the clique are adjacent) that form higher-order networks using probabilistic modeling based on the Stochastic Block Model (SBM) [6]–[8]. SBM is a generative model for random graphs that includes the following parameters: the number of nodes, the number of disjoint communities to which each node belongs, and the probability of edge connections between each community. The most common form of SBM assumes that the number of nodes in each community and the probability of edge connections within the same community are equal; nevertheless, several modified variants of SBM have also been studied [9], [10].

Studies related to community detection (or network clustering), on the other hand, have also been actively pursued [11]. The goal of these studies is to divide the entire system’s nodes into several communities, with nodes in each community being densely connected internally [12]. Research involving the higher-order network analysis has also advanced in this field, including the Bayesian framework [13],  $d$ -wise hypergraph SBM on the probability of a hyperedge [14], the sum-of-squares method of SBM [8], and spectral analysis based on the Planted Partition Model (PPM), a variant of SBM [7], [15]. Notably, many studies solely evaluate the network’s internal topology and do not account for prior information. However, in many real-world networks, even if it is a small part of the overall number of nodes, prior information is available; that is, we can use some known node labels as well as the total number of labels (communities). It has been reported that with only limited prior information, prediction accuracy and robustness in real noisy networks can be significantly improved [16], [17], and various methods have been suggested, including discrete potential theory method [18], spin-glass model in statistical physics application [16], strategies integrating known cluster assignments for a fraction of nodes [19], and nonnegative matrix factorization model [17]. Recently, the integration of semi-supervised learning with Graph Neural Network (GNN) has significantly enhanced performance of node classification on large datasets. Graph Convolutional Network (GCN) [20] generates new node representations by aggregating features from nodes and their neighbors. Graph-SAGE [21] introduces an inductive learning approach using node sampling and aggregation. Graph Attention Network (GAT) [22] apply attention mechanisms to neighboring nodes, and Jumping Knowledge Network (JKNet) [23] and Motif Graph Neural Network [50] collect information from an extended neighborhood range.

In this study, we propose a novel probability-based objective (loss) function for the semi-supervised node classification

Eunho Koo: Chonnam National University, Gwangju 61186, Republic of Korea (kooeunho@jnu.ac.kr)  
Tongseok Lim: Mitchell E. Daniels, Jr. School of Business, Purdue University, West Lafayette, Indiana 47907, USA (lim336@purdue.edu)

(community detection) task using higher-order networks. The loss function is motivated by the intuition that nodes densely interconnected with edges in a given network are likely to exhibit similar labels. It is intended to incentivize nodes in a hyperedge (a clique) to have the same label by imposing a natural penalty when nodes within the hyperedge have diverse labels. It is worth noting that the intuition is consistent with SBM's general assumption that nodes with the same label are more likely to be connected in a network. In conjunction with the objective function, we use discrete potential theory to initialize the node probability distribution, specifically the solution to an appropriate Dirichlet boundary value problem on graphs, which can be effectively solved using the concept of equilibrium measures [24].

We also propose a novel graph generation model, Stochastic Block Tensor Model (SBTM). In general, traditional SBM-generated networks differ significantly from many real-world networks. Specifically, when comparing networks of equivalent density (that is, networks with an identical count of nodes and edges), SBM-based models typically exhibit far fewer higher-order polyhedrons (simplices or cliques), than what is observed in real-world graphs. This limitation of SBM stems from its nature as an edge-generation model. For example, in social networks, while two people might form a friendship, it is also possible for three or more individuals to simultaneously establish a friendship. In light of this, we suggest that edge-generation models (SBM) have limits in producing network data that is similar to what is observed in the real world, and we offer a revised model capable of incorporating higher-order structures such as triangles and tetrahedrons into the network.

Finally, we propose a method that achieves additional performance enhancements by integrating our proposed objective function with state-of-the-art (SOTA) GNN-based semi-supervised learning techniques. To the best of the authors' knowledge, many SOTA algorithms do not distinctively utilize all higher-order structures within graphs, but often focusing primarily on edges (pairwise interactions) instead. We explore how combining the strengths of both approaches can lead to further performance gains, and we validate this synergy on real citation network datasets (Cora, CiteSeer, and PubMed).

Our contributions are as follows:

- We propose a novel probability-based objective function for the semi-supervised node classification task, designed to leverage the full higher-order network structures.
- We extend SBM and introduce SBTM, a network generation model designed to simulate more realistic networks by incorporating higher-order structures.
- We explore and validate an approach for improving performance by combining the most recent graph neural network-based node classification methods with the objective function suggested in this paper.

This paper is structured as follows. Some preliminary information is provided in Section II. The objective function is discussed in Section III. The experimental setup is described in Section IV. The result is evaluated in Section V. Finally, in Section VI, the conclusion and future work are presented.

## II. PRELIMINARIES

In this section, we describe the preliminary graph node classification approach based on discrete potential theory after introducing some essential mathematical concepts.

### A. Higher-order networks

In many real-world systems, network interactions are not only pairwise, but involve the joint non-linear couplings of more than two nodes [25]. Here, we fix some terminology on higher-order networks that will be used throughout the paper.

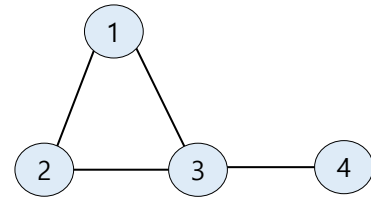
A (undirected) graph  $G = (V, E)$  consists of a set  $V = \{1, 2, \dots, n\}$  of  $n$  nodes, and a set  $E \subset \{(i, j) \mid i, j \in V, i \neq j\}$  of edges. We have  $(i, j) = (j, i) = \{i, j\}$  as  $G$  is undirected. We assume that a graph  $G$  is connected.<sup>1</sup>

A hypergraph generalizes  $E$  as  $\mathcal{E} \subset 2^V$  where  $2^V$  denotes the power set of  $V$ , and we denote the hypergraph as  $\mathcal{H} = (V, \mathcal{E})$ . In this paper, we focus on the case where  $\mathcal{E}$  consists of the *simplices* in  $G$ : For  $k \in \mathbb{N} \cup \{0\}$  (the set of non-negative integers), a subset  $\sigma = \{n_0, n_1, \dots, n_k\}$  of  $V$  is called a  $k$ -simplex (which is also called a  $(k+1)$ -clique) if the vertices  $n_i \in V$  are distinct (i.e.,  $|\sigma| = k+1$ ) and for every  $0 \leq i < j \leq k$ , we have  $(n_i, n_j) \in E$ . Let  $K_k = E_{k-1}$  denote the set of all  $k$ -cliques, or  $(k-1)$ -simplices, in  $G$ . Note that  $E_0 = V$ ,  $E_1 = E$ ; a node is a 0-simplex, an edge a 1-simplex, a triangle a 2-simplex, a tetrahedron a 3-simplex, and so on. The set comprising all cliques of a graph  $G$ ,

$$K(G) := \bigcup_{k=1}^{\omega(G)} K_k(G), \quad (1)$$

is called the *clique complex* of the graph  $G$ . The clique number  $\omega(G)$  is the number of vertices in a largest clique of  $G$  [26]. In this paper, we will consider  $\mathcal{E}$  a subset of  $K(G)$  in terms of a hypergraph.

**Example 1.** For a given  $V = \{1, 2, 3, 4\}$ , consider  $\mathcal{E} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 4\}, \{1, 2, 3\}\}$ .



There are four 0-simplices:  $K_1 = \{\{1\}, \{2\}, \{3\}, \{4\}\}$ ; four 1-simplices:  $K_2 = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 4\}\}$ ; and one 2-simplex:  $K_3 = \{\{1, 2, 3\}\}$ .

### B. Classification algorithm based on random walk on graphs

In the semi-supervised node classification (partially labeled data classification) tasks, a classic and widely used algorithm based on a random walk on a graph is the following. Given a graph  $G = (V, E)$ , a (unbiased) random walk moves from

<sup>1</sup>Because our proposed algorithm can be applied to each connected component of a graph, the assumption can be made without loss of generality.

$i \in V$  to  $j \in V$  with probability  $1/k$  if  $(i, j) \in E$  (i.e.,  $i$  and  $j$  are adjacent) and the degree of  $i$  (the number of nodes adjacent to  $i$ ) is  $k$ . For a node set  $V = \{1, 2, \dots, n\}$  and a label-index set  $I = \{1, \dots, l\}$ , we assume that each node corresponds to one label in  $I$  and we know the labels for only a small proportion of nodes relative to  $|V| = n$ . For  $i \in I$  and  $y \in V$ , let  $P_i(y)$  denote the probability that a random walk starting from an unlabeled node  $y$  will reach an  $i$ -labeled node before arriving at any other labeled nodes. If  $\arg\max_{i \in I} P_i(y) = k$ , the algorithm predicts that the label of the unlabeled node  $y$  is  $k$ . If a node  $y$  is already labeled as  $k$ , we have  $P_i(y) = 1$  if  $i = k$  and  $P_i(y) = 0$  if  $i \neq k$ . In this study, we refer to this algorithm as **RW**, which stands for Random Walk.

Now the question is how to obtain  $P_i(y)$  for all  $y \in V$  and  $i \in I$ . Potential theory shows  $P_i(y)$  can be obtained from the solution  $u$  of the following Dirichlet boundary value problem

$$\begin{aligned} Lu(x) &= 0 \quad \text{if } x \in F = (E_i \cup H_i)^c, \\ u(x) &= 1 \quad \text{if } x \in E_i, \\ u(x) &= 0 \quad \text{if } x \in H_i, \end{aligned} \quad (2)$$

where  $L = D - A$  is the graph Laplacian matrix where  $D, A$  are degree and adjacency matrix of a given graph  $G$  (see [26]),  $E_i$  is the set of  $i$ -labeled nodes,  $H_i$  is the set of labeled nodes excluding  $i$ -labeled nodes, and  $u$  is a function on  $V$ , valued in  $[0, 1]$ . Then it holds  $P_i(y) = u(y)$  for all  $y \in V$ .

Bendito, Carmona and Encinas [24] proposed an elegant solution to the Dirichlet problem (2) in terms of *equilibrium measures*. For any decomposition  $V = F \cup F^c$  where  $F$  and  $F^c$  are both non-empty, they showed that there exists a unique measure (function) such that  $Lv(x) = 1$  (and  $v(x) > 0$ ) for all  $x \in F$  and  $Lv(x) = 0$  (and  $v(x) = 0$ ) for all  $x \in F^c$ . The measure is called the equilibrium measure and denoted by  $v^F$ . Now for  $V = F \cup F^c$  where  $F, F^c$  are the set of unlabeled and labeled nodes, the solution  $u$  of (2) can be represented by

$$u(x) = \sum_{z \in E_i} \frac{v^{\{z\} \cup F}(x) - v^F(x)}{v^{\{z\} \cup F}(z)}, \quad x \in V. \quad (3)$$

Because  $v^F$  can be obtained by solving a linear program, (3) provides an efficient way to solve the Dirichlet problem (2); see [24] for more details.

We will use **RW** as a baseline algorithm for the semi-supervised node classification. Note that **RW** employs random walks and does not utilize higher-order interactions (HOI). However, **RW** will be useful not only for comparing performance with our HOI-applied strategies, but also for providing a useful initialization method for training HOI algorithms.

### III. PROPOSED MODEL

#### A. Simplicial objective function

We propose a novel objective function for node classification which utilizes higher-order networks. Given a graph  $G$ , let  $V = \{1, \dots, n\}$  denote the node set,  $I = \{1, \dots, l\}$  denote the label index set, so the graph consists of  $n$  nodes, and each node  $j \in V$  has a label  $i \in I$ . Probability distribution over the labels for the node  $j$  is denoted by  $(p_1^j, p_2^j, \dots, p_l^j)$ , where  $p_i^j$  denotes the probability that node  $j$  having label  $i$ , thus  $\sum_{i=1}^l p_i^j = 1$

for every  $j \in V$ . We define  $K_k$  as the set of  $(k-1)$ -simplices in the graph, e.g.,  $K_1, K_2, K_3$  correspond to the set of nodes, edges, triangles, respectively. Let  $M = \omega(G)$  be the maximum possible value of  $k$ , that is, the simplex composed of the most nodes in the graph is a  $(M-1)$ -simplex with  $M$  nodes. Now we define an objective function for node classification task by

$$J = \sum_{k=2}^M w_k \sum_{(j_1, \dots, j_k) \in K_k} \sum_{(i_1, \dots, i_k) = \theta \in I^k} C_\theta p_{i_1}^{j_1} p_{i_2}^{j_2} \dots p_{i_k}^{j_k} \quad (4)$$

where  $I^k = I \times \dots \times I$  ( $k$  times),  $w_k \geq 0$  are weight hyperparameters, and  $C_\theta := \binom{k}{e_1, e_2, \dots, e_l} = \frac{k!}{e_1! e_2! \dots e_l!}$  where  $e_i$  is the number of occurrences of the label  $i$  in  $(i_1, i_2, \dots, i_k) = \theta$  with  $\sum_{i=1}^l e_i = k$ . Hence  $J$  is determined by the underlying graph structure, and is a function of probabilities  $\{p_i^j\}_{i \in I, j \in V}$ .

We then proceed to solve the minimization problem:

$$\text{Minimize } J \text{ over } \Delta^n = \Delta \times \Delta \times \dots \times \Delta \quad (5)$$

where  $\Delta$  is the probability simplex in  $\mathbb{R}^l$ , such that  $p^j := (p_1^j, p_2^j, \dots, p_l^j) \in \Delta$ .

The idea is that a higher penalty is assigned via a multinomial coefficient  $C_\theta$  to the simplex with a greater diversity of labels and vice versa, and we sum the penalties over all  $(k-1)$ -simplices, finally taking a weighted sum over  $K_k$ . This leads us to expect that a probability distribution  $\{p_i^j\}_{i \in I, j \in V}$  minimizing  $J$  over  $\Delta^n$  will find a node labeling that encourages the least diversity of labels within each simplex on average. This is consistent with our model assumption that the connection probability within the same label is higher than between different labels in forming the network. Finally, from a computational standpoint, solving the problem (5) may necessitate a suitable initialization of the value  $\{p_i^j\} \in \Delta^n$ . We employ **RW** and use its solution as our initial value, which is simple to compute through linear programs. This is the main idea of this paper and is illustrated in Figure 1.

In this paper, we use an exponential base weight  $w_k = \alpha^{k-1}$  in (4) with various base values  $\alpha > 0$ , and the default value of  $\alpha$  is set to 1 leading to the constant weight. It is worth noting that the expected number of  $k$ -cliques can be shown to be bounded by  $\mathbb{E}[|K_k|] \leq 2^{k-1} |V|^k p^{\frac{k(k-1)}{2}}$ , if the underlying network is generated by a stochastic block model with an edge-probability matrix with probability  $p$  on the diagonal and  $q$  off-diagonal, where  $p > q$ . This explains why, contrast to the many real-world network datasets, higher-order simplicial structures become increasingly difficult to observe in SBM as  $k$  increases, due to the above bounding estimate with small  $p$ .

**Example 2.** Let  $I = \{1, 2\}$  and let  $K_3$  be the set of all 2-simplices in a given graph. Fix a member  $(j_1, j_2, j_3) \in K_3$ . Then possible combinations for the binary labeling leads to  $2^3 = 8$  terms of the form  $p_{i_1}^{j_1} p_{i_2}^{j_2} p_{i_3}^{j_3}$  where  $i_1, i_2, i_3 \in I$ , which represents the probability that the nodes  $j_1, j_2, j_3$  have labels  $i_1, i_2, i_3$ , respectively. We impose a penalty of multinomial order according to the distinct node labels within a given simplex. For a 2-simplex, the objective with respect to two

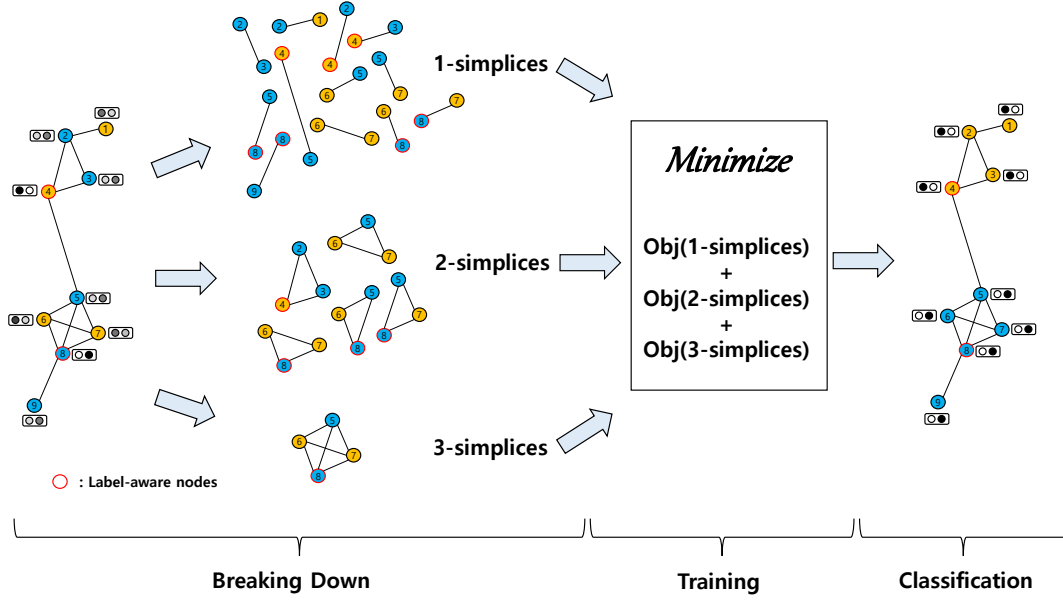


Fig. 1. Overall optimization process. First, we classify simplices in a given network based on their sizes (Breaking Down). Then, we multiply each size category by a distinct multinomial coefficient, sum these weighted values, and use this combined information to train a model (Training). Finally, we evaluate the model's classification performance (Classification).

labels classification, which is the third summation term in (4), consists of the following eight terms:

$$\begin{aligned} & \binom{3}{3,0} p_1^{j_1} p_1^{j_2} p_1^{j_3} + \binom{3}{2,1} p_1^{j_1} p_1^{j_2} p_2^{j_3} + \binom{3}{2,1} p_1^{j_1} p_2^{j_2} p_1^{j_3} \\ & + \binom{3}{2,1} p_2^{j_1} p_1^{j_2} p_1^{j_3} + \binom{3}{1,2} p_1^{j_1} p_2^{j_2} p_2^{j_3} + \binom{3}{1,2} p_2^{j_1} p_1^{j_2} p_2^{j_3} \\ & + \binom{3}{1,2} p_2^{j_1} p_2^{j_2} p_1^{j_3} + \binom{3}{0,3} p_2^{j_1} p_2^{j_2} p_2^{j_3}, \end{aligned}$$

where  $\binom{3}{3,0} = \binom{3}{0,3} = 1$ ,  $\binom{3}{2,1} = \binom{3}{1,2} = 3$ , and  $p_1^j + p_2^j = 1$ .

As can be inferred from Example 2, in the computation related to the edge set  $K_2$  in (4), the computational complexity is  $O(|K_2|k^l)$  where  $|K_2|$  is the number of edges in the network, and  $l$  is the number of labels. In general, for each  $k$ -clique set  $K_k$ , the computational complexity can be derived as  $O(|K_k|k^l)$ , and therefore, the total complexity for all simplices up to size  $M = \omega(G)$  would be  $O(\sum_{k=2}^M |K_k|k^l)$ . Since the complexity depends on the number of cliques in the network and the number of labels, the complexity of the proposed objective function increases with the inclusion of more higher-order simplices and a greater number of distinct labels.

### B. SBTM: a new graph generation approach

Initially conceptualized in social networks and bioinformatics, Stochastic Block Model (SBM) [6] harnesses a probabilistic approach, giving a simple generative model for random graphs. For a node set and a label set  $V$  and  $I$  as above, we consider  $l = |I|$  distinct labels or communities, each being a non-empty disjoint subsets of  $V$  for each  $i \in I$ . The connection probabilities between nodes in  $V$  by edges is identified by a  $l \times l$  edge-probability matrix  $B$ , where  $B_{ij}$

indicates the probability that a node belonging to the  $i$ th label connects with a node in the  $j$ th label by an edge. In many cases, the diagonal elements of  $B$  are greater than off-diagonal elements, implying that the connection probability within the same label is higher than between different labels.

However, SBM has notable limitations: it lacks robust local clustering, failing to capture dense substructures like triangles essential for representing complex network dynamics. SBMs also assume edge independence, which is unrealistic as actual networks exhibit interdependencies among edges and a mix of hierarchical community structures [27]. Consequently, SBMs may not accurately reflect the diverse characteristics and higher-order interactions commonly observed in real-world networks. To address this, in this paper we present the Stochastic Block Tensor Model (SBTM), which generalizes SBM by assigning probability tensors to generate  $k$ -cliques for each  $k \geq 2$ , facilitating the development of higher-order simplices in the network. This results in the formation of higher-order simplices in the synthetic network, yielding networks that closely mimic real-world networks.

To explain the concept of SBTM, we recall that a SBM is described by its probability matrix  $B \in \mathbb{R}^{l^2}$ , whose  $(i, j)$  entry  $B_{ij}$  reflects the probability that nodes  $i$  and  $j$  are connected by an edge in a random graph  $G$ . Each edge between  $i, j$  is generated independently in  $G$  through  $B_{ij}$ . Motivated by this, for each  $k \geq 2$ , let  $T_k \in \mathbb{R}^k$  be a probability tensor whose  $(i_1, \dots, i_k)$  entry  $(T_k)_{i_1, \dots, i_k}$  indicates the probability that the nodes  $i_1, \dots, i_k$  are interconnected as a  $k$ -clique in  $G$ . Given probability tensors  $T_2, T_3, \dots$ , we assume that an SBTM generates a random graph  $G$  as follows: for each  $k \geq 2$ , we generate  $k$ -cliques independently for every size- $k$  subset of  $V$  using  $T_k$ . Let  $G_k$  denote the graph that represents the

(unweighted) overlap of all randomly created  $k$ -cliques. The final graph  $G$  is defined as the full overlap of all  $G_k$ . Note that an edge between two nodes  $i, j$  can be generated more than once in this procedure. In this paper, we assume that  $G$  is unweighted, meaning that an edge between  $i, j$  exists in  $G$  if and only if it is produced in at least one of  $G_k$ . However, one can also define a weighted graph by counting the number of edge generation between each pair of edges. As a result, SBM is a specific instance of SBTM, with  $T_k \equiv 0$  for all  $k \geq 3$ .

Thus SBTM is given by a set of probability tensors  $(T_k)_{k \geq 2}$ . In this study, we will define these tensors as follows. Let  $N = [N_1, \dots, N_l]$  denote the number of nodes for each label. For each  $k \geq 2$ , we choose a probability matrix  $B_k \in \mathbb{R}^{l^2}$ .  $T_k$  will be defined using  $B_k$ . Given a sequence of labels  $(i_1, \dots, i_k) = \theta \in I^k$ , we define the  $\theta$ -entry of  $T_k$ , denoted by  $(T_k)_\theta$ , as

$$(T_k)_\theta := \prod_{i=1}^l \binom{N_i}{e_i} \prod_{1 \leq a < b \leq k} (B_k)_{i_a i_b}, \quad (6)$$

where  $e_i$  is the number of occurrences of label  $i$  in  $(i_1, \dots, i_k)$ . For example, in 4-clique generation, if  $N = [N_1, N_2, N_3]$  and  $\theta = (1, 1, 3, 1)$ , then  $e_1 = 3, e_2 = 0, e_3 = 1$ , and the entry at  $(1, 1, 3, 1)$  in the tensor  $T_4$  is given by (6) with  $l = 3, k = 4$ .

The formula (6) is motivated from calculating the expected number of  $k$ -cliques in the traditional SBM: The first term  $\prod_{i=1}^l \binom{N_i}{e_i}$  calculates the number of ways to select  $e_i$  nodes from each group of  $N_i$  nodes, and the second term  $\prod_{1 \leq a < b \leq k} (B_k)_{i_a i_b}$  is the product of the connection probabilities provided by  $B_k$  between nodes in a  $k$ -clique. The second term will be small if the clique has diversified labels because, intuitively, the diagonal entries of  $B_k$  are larger than the off-diagonal entries, reflecting a higher probability of connections within the same cluster than between different clusters. By assigning appropriate probabilities to each element of  $B_k$ , we intend to systematically determine a probability value to each entry of  $T_k$ , where it takes lower values for diversified labels and vice versa for uniform labels.

### C. Related works on graph semi-supervised learning

Graph semi-supervised learning (GSSL) is designed to utilize inexpensive unlabeled data to improve the model's effectiveness using only a limited number of expensive labeled data. Hence, the semi-supervised learning framework aligns well with numerous practical scenarios where acquiring labels is challenging. According to a taxonomy framework [28], GSSL can be categorized into graph regularization methods, matrix factorization-based methods, random walk-based methods, autoencoder-based methods, and GNN-based methods.

The graph regularization method in GSSL involves adding a smoothness constraint to the prediction function, ensuring that nodes with similar features have similar labels, often using the graph Laplacian to enforce this constraint. Examples of graph regularization methods include Gaussian random field [29], local and global consistency [30], directed regularization [31], linear neighborhood propagation [32], and manifold regularization with cosine distance [33]. The matrix factorization-based method in GSSL involves factorizing a matrix that represents relationships between node pairs to obtain node

embeddings, utilizing matrices like the adjacency matrix or the normalized Laplacian matrix to capture underlying graph structures. Notable works in this category include locally linear embedding [34], Laplacian eigenmaps [35], GraRep [36], and HOPE [37]. The random walk-based method in GSSL uses random walks to capture graph properties like node centrality and similarity. Key works in this category include DeepWalk [38], LINE [39], Planetoid [40], node2vec [41], and VG-GCN [42]. The autoencoder-based method in GSSL uses autoencoders to encode each node into a low-dimensional embedding while preserving graph structure. Significant autoencoder-based methods include structural deep network integration [43], deep neural networks for learning graph representations [44], graph autoencoder [45], and mGNN [46]. The GNN-based method in GSSL utilizes Graph Neural Networks to generate node embeddings by aggregating information from a node's neighbors. Prominent GNN-based methods include the graph neural network model [47], graph convolutional network [20], deep sets [48], graph-SAGE [21], graph attention network [22], jumping knowledge network [23], simplifying graph convolutional networks [49], Motif graph neural network (MGNN) [50], GDPNet [51], Infograph [52], NeuralSparse [53], PTDNet [54], and SPC-GNN [55].

## IV. EXPERIMENTAL SETUP

In this study, we utilize SBTM to generate balanced and imbalanced graphs to evaluate the proposed objective function. The maximum clique size  $M$  is set to 5 in each experiment. The balanced graph is set with  $N = [200, 200, 200, 200, 200]$ , meaning  $|I| = 5, |V| = 1000$ , and the number of nodes corresponding to each label is 200. The imbalanced graph is set with  $N = [300, 300, 100, 100, 100, 100]$ , thus  $|I| = 6$  and  $|V| = 1000$ . Additionally, the  $k$ -clique probability matrix  $B_k \in \mathbb{R}^{l^2}$  for  $k = 2, 3, 4, 5$  with different probability entries are chosen for both balanced and imbalanced networks. Detailed information is presented in Table I.

The diagonal and off-diagonal probability settings in  $(B_k)_k$  in both balanced and imbalanced graphs aim to investigate the relationship between performance improvement and variations in the number and distribution of higher-order simplices. The experimental setting ranges  $k$  from 2 to 5, allowing the performance of the proposed objective function to be tested across a wide range of higher-order clique distributions.

We define SI- $m$  (Simplicial Interactions up to order  $m$ ) as the experiment on the objective function utilizing up to  $K_m$  (the set of  $(m-1)$ -simplices in the network) for  $m = 3, 4, 5$ . More specifically, we define on the domain  $\Delta^n$  the objective

$$J_m = \sum_{k=2}^m w_k \sum_{(j_1, \dots, j_k) \in K_k} \sum_{(i_1, \dots, i_k) = \theta \in I^k} C_\theta p_{i_1}^{j_1} p_{i_2}^{j_2} \dots p_{i_k}^{j_k},$$

noting that the sum is over  $k = 2$  to  $m$ , thereby employing up to  $(m-1)$ -simplices in the graph. In particular, SI-2 utilizes only pairwise interactions through edges in the network, and the corresponding objective function becomes

$$J_2 = \sum_{(j_1, j_2) \in K_2} \sum_{(i_1, i_2) = \theta \in I^2} C_\theta p_{i_1}^{j_1} p_{i_2}^{j_2}.$$

Balanced Experiment, N = [200, 200, 200, 200, 200]			
	3-cliques Exp (SI-3)	4-cliques Exp (SI-4)	5-cliques Exp (SI-5)
Matrices used	$B_2, B_3$	$B_2, B_3, B_4$	$B_2, B_3, B_4, B_5$
Matrices	$(B_2)_{i,j} = \begin{cases} 0.01 & i = j \\ 0.001 & i \neq j \end{cases}$ $(B_3)_{i,j} = \begin{cases} 0.01 \text{ to } 0.05 & i = j \\ 0.001 \text{ to } 0.005 & i \neq j \end{cases}$	$(B_2)_{i,j} = \begin{cases} 0.01 & i = j \\ 0.001 & i \neq j \end{cases}$ $(B_3)_{i,j} = \begin{cases} 0.01 \text{ to } 0.05 & i = j \\ 0.001 \text{ to } 0.005 & i \neq j \end{cases}$ $(B_4)_{i,j} = \begin{cases} 0.05 \text{ to } 0.1 & i = j \\ 0.005 \text{ to } 0.01 & i \neq j \end{cases}$	$(B_2)_{i,j} = \begin{cases} 0.01 & i = j \\ 0.001 & i \neq j \end{cases}$ $(B_3)_{i,j} = \begin{cases} 0.01 \text{ to } 0.05 & i = j \\ 0.001 \text{ to } 0.005 & i \neq j \end{cases}$ $(B_4)_{i,j} = \begin{cases} 0.05 \text{ to } 0.1 & i = j \\ 0.005 \text{ to } 0.01 & i \neq j \end{cases}$ $(B_5)_{i,j} = \begin{cases} 0.12 \text{ to } 0.17 & i = j \\ 0.012 \text{ to } 0.017 & i \neq j \end{cases}$
Averaged total number of k-cliques, clique distributions of Intra-cluster, and number of inter-cluster cliques			
avg. # 1-cliques	1000, [200, 200, 200, 200, 200], 0	1000, [200, 200, 200, 200, 200], 0	1000, [200, 200, 200, 200, 200], 0
avg. # 2-cliques	2012, [300, 300, 300, 300, 300], 512	2279, [351, 351, 351, 351, 351], 524	2442, [380, 380, 380, 380, 380], 542
avg. # 3-cliques	219, [39, 39, 39, 39, 39], 24	370, [70, 70, 70, 70, 70], 20	536, [102, 102, 102, 102, 102], 26
avg. # 4-cliques	0, [0, 0, 0, 0, 0], 0	35, [7, 7, 7, 7, 7], 0	110, [22, 22, 22, 22, 22], 0
avg. # 5-cliques	0, [0, 0, 0, 0, 0], 0	0, [0, 0, 0, 0, 0], 0	15, [3, 3, 3, 3, 3], 0

Imbalanced Experiment, N = [300, 300, 100, 100, 100, 100]			
	3-cliques Exp (SI-3)	4-cliques Exp (SI-4)	5-cliques Exp (SI-5)
Matrices used	$B_2, B_3$	$B_2, B_3, B_4$	$B_2, B_3, B_4, B_5$
Matrices	$(B_2)_{i,j} = \begin{cases} 0.01 & i = j \\ 0.001 & i \neq j \end{cases}$ $(B_3)_{i,j} = \begin{cases} 0.03 \text{ to } 0.04 & i = j, i = 1,2 \\ 0.06 \text{ to } 0.08 & i = j, i = 3,4,5,6 \\ 0.006 \text{ to } 0.008 & i \neq j \end{cases}$	$(B_2)_{i,j} = \begin{cases} 0.01 & i = j \\ 0.001 & i \neq j \end{cases}$ $(B_3)_{i,j} = \begin{cases} 0.03 \text{ to } 0.04 & i = j, i = 1,2 \\ 0.06 \text{ to } 0.08 & i = j, i = 3,4,5,6 \\ 0.006 \text{ to } 0.008 & i \neq j \end{cases}$ $(B_4)_{i,j} = \begin{cases} 0.04 \text{ to } 0.07 & i = j, i = 1,2 \\ 0.08 \text{ to } 0.14 & i = j, i = 3,4,5,6 \\ 0.008 \text{ to } 0.014 & i \neq j \end{cases}$	$(B_2)_{i,j} = \begin{cases} 0.01 & i = j \\ 0.001 & i \neq j \end{cases}$ $(B_3)_{i,j} = \begin{cases} 0.03 \text{ to } 0.04 & i = j, i = 1,2 \\ 0.06 \text{ to } 0.08 & i = j, i = 3,4,5,6 \\ 0.006 \text{ to } 0.008 & i \neq j \end{cases}$ $(B_4)_{i,j} = \begin{cases} 0.04 \text{ to } 0.07 & i = j, i = 1,2 \\ 0.08 \text{ to } 0.14 & i = j, i = 3,4,5,6 \\ 0.008 \text{ to } 0.014 & i \neq j \end{cases}$ $(B_5)_{i,j} = \begin{cases} 0.11 \text{ to } 0.14 & i = j, i = 1,2 \\ 0.17 \text{ to } 0.23 & i = j, i = 3,4,5,6 \\ 0.017 \text{ to } 0.023 & i \neq j \end{cases}$
Averaged total number of k-cliques, clique distributions of Intra-cluster, and number of inter-cluster cliques			
avg. # 1-cliques	1000, [300, 300, 100, 100, 100, 100], 0	1000, [300, 300, 100, 100, 100, 100], 0	1000, [300, 300, 100, 100, 100, 100], 0
avg. # 2-cliques	2881, [821, 821, 153, 153, 153, 153], 627	3211, [899, 899, 195, 195, 195, 195], 633	3392, [1027, 1027, 236, 236, 236, 236], 594
avg. # 3-cliques	517, [144, 144, 38, 38, 38, 38], 77	764, [212, 212, 65, 65, 65, 65], 80	1260, [357, 357, 116, 116, 116, 116], 82
avg. # 4-cliques	0, [0, 0, 0, 0, 0, 0], 0	54, [15, 15, 6, 6, 6, 6], 0	261, [77, 77, 26, 26, 26, 26], 3
avg. # 5-cliques	0, [0, 0, 0, 0, 0, 0], 0	0, [0, 0, 0, 0, 0, 0], 0	40, [12, 12, 4, 4, 4, 4], 0

	Cora	Citeseer	Pubmed
The number of k-cliques, clique distributions of Intra-cluster, and number of inter-cluster cliques			
avg. # 1-cliques	2708, [351, 217, 418, 818, 426, 298, 180], 0	3327, [264, 590, 668, 701, 596, 508], 0	19717, [4103, 7739, 7875], 0
avg. # 2-cliques	5429, [534, 409, 827, 1175, 660, 417, 253], 1154	4732, [97, 452, 1041, 628, 689, 441], 1384	44324, [5212, 14563, 15790], 8773
avg. # 3-cliques	1630, [144, 163, 309, 350, 219, 96, 82], 267	1167, [17, 121, 490, 117, 116, 54], 252	12520, [1176, 3098, 5295], 2951
avg. # 4-cliques	220, [15, 27, 32, 60, 30, 8, 13], 35	255, [1, 48, 164, 9, 5, 4], 24	3275, [327, 1052, 1205], 691
avg. # 5-cliques	9, [0, 2, 0, 4, 0, 0, 0], 3	46, [0, 15, 30, 0, 0, 0], 1	859, [88, 501, 139], 131
avg. # 6-cliques	0, [0, 0, 0, 0, 0, 0, 0], 0	4, [0, 2, 2, 0, 0, 0], 0	217, [17, 157, 6], 37
avg. # 7-cliques	0, [0, 0, 0, 0, 0, 0, 0], 0	0, [0, 0, 0, 0, 0, 0, 0], 0	31, [2, 20, 0], 9
avg. # 8-cliques	0, [0, 0, 0, 0, 0, 0, 0], 0	0, [0, 0, 0, 0, 0, 0, 0], 0	1, [0, 0, 0], 1

TABLE I

BALANCED AND IMBALANCED SI-3, SI-4, SI-5 EXPERIMENT SETTINGS, BENCHMARK DATA, AND CLIQUE DISTRIBUTIONS IN EACH EXPERIMENT.



## Accuracy Gains for Balanced Experiments

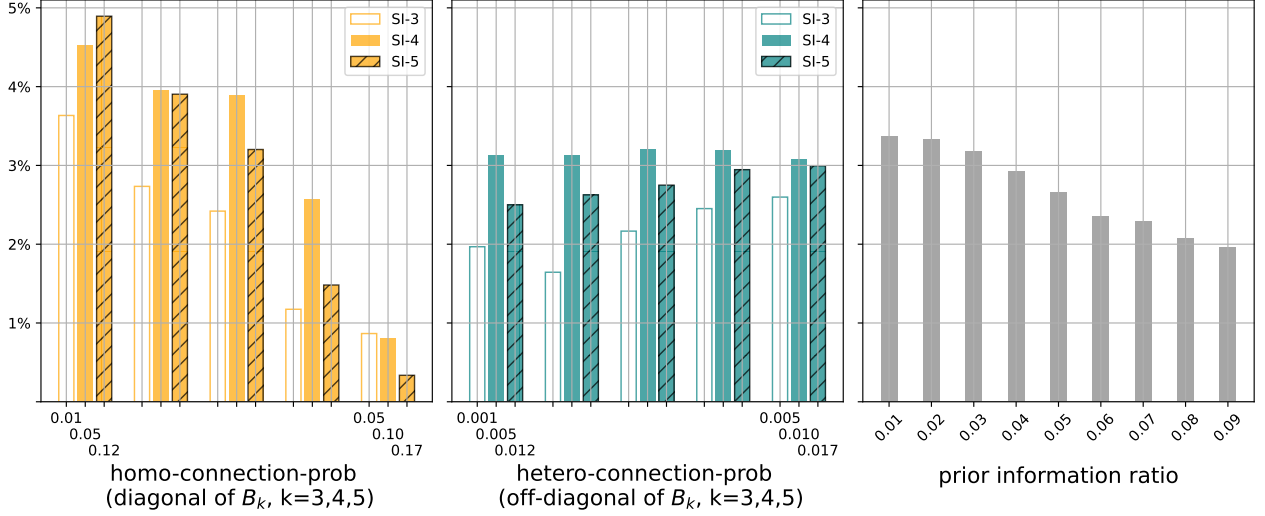


Fig. 2. Accuracy gains for balanced SI-3, SI-4 and SI-5 experiments with respect to variations in homo-connection probability (left), hetero-connection probability (middle), and prior information ratio in  $N=[200,200,200,200,200]$  setting. In left, diagonal of  $B_3$ ,  $B_4$ , and  $B_5$  varies from 0.01 to 0.05, from 0.05 to 0.10, and from 0.12 to 0.17, respectively. In middle, off-diagonal of  $B_3$ ,  $B_4$ , and  $B_5$  varies from 0.001 to 0.005, from 0.005 to 0.010, and from 0.012 to 0.017, respectively. In right, the performances are tested ranging from 10 known nodes (= prior information ratio 0.01) to 90 known nodes (PIR 0.09).

The experimental results show that the performances of RW and SI-2 are comparable. Because of this, the average performance of these two methods in experiments that do not utilize higher-order networks will be presented as PI (Pairwise Interactions). In this semi-supervised learning research, the prior information ratio (the proportion of nodes whose labels are revealed) is varied from 1% to 9%, and nodes corresponding to the given prior information ratio are randomly sampled for each cluster and used as label-aware nodes. The results are evaluated for the performance gains achieved by SI- $M$  (where  $M = \omega(G)$ ) over the PI for each prior information ratio. In this study, the accuracy gain is defined as the relative performance improvement by  $y/x - 1$  where  $x$  and  $y$  represents the accuracies obtained from PI and SI- $M$ , respectively.

We adopt a transductive learning framework using three well-known citation network datasets: Cora, Citeseer, and Pubmed, as outlined by [56] and following the methodology of [40] and [22]. These datasets include the Cora dataset with 2708 nodes, 5429 undirected citation edges, and 7 classes; the Citeseer dataset comprises 3327 nodes, 4732 edges, and 6 classes; and the Pubmed dataset with 19717 nodes, 44338 edges, and 3 classes. In our experimental setup, documents are represented as nodes, and citations are represented as edges. The distribution of simplices in the benchmark dataset is illustrated in Table I.

In the training process, Glorot initialization [57] is utilized to initialize the parameters, and the Adam SGD optimizer [58] is employed for optimization. For all experiments, the learning rate is set to 0.4 and the number of epochs to 10. The proposed objective function aims to learn the probability distribution assigned to each node in the network. We apply the softmax function to describe node probability distributions. At the end of the training process, the argmax function is used to obtain the final classification results. The performance is evaluated

using the accuracy error metric.

In this study, we introduce a simple and concise learning process that assigns an  $l$ -dimensional probability distribution to each node in a given network, categorizes the network's simplices by their size, and integrates them into objective function (4) for training (Figure 1). Specifically, for a fixed weight hyperparameter  $w_k$  in (4), this research utilizes only the learning rate and epochs as hyperparameters. As a result, the proposed algorithm (which does not rely on node embedding dimensions, number of layers in the neural network architecture, number of units per layer, dropout rates, or activation functions as hyperparameters) offers enhanced implementation efficiency compared to the existing GNN architectures that demand extensive hyperparameter tuning. Leveraging this efficiency, we evaluate the proposed objective function exclusively with training data (label-aware nodes) and test data (label-unaware nodes) in our experiments involving both synthetic graphs and real citation networks (Cora, Citeseer, Pubmed), without the use of validation nodes.

## V. RESULTS

In this section, we evaluate the node classification performance of the proposed higher-order networks based objective function on balanced data (Section V-A), and imbalanced data (Section V-B). In both sections, we primarily evaluate the accuracy gain achieved by using the proposed objective function with higher-order structures, compared to the performance of PI (mean performance of RW and SI-2). Furthermore, we discuss how the objective function can be combined with various semi-supervised node classification methodologies to achieve additional performance gains (Section V-C). Result summary and discussion is presented in Section V-D. The implementation algorithm can be found at [https://github.com/kooeunho/HOI\\_objective](https://github.com/kooeunho/HOI_objective).

## Accuracy Gains for Imbalanced Experiments

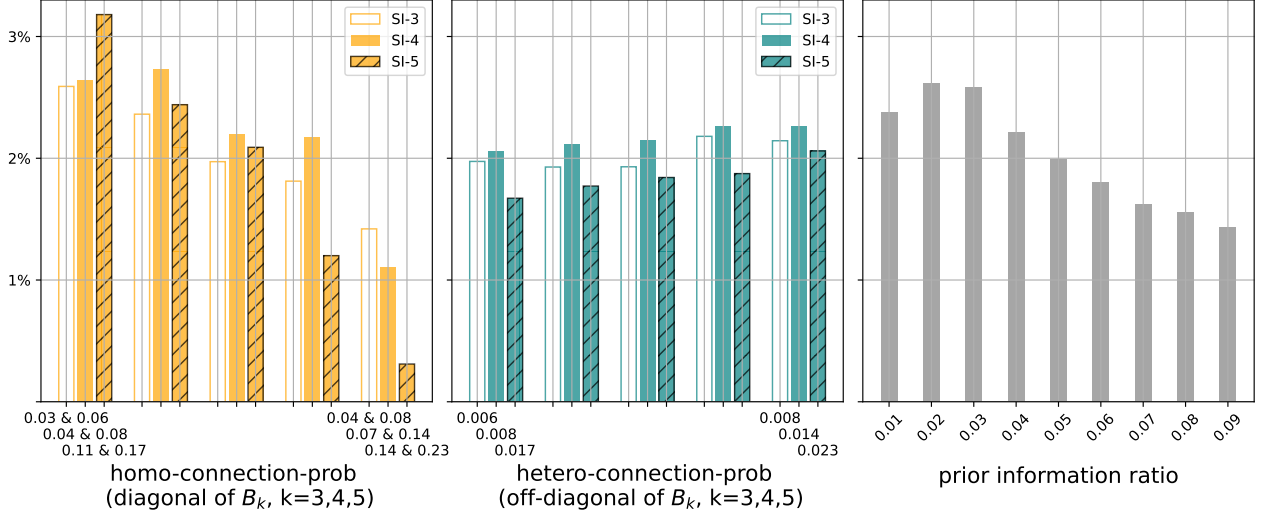


Fig. 3. Accuracy gains for imbalanced SI-3, SI-4 and SI-5 experiments with respect to variations in homo-connection probability (left), hetero-connection probability (middle), and prior information ratio in  $N=[300,300,100,100,100,100]$  setting. In left, diagonal of  $B_3$ ,  $B_4$  and  $B_5$  varies from 0.03 & 0.06 to 0.04 & 0.08 (that is, the diagonal entries of  $B_3$  varies from [0.03, 0.03, 0.06, 0.06, 0.06, 0.06] to [0.04, 0.04, 0.08, 0.08, 0.08, 0.08]), from 0.04 & 0.08 to 0.07 & 0.14, and from 0.11 & 0.17 to 0.14 & 0.23, respectively. In middle, off-diagonal of  $B_3$ ,  $B_4$  and  $B_5$  varies from 0.006 to 0.008, from 0.008 to 0.014, and from 0.017 to 0.023. In right, the performances are tested ranging from 10 known nodes (= prior information ratio 0.01) to 90 known nodes (PIR 0.09).

### A. Balanced experiment

We present the experimental results based on the balanced experiment settings depicted in Table I. Three experiment settings (SI-3, SI-4 and SI-5) are designed to generate higher-order simplices encompassing 3-cliques, 4-cliques, and 5-cliques, respectively. We compare the accuracy gains of our proposed objective function, which utilizes the higher-order simplices distributed across the network, with PI. Figure 2 illustrates the accuracy gains of our strategy relative to the PI with respect to the homo-connection probability (diagonals of  $B_k$  for each  $k = 3, 4, 5$ ), the hetero-connection probability (off-diagonals of  $B_k$  for each  $k = 3, 4, 5$ ), and the prior information ratio. The weight hyperparameter  $w_k$  is set to 1 in balanced experiments. The experimental results show that the accuracy gains for the SI-3, SI-4 and SI-5 experiments are 2.17%, 3.15%, and 2.76%, respectively. This shows that the proposed objective function enhances performance by utilizing higher-dimensional simplices such as triangles, tetrahedrons and pentachorons. Furthermore, according to Figure 2, greater accuracy increases occur with lower homo-connection probabilities, higher hetero-connection probabilities, and lower prior information ratios. This implies that the objective function can achieve significant additional performance gains by exploiting higher-order structures in challenging scenarios where the network structure is ambiguous and prior information is limited.

### B. Imbalanced experiment

In the imbalanced experiments, the parameters of the SBTM (that is,  $B_k$  for  $k = 3, 4, 5$ ) differ from those in the balanced experiments, as shown in Table I. We adjust these parameters because our objective function aims to enhance node classification performance by leveraging the higher-order structures

in the network, and a particular cluster containing too few of these structures would not allow for an accurate evaluation of the proposed function. The resulting distribution of  $k$ -cliques in each cluster corresponding to the set parameters is presented in Table I. The experimental results show that the accuracy gains for the SI-3, SI-4 and SI-5 experiments against PI are 2.03%, 2.17%, and 1.84%, respectively. This result implies that higher-order structures have a significant impact on node classification performance as in the balanced experiments. Additionally, the characteristic of the proposed objective function to achieve more performance gains in challenging scenarios is also evident in the imbalanced settings, as seen in Figure 3.

The weight hyperparameter  $w_k$  on  $k$ -cliques in the objective function (4) is designed to emphasize higher-order simplices over lower-order ones. In the imbalanced experiments, additional tests employing  $w_k = \alpha^{k-1}$  (with  $\alpha = 1, 2, 3, 4, 5$ ) on SI-5 experiments (Figure 4) show that mean accuracy gains for SI-5 against PI are 2.02%, 2.35%, 2.49%, 2.55%, and 2.58%, for  $\alpha = 1, 2, 3, 4$  and 5, respectively. Greater weights on higher-order simplices lead to enhanced performance; however, incremental benefits diminish as  $\alpha$  increases. Employing the weighting strategy that favors higher-order cliques is suggested to potentially improve node classification in imbalanced networks, including those with long-tail distributions.

### C. Integration of the objective function with GNN

In this subsection, we propose a method to enhance node classification performance by integrating the objective function proposed in this study with existing GNN-based node classification methods, which do not directly utilize higher-order structures in training, including the latest semi-supervised and unsupervised learning techniques. Existing methodologies employ various strategies such as integrating random walks



### Accuracy Gains on $w_k = \alpha^{k-1}$ for Imbalanced SI-5 Experiments

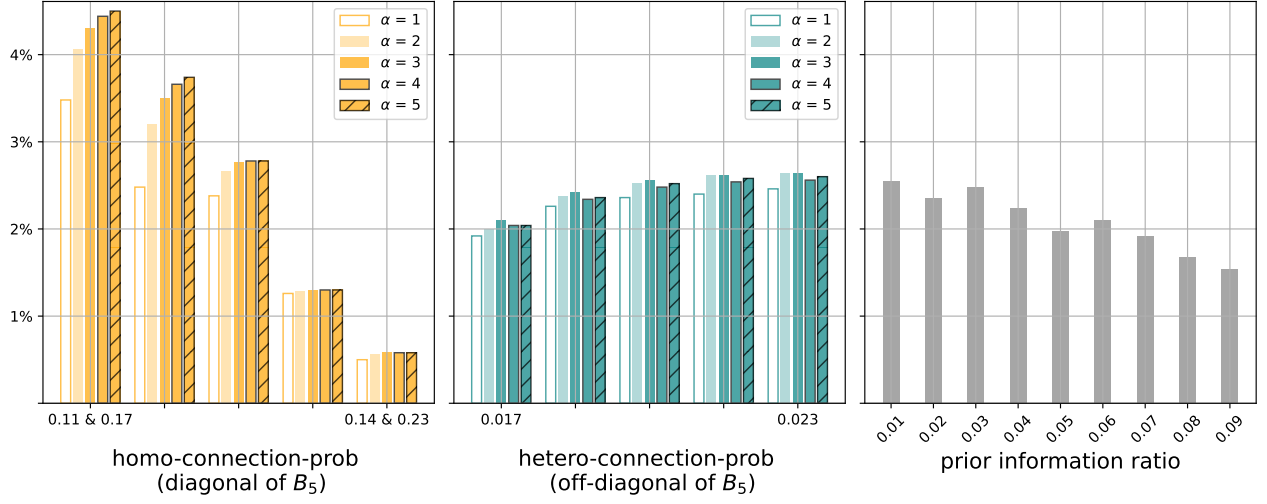


Fig. 4. Accuracy gains on  $w_k = \alpha^{k-1}$  in the objective (4) for imbalanced SI-5 experiments with respect to variations in homo-connection probability (left), hetero-connection probability (middle), and prior information ratio in  $N=[300,300,100,100,100,100]$  setting. In left, diagonal of  $B_5$  varies from [0.11, 0.11, 0.17, 0.17, 0.17, 0.17] to [0.14, 0.14, 0.23, 0.23, 0.23, 0.23]. In middle, off-diagonal of  $B_5$  varies from 0.017 to 0.023. In right, the performances are tested ranging from 10 known nodes (= prior information ratio 0.01) to 90 known nodes (PIR 0.09).

with the Word2Vec model [38], adjusting exploration and return variables of random walks [41], leveraging second-order proximities between nodes [39], utilizing 3-motifs for distinguishing high-order graph structures [50], or utilizing attention mechanisms [22] to train embedding vectors for all nodes in the network and ultimately validate classification performance based on the output layer using simple neural network structures like linear maps. To integrate our approach with these, we apply the softmax function to the  $l$ -dimensional output vector (where  $l = |I|$  is the number of clusters or labels) of the existing architecture to create a probability distribution for each node. We then set these distributions as the initial parameters and use the proposed objective function to retrain. In other words, we use the output of the GNN architectures via softmax function instead of the RW to initialize our proposed optimization method (5). This strategy allows us to leverage ( $l$ -dimensional) classification vectors resulting from the trained embedding vectors of GNN-based architectures and use these classification vectors to learn the various higher-order structures included in the network using the proposed objective function, potentially improving classification performance.

In this experiment, we integrate GAT [22] with the proposed objective function and evaluate the performance gains using the Cora, Citeseer, and Pubmed datasets. GAT uses an attention mechanism to learn node embeddings. The node features are created using the bag-of-words representation of documents, with the dimensions of the node features for Cora, Citeseer, and Pubmed being 1433, 3703, and 500, respectively. These features are used to learn embedding vectors using a multi-head attention structure, and a linear map is employed to generate outputs corresponding to the number of clusters. In our method, the total training and validation data used in [22]—640 for Cora, 620 for Citeseer, and 560 for Pubmed—are treated as the overall prior information. In

Method	Cora	Citeseer	Pubmed
GAT	83.0 %	72.5 %	79.0 %
GAT + Higher-order training	84.9 %	72.9 %	80.1 %

TABLE II  
SUMMARY OF ACCURACY RESULTS FOR BENCHMARK DATASETS.

contrast to [22], which evaluated performance using 1,000 test nodes for each dataset, we assess performance on all remaining nodes not designated for training or validation. According to the results of integration experiments, the accuracies using training and validation nodes in the GAT study as prior information (label-aware nodes) are 84.9%, 72.9%, and 80.1% for Cora, Citeseer, Pubmed dataset respectively (Table II).

#### D. Summary and discussion

In this paper, we examined the performance of the proposed objective function (4) in a variety of contexts, including balanced, imbalanced, and benchmark datasets. Several notable experimental results and discussions are listed below.

First, SBTM is a graph generation model that enables control over the distribution of  $k$ -cliques in the created network. Each element  $(T_k)_\theta$  of a  $k$ -tensor  $T_k \in \mathbb{R}^{l^k}$  reflects the probability of creating a  $k$ -clique given the label distribution  $\theta$ . We generated  $T_k$  systematically from a  $l \times l$  matrices  $B_k$  using the formula (6) for each  $k \geq 2$ . The motivation is that by modifying the entries of  $B_k$ , one can effectively mimic networks with clique distributions that reflect real networks.

Second, it is clear that nodes corresponding to each label become more distinguishable in node classification tasks as the homo-connection probability increases, the hetero-connection

probability decreases, and the prior information ratio rises, facilitating community detection. Given this, it is interesting that the proposed objective function showed significant performance gains in the opposite circumstance (lower diagonal, larger off-diagonal in  $B_k$ , and lower prior information ratio). This tendency is consistent across both balanced and imbalanced studies, showing that the suggested objective function can be applied in challenging classification scenarios, such as recognizing overlapping communities.

Third, it is discovered that in the balanced PI experiment, the nodes representing incorrect predictions are uniformly distributed across all labels. In contrast, in the imbalanced PI experiment, nodes corresponding to incorrect predictions are weighted toward label indices with more nodes than other labels. Furthermore, experimental results show that the weight parameter  $w_k = \alpha^{k-1}$  applied to  $k$ -cliques in the objective function (4) yields that as  $\alpha$  increases (meaning greater penalties are applied to higher-order simplices), additional performance gains are more pronounced in imbalanced networks than balanced ones. This suggests that the performance gain obtained with the proposed objective function, which is attributed to correcting the error distribution by using higher-order simplices in the training process, combined with the strategy of imposing larger penalties on higher-order simplices, could be more effective in imbalanced scenarios.

Fourth, the learning strategy used in this study does not rely on the many hyperparameters typically used in GNN-based algorithms, such as node embedding dimensions, the depth of the neural network architecture, the number of units in each layer, dropout ratios, or activation mechanisms. Additionally, experimental results on both synthetic networks and real citation networks have shown that the standard deviation of the results from each experiment using the proposed algorithm is negligible. This demonstrates that the proposed algorithm offers advantages in terms of consistency of results, stability, simplicity, and generalization performance.

Fifth, as described in the objective (4), we use a strategy where we first aggregate all  $k$ -cliques in a given network for each  $k$ , calculate the errors, and then repeat for all  $k = 2, 3, \dots, M$ . After that, we combine all of the errors and minimize the outcome. The multinomial coefficients ( $C_\theta$  in (4)) corresponding to the  $k$ -cliques differ for each  $k$ , making it difficult to apply a uniform coefficient across different  $k$  values. This variability prevents standardizing the method for extracting the characteristics of cliques of different sizes in the network into a single method. As a result, this characteristic of the objective function acts as a barrier to parallelization. We suggest that future research should focus on finding an effective method for gathering higher-order simplices in the network to enable parallel computing.

Sixth, according to experimental results with benchmark data, using the training and validation data employed in GAT research [22] as prior information improves accuracy by 1.9%, 0.4%, and 1.1% for Cora, Citeseer, and Pubmed, respectively, compared to the result in [22]. The objective function used in this study is intended to promote that all nodes forming higher-order simplices exhibit similar node probability distributions. As a result, if the network has a large number of inter-

simplices, where nodes producing higher-order simplices belong to distinct labels or clusters, achieving good performance with the proposed objective function becomes challenging. The proportions of inter-cluster simplices among all higher-order simplices beyond 2-simplices are 16.4%, 18.8%, and 22.6% (therefore, intra-cluster simplices ratio would be 83.6%, 81.2%, 77.4%) for Cora, Citeseer, and Pubmed, respectively. This observation explains why a higher accuracy improvement is observed with Cora compared to Citeseer and Pubmed.

Finally, this study demonstrated that the combination of GAT with the proposed objective function improved classification performance on benchmark data. While numerous recent advanced methodologies are being developed in the field of semi-supervised learning, almost all of them do not directly address higher-order simplices. If the given data supports the hypothesis of this study—densely connected nodes tend to exhibit similar attributes—it is speculated that further performance gains could be achieved by utilizing combinations with state-of-the-art approaches such as MGNN and GAT.

## VI. CONCLUSION

In this paper, we propose a probability-based objective function for semi-supervised node classification that takes advantage of simplicial interactions of varying order. Given that densely connected nodes are likely to have similar properties, our proposed objective function imposes a greater penalty when nodes connected via higher-order simplices have diversified labels. For a given number of distinct labels  $l$ , each node is equipped with an  $l$ -dimensional probability distribution, and we seek the distribution across all nodes that minimizes the objective function under the constraint that the sum of node probabilities is one. Furthermore, based on the recognition that traditional stochastic block models do not adequately mimic many real datasets, particularly in representing the distribution of higher-order simplices within each cluster, we propose the stochastic block tensor model (SBTM). For each  $k \geq 2$ , the SBTM uses probability parameters to control the number of  $k$ -cliques within or between clusters, adjusting the distribution of higher-order simplices in the network, thus better reflecting real data characteristics. The evaluation of our proposed function was conducted using graphs generated by the stochastic (SBTM) and in integration with graph neural network-based architectures (GAT). In challenging classification scenarios, where the probability of connections within the same label is low, the probability of connections between different labels is high, and there are fewer nodes with known labels, our proposed function integrating higher-order networks outperformed results from simple pairwise interactions or random walk-based probabilistic methods. Especially in imbalanced data, by adjusting the weight parameter within the objective function, further accuracy gains were achieved when the distribution of misclassified nodes was biased towards certain label indices containing more nodes than other labels. This offers potential applications in many node classification problems in network data, when combined with several semi-supervised studies that do not directly use higher-order simplices dispersed in networks. Our suggested objective function,

which conducts different calculations depending on the size of the simplices, confronts a computational challenge because uniform operations cannot be applied to all simplices, making GPU-based parallel computing approaches difficult to deploy. Overcoming this limitation is proposed for future research.

#### ACKNOWLEDGMENTS

Eunho Koo gratefully acknowledges the support of the Korea Institute for Advanced Study (KIAS) under the individual grant AP086801. Tongseok Lim wishes to express gratitude to the KIAS AI research group and director Hyeon, Changbong for their hospitality and support during his stay at KIAS in 2023 and 2024, where parts of this work were performed.

#### REFERENCES

- [1] S. Fortunato and D. Hric, “Community detection in networks: A user guide,” *Physics reports*, vol. 659, pp. 1–44, 2016.
- [2] M. T. Schaub, J.-C. Delvenne, M. Rosvall, and R. Lambiotte, “The many facets of community detection in complex networks,” *Applied network science*, vol. 2, no. 1, pp. 1–13, 2017.
- [3] D. Easley, J. Kleinberg *et al.*, *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge university press Cambridge, 2010, vol. 1.
- [4] S. Klamt, U.-U. Haus, and F. Theis, “Hypergraphs and cellular networks,” *PLoS computational biology*, vol. 5, no. 5, p. e1000385, 2009.
- [5] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri, “Networks beyond pairwise interactions: Structure and dynamics,” *Physics Reports*, vol. 874, pp. 1–92, 2020.
- [6] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps,” *Social networks*, vol. 5, no. 2, pp. 109–137, 1983.
- [7] D. Ghoshdastidar and A. Dukkipati, “Consistency of spectral partitioning of uniform hypergraphs under planted partition model,” *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [8] C. Kim, A. S. Bandeira, and M. X. Goemans, “Community detection in hypergraphs, spiked tensor models, and sum-of-squares,” in *2017 International Conference on Sampling Theory and Applications (SampTA)*. IEEE, 2017, pp. 124–128.
- [9] B. Karrer and M. E. Newman, “Stochastic blockmodels and community structure in networks,” *Physical review E*, vol. 83, no. 1, p. 016107, 2011.
- [10] E. M. Airoldi, D. Blei, S. Fienberg, and E. Xing, “Mixed membership stochastic blockmodels,” *Advances in neural information processing systems*, vol. 21, 2008.
- [11] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.
- [12] M. E. Newman, “Modularity and community structure in networks,” *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [13] A. Vazquez, “Finding hypergraph communities: a bayesian approach and variational solution,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, no. 07, p. P07006, 2009.
- [14] I. Chien, C.-Y. Lin, and I.-H. Wang, “Community detection in hypergraphs: Optimal statistical limit and efficient algorithms,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 871–879.
- [15] D. Ghoshdastidar and A. Dukkipati, “Consistency of spectral hypergraph partitioning under planted partition model,” 2017.
- [16] E. Eaton and R. Mansbach, “A spin-glass model for semi-supervised community detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, 2012, pp. 900–906.
- [17] X. Ma, L. Gao, X. Yong, and L. Fu, “Semi-supervised clustering algorithm for community structure detection in complex networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 1, pp. 187–197, 2010.
- [18] D. Liu, X. Liu, W. Wang, and H. Bai, “Semi-supervised community detection based on discrete potential theory,” *Physica A: Statistical Mechanics and its Applications*, vol. 416, pp. 173–182, 2014.
- [19] K. Nowicki and T. A. B. Snijders, “Estimation and prediction for stochastic blockstructures,” *Journal of the American statistical association*, vol. 96, no. 455, pp. 1077–1087, 2001.
- [20] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International Conference on Learning Representations*, 2017.
- [21] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Advances in neural information processing systems*, vol. 30, 2017.
- [22] P. Velićković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” *International Conference on Learning Representations*, 2018.
- [23] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, “Representation learning on graphs with jumping knowledge networks,” in *International conference on machine learning*. PMLR, 2018, pp. 5453–5462.
- [24] E. Bendito, A. Carmona, and A. M. Encinas, “Solving dirichlet and poisson problems on graphs by means of equilibrium measures,” *European Journal of Combinatorics*, vol. 24, no. 4, pp. 365–375, 2003.
- [25] C. Bick, E. Gross, H. A. Harrington, and M. T. Schaub, “What are higher-order networks?” *SIAM Review*, vol. 65, no. 3, pp. 686–731, 2023.
- [26] L.-H. Lim, “Hodge laplacians on graphs,” *Siam Review*, vol. 62, no. 3, pp. 685–715, 2020.
- [27] S. Paul, O. Milenkovic, and Y. Chen, “Higher-order spectral clustering under superimposed stochastic block models,” *Journal of Machine Learning Research*, vol. 24, no. 320, pp. 1–58, 2023.
- [28] Z. Song, X. Yang, Z. Xu, and I. King, “Graph-based semi-supervised learning: A comprehensive review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8174–8194, 2022.
- [29] X. Zhu, Z. Ghahramani, and J. D. Lafferty, “Semi-supervised learning using gaussian fields and harmonic functions,” in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.
- [30] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, “Learning with local and global consistency,” *Advances in neural information processing systems*, vol. 16, 2003.
- [31] D. Zhou, J. Huang, and B. Schölkopf, “Learning from labeled and unlabeled data on a directed graph,” in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 1036–1043.
- [32] F. Wang and C. Zhang, “Label propagation through linear neighborhoods,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 985–992.
- [33] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” *Journal of machine learning research*, vol. 7, no. 11, 2006.
- [34] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [35] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” *Advances in neural information processing systems*, vol. 14, 2001.
- [36] S. Cao, W. Lu, and Q. Xu, “Grarep: Learning graph representations with global structural information,” in *Proceedings of the 24th ACM international conference on information and knowledge management*, 2015, pp. 891–900.
- [37] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, “Asymmetric transitivity preserving graph embedding,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1105–1114.
- [38] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.
- [39] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding,” in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.
- [40] Z. Yang, W. Cohen, and R. Salakhudinov, “Revisiting semi-supervised learning with graph embeddings,” in *International conference on machine learning*. PMLR, 2016, pp. 40–48.
- [41] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016.
- [42] X. Hong, T. Zhang, Z. Cui, and J. Yang, “Variational gridded graph convolution network for node classification,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 10, pp. 1697–1708, 2021.
- [43] D. Wang, P. Cui, and W. Zhu, “Structural deep network embedding,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

- [44] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.
- [45] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *NIPS Workshop on Bayesian Deep Learning*, 2016.
- [46] K. Wang, J. An, M. Zhou, Z. Shi, X. Shi, and Q. Kang, "Minority-weighted graph neural network for imbalanced node classification in social networks of internet of people," *IEEE Internet of Things Journal*, vol. 10, no. 1, pp. 330–340, 2022.
- [47] 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.
- [48] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," *Advances in neural information processing systems*, vol. 30, 2017.
- [49] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97. PMLR, 2019.
- [50] X. Chen, R. Cai, Y. Fang, M. Wu, Z. Li, and Z. Hao, "Motif graph neural network," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [51] L. Wang, W. Yu, W. Wang, W. Cheng, W. Zhang, H. Zha, X. He, and H. Chen, "Learning robust representations with graph denoising policy network," in *IEEE International Conference on Data Mining*, 2019.
- [52] F. Sun, J. Hoffmann, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *8th International Conference on Learning Representations, ICLR 2020*.
- [53] C. Zheng, B. Zong, W. Cheng, D. Song, J. Ni, W. Yu, H. Chen, and W. Wang, "Robust graph representation learning via neural sparsification," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.
- [54] D. Luo, W. Cheng, W. Yu, B. Zong, J. Ni, H. Chen, and X. Zhang, "Learning to drop: Robust graph neural network via topological denoising," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021.
- [55] M. Gong, H. Zhou, A. K. Qin, W. Liu, and Z. Zhao, "Self-paced co-training of graph neural networks for semi-supervised node classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 9234–9247, 2022.
- [56] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.
- [57] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings*, 2010, pp. 249–256.
- [58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, 2015.