MAXIMAL CLIQUE INTERACTION FOR ENHANCED NODE CLASSIFICATION IN NETWORKS

EUNHO KOO AND TONGSEOK LIM

Abstract. In network analysis with higher-order interactions, utilizing all of the cliques in the network appears natural and intuitive. However, this strategy frequently experiences computational inefficiencies due to overlapping information in both higher-order and lower-order cliques. This paper describes and validates a strategy based on the maximal cliques for semi-supervised node classification tasks that takes advantage of higher-order network structure. The findings indicate that the maximal clique approach performs similarly while training on significantly fewer cliques than the all-cliques strategy, and furthermore, maximal cliques outperform pairwise interactions in both balanced and imbalanced networks. This implies that the network structure can be adequately extracted using only maximal cliques, despite utilizing significantly fewer cliques than the all-cliques approach, resulting in substantial computational reduction.

Index terms: Node classification, semi-supervised, simplex, clique, maximal clique, node interaction, higher order networks, hypergraph, probabilistic objective function.

1. Introduction

Systems have been analyzed using various methodologies, often represented as graphs, which involve identifying important nodes or hubs within the graph [Fri91, Gle15]. However, many real-world networks exhibit phenomena that cannot be fully explained by traditional graph-based approaches, which typically focus solely on pairwise interactions, i.e., interactions between two nodes connected by an edge. For instance, economic activities rely on agreements among buyers, sellers, and intermediaries [BHJ04], while relationships such as friendships [EK+12] and large-scale online networks [FIA11] involve

Eunho Koo: Department of big data convergence, Chonnam National University, Gwangju, Republic of Korea. E-mail address: kooeunho@jnu.ac.kr

Tongseok Lim: Mitchell E. Daniels, Jr. School of Business, Purdue University, West Lafayette, Indiana 47907, USA. E-mail address: lim336@purdue.edu

Date: March 10, 2024.

The authors all contributed equally to the research. Corresponding author: Tongseok Lim.

interactions among multiple nodes. Therefore, there is a need to develop methods that extend beyond pairwise interactions to understand the mechanisms of complex networks.

A notable approach to addressing the perspective of network analysis involves the application of higher-order network methodologies, which are based on a probability framework derived from the Stochastic Block Model (SBM) [HLL83, GD14, LML$^+$17, KBG17]. SBM, functioning as a generative model, enables the construction of graphs that exhibit community structures within distinct node communities. This model leverages a comprehensive set of nodes, partitioned into disjoint subsets covering all nodes, along with the probabilities of connections between nodes within each subset, serving as its foundational parameters. SBM's utility extends to the exploration and elucidation of a network's inherent structures, as well as to the facilitation of clustering endeavors [LW19]. Meanwhile, in the domain of traditional supervised learning methodologies, the necessity for extensive labeled datasets for classification tasks poses a significant challenge due to the difficulty of acquiring such data. To address the labor-intensive process of data labeling, the field of semi-supervised learning has emerged as a promising alternative [Zhu05]. This area is actively researched [MP07, BSS$^+$08, TRP$^+$08], typically involving the strategic selection of a minimal subset of nodes to serve as prior knowledge in node classification tasks. Empirical evidence suggests that even a limited amount of prior information can significantly enhance prediction performance [EM12, VSGA11, MGYF10, AVSG10, Zha13, ZSW13].

Recent advancements have introduced methodologies to leverage the capabilities of higher-order interactions [BCI$^+$20, BGHS23, KG24]. Specifically, in the context of semi-supervised node classification, [KL23] suggested utilizing node interactions within each higher-order clique in the network. Throughout the optimization phase, the clique interaction-based optimization scheme in their work enforces a requirement for nodal entities within a given clique to exhibit similar probability distributions, based on the hypothesis that nodes with dense interconnections are likely to share similar distribution characteristics. Consequently, nodes situated within higher-order cliques incur a greater penalty for distributional diversity compared to their counterparts in lower-order cliques. This methodology has demonstrated superior performance in classification tasks over traditional non-hypergraph approaches, which rely solely on pairwise interactions. However, the comprehensive nature of higher-order cliques, which integrate multiple lower-order cliques, renders the utilization of all cliques for learning purposes inefficient. As a result, it appears clear that a meticulous choice of a subset of cliques is required.

In response to this challenge, we propose a maximal clique strategy, wherein training processes exclusively employ maximal cliques rather than the entire clique spectrum. A maximal clique is defined as a clique that cannot be expanded through the inclusion of an adjacent node, and therefore does not reside within a larger clique. We derive the expected number of maximal cliques in networks generated by the Planted Partition Model (PPM), a special case of SBM, and examine the fluctuation of this metric in response to increases in node numbers and nodal connection probabilities (see Proposition A.1 and Figure 2). Moreover, we conduct a comparative analysis of prediction performances between strategies that utilize all cliques versus those that employ only maximal cliques.

Our contributions are outlined as follows:

• We derive the expected number of maximal cliques (as well as general cliques) in the PPM, enabling us to analyze the distribution of higher-order cliques as the number of nodes and connection probability increase. This provides quantitative evidence that the classification strategy using maximal cliques in this paper reduces computational complexity significantly compared to the strategy using all cliques considered in [KL23].

• We evaluate the comparative prediction performance of the full clique strategy and the maximal clique strategy, as well as comparing them with random walk and pairwise interaction-based strategies. Our experimental results indicate that the maximal clique strategy demonstrates comparable classification performance with the full clique strategy, while outperforming the strategies based on random walk and pairwise interaction.

This paper is structured as follows. Preliminaries are given in Section 2. The maximal clique strategy is described in Section 3. The experimental setup is detailed in Section 4. The results are discussed in Section 5. The conclusion is presented in Section 6.

2. Preliminaries

In this section, we give basic definitions and mathematical concepts used in this study.

2.1. **Hypergraphs.** Conventional graph analysis primarily focuses on pairwise interactions between two nodes. The objective of this paper is to propose an efficient strategy capable of leveraging the hypergraph structure inherent in the underlying graph. A hypergraph is a generalization of the traditional graph in which an edge can contain more than two nodes [BGHS23]. We explain notations on undirected hypergraphs used in this study. An undirected graph $G = (V, E)$ consists of a node set $V = \{1, 2, ..., N\}$, and an edge set $E \subseteq \{(i, j) \,|\, i, j \in V,\, i \neq j\}$ where $(i, j) = (j, i)$. A hypergraph generalizes

E as $\mathcal{E} \subseteq 2^V$ where 2^V denotes the power set of V, and we denote the hypergraph as $\mathcal{H} = (V, \mathcal{E})$. For a positive integer k, we define $\sigma = \{n_1, n_2, \ldots, n_k\} \in \mathcal{E}$ as a k-clique if $(n_i, n_j) \in E$ for every $1 \leq i < j \leq k$. Let K_k denote the set of all k-cliques in G, such that K_1, K_2, K_3, and K_4 represent the sets of nodes, edges, triangles, and tetrahedra in G, respectively. The collection of all k-cliques in G, denoted by $K \equiv \bigcup_{k=1}^{M} K_k$, is referred to as the clique complex of the graph G, where M represents the size of the largest clique in G. We interpret $\mathcal{E}$ as a subset of K in the context of a hypergraph.

2.2. **Planted Partition Model.** The stochastic block model was initially introduced in the field of social networks [HLL83]. In this model, the node set is partitioned into disjoint nonempty subsets C_i for $i = 1, 2, \ldots, l$, such that $V = \bigcup_{i=1}^{l} C_i$ and $C_i \cap C_j = \emptyset$ if $i \neq j$. Additionally, there exists a symmetric $l \times l$ matrix of edge probabilities, where the connection probability between two nodes u and v is the (i, j) component of the $l \times l$ matrix for all $u \in C_i$ and $v \in C_j$. The Planted Partition Model (PPM) represents a special case of the stochastic block matrix, where the entries of the $l \times l$ matrix are constant p on the diagonal and q off the diagonal, with $p > q$. Adaptive algorithms that determine optimal block configurations and recognize diverse interaction patterns between blocks have broadened the functionalities of PPM [Abb18].

2.3. **Node classification algorithm based on random walk on graphs.** In semi-supervised node classification tasks, a widely used algorithm based on random walks on a graph operates as follows. An unbiased random walk transitions from node $i \in V$ to node $j \in V$ with probability $1/d$ if $(i, j) \in E$ (i.e., i and j are adjacent), where d represents the degree of i (the number of nodes adjacent to i). Given a node set $V = \{1, 2, ..., N\}$ and a label-index set $I = \{1, 2, ..., l\}$, it is assumed that each node corresponds to a label in I, with labels known for only a small proportion of nodes relative to $|V|$. 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 reaches a i-labeled node before reaching any other labeled node. If $\operatorname{argmax}_{i \in I} p_i(y) = k$, the algorithm assigns the label k to the node y. (If a node y already has the label k, then $p_i(y) = 1$ if and only if $i = k$.)

The task is to determine $p_i(y)$ for all $y \in V$ and $i \in I$. It is known that $p_i(y)$ can be derived as the solution to the following Dirichlet boundary value problem:

$$\begin{aligned} Lp_i(x) &= 0 \ \text{ if } \ x \in F = (E_i \cup H_i)^c, \\ p_i(x) &= 1 \ \text{ if } \ x \in E_i, \\ p_i(x) &= 0 \ \text{ if } \ x \in H_i, \end{aligned} \tag{2.1}$$

where $L = D - A$ is the graph Laplacian matrix, D and A are the degree and adjacency matrices of the given graph G, respectively [Lim20]. E_i is the set of i-labeled nodes, H_i is the set of labeled nodes excluding i-labeled nodes.

Bendito, Carmona, and Encinas [BCE03] proposed a novel approach to solve the Dirichlet problem (2.1) using *equilibrium measures*. For any decomposition $V = F \cup F^c$ where F and F^c are both non-empty, they proved the existence of a unique measure (function) v on V 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. This measure is called the equilibrium measure and denoted by v^F. For $V = F \cup F^c$ where F and F^c are the sets of unlabeled and labeled nodes, respectively, they showed that the solution p_i of (2.1) can be represented as:

$$p_i(x) = \sum_{z \in E_i} \frac{v^{\{z\} \cup F}(x) - v^F(x)}{v^{\{z\} \cup F}(z)}, \quad x \in V. \tag{2.2}$$

Since v^F can be obtained by solving a linear program, (2.2) offers an efficient method for solving the Dirichlet problem (2.1); see [BCE03] for details. The algorithm described above will be referred to as RW in this paper, which stands for random walk. RW method not only serves as a performance benchmark for hypergraph-based strategies, but also plays a crucial role in node probability initialization. This initial step lays the basis for the subsequent training of the objective function that we describe from now on.

2.4. **Clique-based probabilistic objective function.** We employ the clique-based probabilistic objective function on hypergraphs used in [KL23]. Given a graph with a node set $V = \{1, 2, ..., N\}$ and a label set $I = \{1, 2, ..., l\}$, the probability distribution assigned to a node j is denoted by $p^j = (p_1^j, p_2^j, \ldots, p_l^j)$, where p_i^j denotes the probability that node j having label i, so that $\sum_{i=1}^{l} p_i^j = 1$ for all $j \in V$. Recall that K_k and M are the set of k-cliques and the maximum possible value of k in the graph, respectively. Define the permutation set with repetitions (denoted by S_k) as the set of ordered and repetition allowed arrangements of k elements in $I = \{1, ..., l\}$, so that $|S_k| = l^k$. Then

[KL23] proposed the following objective function on hypergraphs for node classification

$$J = \sum_{k=2}^{M} W_k \sum_{(n_1,\dots,n_k)\in K_k} \sum_{(m_1,\dots,m_k)=\theta\in S_k} C_\theta p_{m_1}^{n_1} p_{m_2}^{n_2} \dots p_{m_k}^{n_k}, \tag{2.3}$$

where $W_k \geq 0$ are weight parameters, $C_\theta = \binom{k}{e_1,e_2,\dots,e_l} = \frac{k!}{e_1!e_2!\dots e_l!}$ such that $\sum_{i=1}^{l} e_i = k$, and e_i is the number of occurrences of the label i in $\theta = (m_1, m_2, ..., m_k)$. Then [KL23] considered the following optimization problem

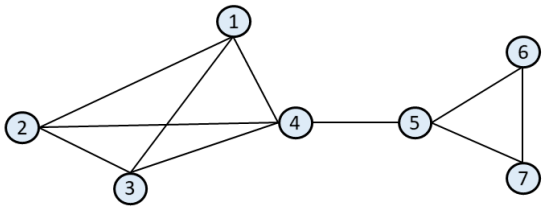
$$\text{minimize } J \text{ over } \Delta^N = \Delta_1 \times \Delta_2 \times \cdots \times \Delta_N \tag{2.4}$$

where $\Delta_j = \{p^j = (p_i^j)_{i\in I} \,|\, \sum_{i=1}^{l} p_i^j = 1, p_i^j \geq 0\}$ is the probability simplex in $\mathbb{R}^l$. Observe that J assigns a higher penalty to cliques with a greater diversity of labels and vice versa, for each k-clique. The penalties for all k-cliques are then summed up, followed by a weighted sum for each K_k. This function is formulated under the assumption that nodes within higher-order cliques in the network tend to have similar labels. Consequently, the objective function is designed such that higher-order cliques exert greater pressure on their constituent nodes to adopt homogeneous labels compared to lower-order cliques.

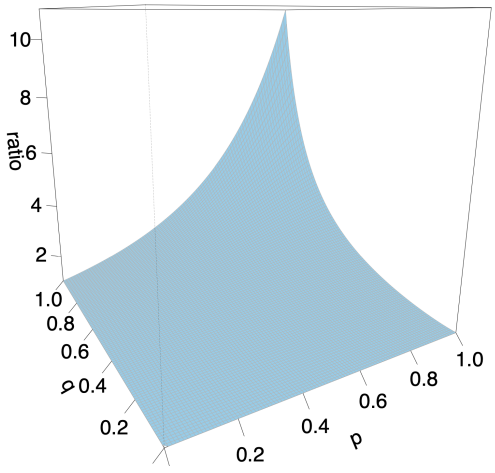
3. Simplified objective function based on the maximal cliques

Recent research findings (referenced in, for instance, [TLJ+24, LRK+18]) underscore the substantial improvement in prediction performance for node classification tasks achieved through the integration of higher-order cliques into the training process. This enhancement stems from the recognition that higher-order cliques capture a more intricate network of node relationships, thereby providing a richer context for learning algorithms. Specifically, the objective function (3.1) is tailored to apply increased pressure on nodes within these complex cliques to adopt more homogeneous labels. However, addressing the minimization problem of J on the domain Δ^n becomes exceedingly challenging due to the rapidly increasing number of total cliques in the graph.

This paper is motivated by the question of how to effectively reduce the computational complexity associated with problem (2.4) while preserving the enhanced prediction performance through higher-order clique interactions. As a result, we propose using an objective function that only considers the network's maximal cliques. A maximal clique is a clique that is not a subset of a larger clique (Figure 1a). Let $\overline{K}_k$ denote the set of maximal k-cliques in G. In this paper, we investigate the reduced objective function



(A) In this graph, there are 16 cliques: $K_2 = \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4),(4,5),(5,6),(5,7),(6,7)\}$, $K_3 = \{(1,2,3),(1,2,4),(1,3,4),(2,3,4),(5,6,7)\}$, and $K_4 = \{(1,2,3,4)\}$. On the other hand, there are 3 maximal cliques: $(1,2,3,4)$, $(4,5)$, $(5,6,7)$.



(B) The ratio represents $\frac{\sum_{k=2}^{4}\mathbb{E}[|K_k|]}{\sum_{k=2}^{4}\mathbb{E}[|K_k^{\max}|]}$.

$$\overline{J} = \sum_{k=2}^{M} W_k \sum_{(n_1,\ldots,n_k)\in\overline{K}_k} \sum_{(m_1,\ldots,m_k)=\theta\in S_k} C_\theta p_{m_1}^{n_1} p_{m_2}^{n_2} \cdots p_{m_k}^{n_k}. \tag{3.1}$$

As a result, no clique interaction is counted twice in $\overline{J}$. To illustrate the computational reduction achieved using $\overline{J}$, we derive the expected number of all cliques and maximal cliques in a network generated by the PPM. We begin by providing a simple example.

Example 3.1. Let $V = \{a,b,c,d\}$ and $I = \{1,2\}$, so there are four nodes and two labels in the network. Let the nodes $C_1 = \{a,b\}$ carry label 1, and the nodes $C_2 = \{c,d\}$ carry label 2. Let the graph $G = (V,E)$ be generated by PPM with parameters p, q as in Section 2.2. Then the expected number of edges in G, denoted by $\mathbb{E}[|K_2|]$, is easily computed as $\mathbb{E}[|K_2|] = 2p + 4q$. Similarly, the expected number of triangles $\mathbb{E}[|K_3|] = \binom{4}{3}pq^2 = 4pq^2$, and the expected number of tetrahedra $\mathbb{E}[|K_4|] = p^2q^4$. Hence, the expected number of cliques in G is given by $\sum_{k=2}^{4}\mathbb{E}[|K_k|] = 2p + 4q + 4pq^2 + p^2q^4$.

To calculate the expected number of maximal cliques in G, consider the edge (a,b) as an example. For (a,b) to be a maximal clique in G, the condition is $\{(a,c),(b,c)\} \nsubseteq E$ and $\{(a,d),(b,d)\} \nsubseteq E$. That is, the edge (a,b) should not be contained in a triangle in G. The probability of this occurring is $(1-q^2)(1-q^2) = (1-q^2)^2$. Thus, the probability of the existence of the edge (a,b) with it being a maximal clique is $p(1-q^2)^2$. Similarly, the probability of the existence of the edge (a,c) with it being a maximal clique is $q(1-pq)^2$. Therefore, the expected number of maximal edges is $2p(1-q^2)^2 + 4q(1-pq)^2$.

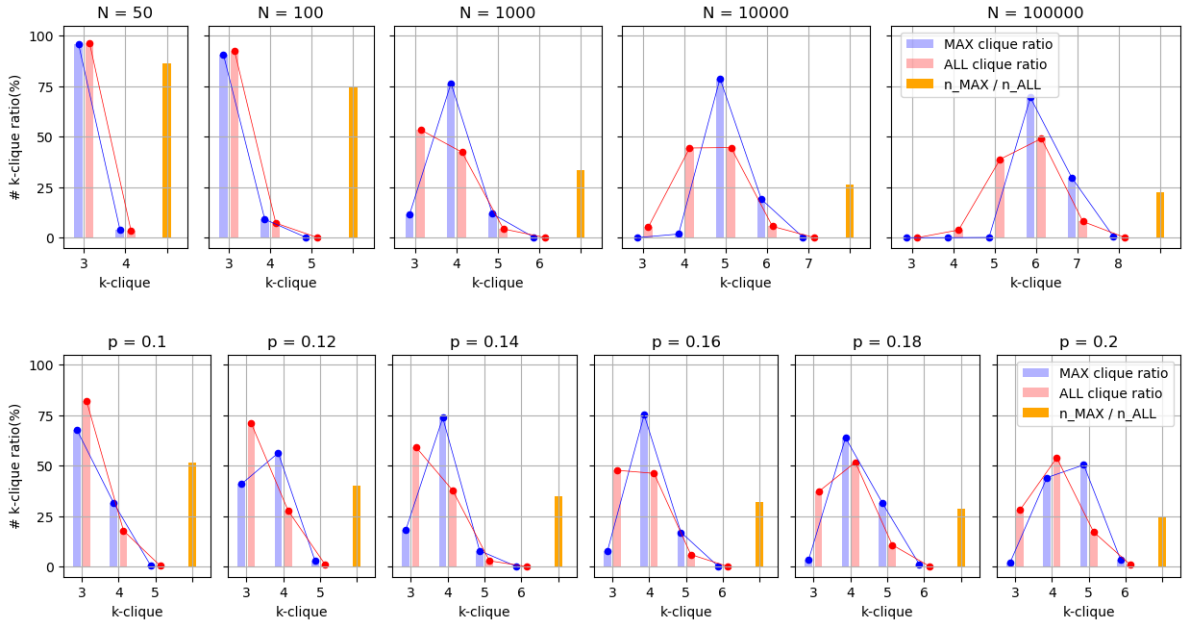


FIGURE 2. The upper and lower figures illustrate changes in the expected proportion of higher-order cliques within maximal cliques and all cliques, respectively. These changes result from variations in the number of nodes (upper) and node connection probability (lower) in the PPM. Specifically, we set $V = C_1 \cup C_2 \cup C_3$, with $|C_1| \approx |C_2| \approx |C_3| \approx N/3$, and $q = 0.015$ is fixed throughout. In the upper and lower figures, $p = 0.15$ and $N = 1000$ are fixed, respectively. Blue and red lines indicate the proportion of the expected number of k-maximal cliques and all cliques in the PPM, respectively, while the orange line represents the ratio of the expected number of maximal cliques to the expected number of all cliques.

For a triangle (a, b, c) to be maximal, it requires $\{(a, d), (b, d), (c, d)\} \nsubseteq E$, with a probability of $1-pq^2$. Hence, the expected number of maximal triangles is $4pq^2(1-pq^2)$. Finally, the tetrahedron (a, b, c, d) is maximal, with a probability of existence of p^2q^4 as before. Therefore, the expected number of maximal cliques in the PPM is given by $\sum_{k=2}^{4} \mathbb{E}[|K_k^{\max}|] = 2p(1-q^2)^2 + 4q(1-pq)^2 + 4pq^2(1-pq^2) + p^2q^4$. Figure 1b illustrates the ratio $\frac{\sum_{k=2}^{4} \mathbb{E}[|K_k|]}{\sum_{k=2}^{4} \mathbb{E}[|K_k^{\max}|]}$. We see that the ratio increases as (p, q) increases.

Detailed explicit formulas for the expected number of all cliques and maximal cliques, along with their derivation for general N, p, q, are provided in Proposition A.1. Figure 2 visually represents the expected relative proportions of the number of k-cliques and maximal k-cliques for each k, as well as the ratio between the expected total number of

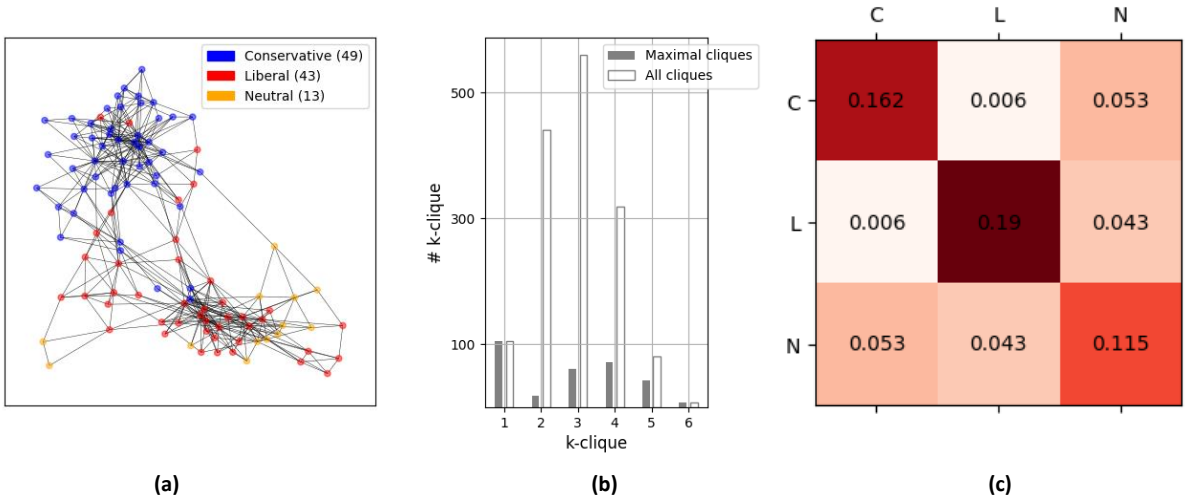


FIGURE 3. Political book dataset. (a) illustrates the dataset configuration using a network, (b) shows the number of k-cliques for maximal strategy (gray) and strategy using all cliques (white), and (c) shows the averaged connection probability between three labels C,L,N, representing conservative, liberal, and neutral books.

maximal cliques and all cliques, across various parameters N, p, q. It is observed that in the collection of maximal cliques, the proportion of higher-order cliques is greater than in the collection of all cliques. This trend becomes more pronounced as N or p increases.

4. Experimental setup

4.1. **Synthetic and real data.** To assess the performance of the proposed strategy, this study considers balanced networks, imbalanced networks generated by the PPM, and political book real data. The balanced data consists of three labels (i.e., $|I| = 3$), with fifty nodes corresponding to each label (thus, $|V| = 150$). The homo-connection probability (connection probability between nodes with the same label) p is selected between 0.1 and 0.2, while the hetero-connection probability (connection probability between nodes with different labels) q is selected between 0.01 and 0.025. Moreover, since this study is based on semi-supervised learning, the prior information ratio (the proportion of nodes whose labels are known) r is chosen between 0.01 and 0.1. For imbalanced data, three labels are used, and the number of nodes corresponding to each label is generated as 60, 40, and 20, respectively. The homo, hetero-connection probability, and prior information ratio are set to the same as in the balanced experiments.

The political book real data [New06, VK] is based on Amazon's book co-purchase records during the 2004 U.S. presidential election period. It comprises 105 books (nodes), each labeled as conservative (49 books), liberal (43 books), and neutral (13 books). Books purchased together by the same buyer are connected by an edge. The averaged homo-connection probability and hetero-connection probability for the dataset are 0.172 and 0.021, respectively. One node from each label is randomly selected as prior information, resulting in a prior information ratio of $3/105 = 0.029$. The configuration of the political book data is illustrated in Figure 3.

4.2. **Error metrics.** Precision, Recall, F1-score, and Accuracy are fundamental error metrics for classification problems used to assess the performance of a model. In the case of binary classification, let TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively, which can be derived from the 2×2 confusion matrix. The error metrics are defined by

$$\mathsf{Precision} = \frac{\mathsf{TP}}{\mathsf{TP}+\mathsf{FP}}, \quad \mathsf{Recall} = \frac{\mathsf{TP}}{\mathsf{TP}+\mathsf{FN}},$$
$$\mathsf{F1\text{-}score} = \frac{2 \cdot \mathsf{Recall} \cdot \mathsf{Precision}}{\mathsf{Recall}+\mathsf{Precision}}, \quad \mathsf{Accuracy} = \frac{\mathsf{TP}+\mathsf{FN}}{\mathsf{TP}+\mathsf{TN}+\mathsf{FP}+\mathsf{FN}}.$$

In multi-label classification, the task can be binarized using the one-vs-rest approach, which allows for consistent calculation of the error metrics. The experiment is performed using synthetic and real datasets, each with three labels, yielding 3×3 confusion matrices. As a result, the values of the error metrics can be obtained using the one-vs-rest method.

4.3. **Optimization.** SLSQP (Sequential Least SQuares Programming) is a numerical optimization algorithm designed to solve optimization problems with constraints [Kra94, Kra88]. This method is used to find the optimal values of parameters that minimize the sum of squared differences of the objective function while simultaneously satisfying both equality and inequality constraints. SLSQP addresses the optimization problem by sequentially resolving quadratic programming problems. In each iteration, the algorithm formulates and solves a quadratic programming sub-problem with linearized constraints, thereby progressively improving the objective. We recall that each node j in a network is associated with a probability vector $(p_i^j)_{i\in I} \in \Delta_j \subseteq \mathbb{R}^l$, where $l = |I|$ denotes the number of distinct labels. As a result, the sum of all p_i^j over $i \in I$ must be 1 (an equality constraint), and each p_i^j must lie between 0 and 1 (inequality constraints). Due to these

considerations, we use SLSQP as our optimization approach in this study, with the initial value $(p_i^j)_{i\in I, j\in V}$ specified by the RW, i.e., the solution to the Dirichlet problem (2.1).

5. Results

The acronym MAX will be used to denote the strategy based on maximal cliques (3.1), while ALL denotes the strategy that includes all cliques in the network (2.3). For performance comparison, in addition to RW, we employ the following optimization

$$\text{minimize} \sum_{(n_1,n_2)\in K_2} \sum_{(m_1,m_2)=\theta\in S_2} C_\theta p_{m_1}^{n_1} p_{m_2}^{n_2} \quad \text{over} \quad \Delta^N. \tag{5.1}$$

Note that this objective function exclusively considers pairwise interactions in the network. Due to this, the outcomes of (5.1) will be referred to as PI. This section presents the performance enhancements of MAX compared to RW and PI. Throughout our experiments, we notice that the performances of RW and PI are similar. Consequently, we report the average of the two results as the performance of the *non-hypergraph approach*, primarily focusing on the performance gains of MAX over the non-hypergraph approach. Furthermore, we examine the performance degradation of MAX compared to ALL.

We evaluate MAX using balanced and imbalanced data generated using PPM, as well as real political book data. The evaluation considers three hyperparameters: p, q, and r (representing homo-connection probability, hetero-connection probability, and prior information ratio, respectively), with respect to Precision, Recall, F1-score, and Accuracy. The implementation algorithm is available at `https://github.com/kooeunho/HOI_max`.

5.1. **Balanced Data.** We evaluate the proposed approach using three p values (0.10, 0.15, 0.20), four q values (0.010, 0.015, 0.020, 0.025), and four r values (0.01, 0.04, 0.07, 0.10). The comprehensive results encompassing three hyperparameters and four error metrics are presented in Figure 4. Moreover, the performance gains of MAX and ALL against the non-hypergraph approach (that is, the average of RW and PI) for three hyperparameters is illustrated in Figure 5. According to the experimental results, on average, MAX utilizes 84.71% of the cliques of order 3 or higher in training compared to ALL for all hyperparameters. Additionally, MAX exhibits an average performance decrement of 2% against ALL for all error metrics. Notably, MAX inherits an important characteristic of ALL, specifically its superior performance in challenging scenario (that is, low homo-connection probability, high hetero-connection probability, and low prior information ratio) against non-hypergraph approach, according to Figure 5. This

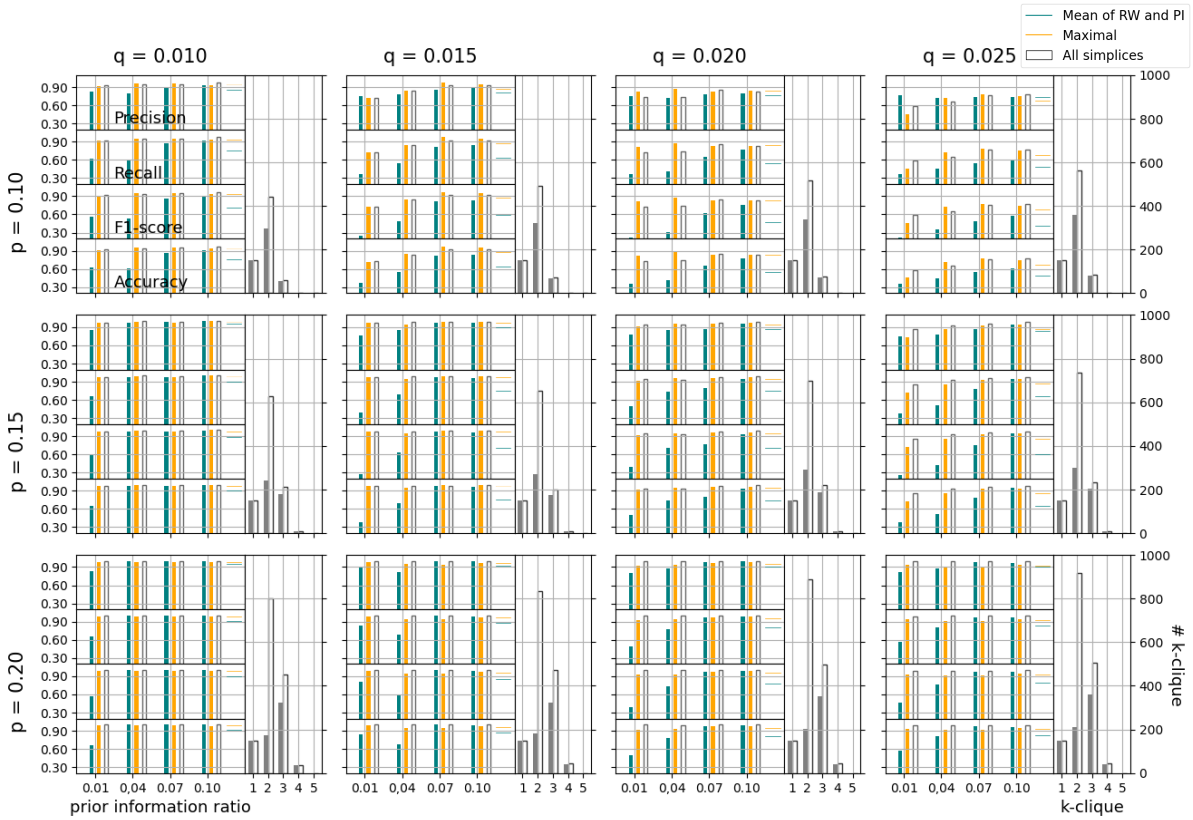


FIGURE 4. Results on balanced graphs. In the random graph generation, we set $V = C_1 \cup C_2 \cup C_3$ (i.e., there are three labels) with $|C_1| = |C_2| = |C_3| = 50$, thus $|V| = 150$. Three rows and four columns correspond to the homo-connection probability p and hetero-connection probability q. Green, orange, and white denote the performance of the non-hypergraph approach (the average of RW and PI), the maximal clique approach (MAX), and the approach utilizing all cliques (ALL), respectively. Each panel displays four rows representing the performance metrics of Precision, Recall, F1-score, and Accuracy. The right side of each panel illustrates the clique distribution for the maximal strategy (in gray) and the strategy utilizing all cliques (in white). On the left side of each panel, the x-axis represents the prior information ratio.

outcome can be attributed to the MAX's use of the majority of higher order cliques employed in ALL, in alignment with the structure of the objective function, which aims to encourage nodes within a higher order clique to possess similar labels.

5.2. **Imbalanced Data.** In the imbalanced experiment, we utilize the same hyperparameter values (p, q, r) as in the balanced experiment. The overall results concerning hyperparameters and error metrics are depicted in Figure 6. Additionally, Figure 7

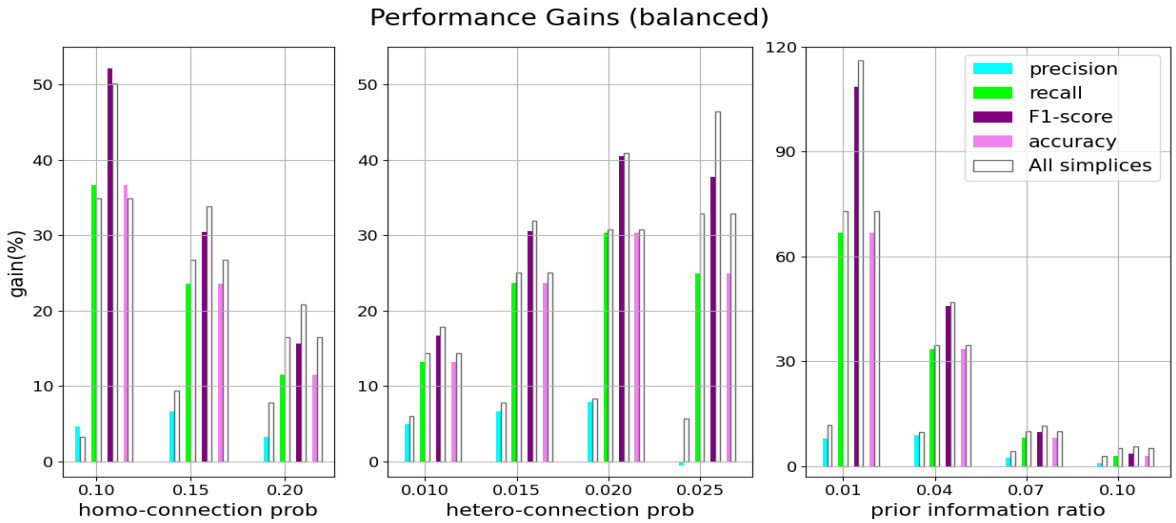


FIGURE 5. Performance gains in balanced experiments. Cyan, green, purple, and magenta represent the performance gains of the maximal clique strategy compared to the non-hypergraph strategy (the average performance of RW and PI) in terms of Precision, Recall, F1-Score, and Accuracy. White indicates the performance gains of the strategy using all cliques against non-hypergraph approaches. The left, middle, and right panels correspond to the performance gains concerning the homo-connection probability p, hetero-connection probability q, and prior information ratio r.

illustrates the performance gains of MAX and ALL compared to the non-hypergraph approach. For all hyperparameters, MAX employs 83.24% of the number of cliques of order 3 or higher used in training by ALL. Meanwhile, the average performance decreases of MAX against ALL are 8.89%, 10.67%, 10.04%, and 9.15% for Precision, Recall, F1-score, and Accuracy, respectively. Notably, in the imbalanced experiment, MAX exhibits characteristics similar to the significant performance gains of ALL, namely, low Precision and high Recall and F1-score (Figure 7). This highlights the utility of both MAX and ALL in scenarios, such as medical diagnostics, where identifying a high proportion of actual true positives is crucial, even at the expense of accepting some incorrect positive predictions.

5.3. **Real Data.** The political book real data is an imbalanced network that consists of 49 conservative books, 43 liberal books, and 13 neutral books. We randomly select one node from each label to use as prior information in semi-supervised learning, resulting in a prior information ratio of $r = 0.029$. The number of cliques of order 3 or higher used in training in MAX is 18.17% of that used in ALL, and the average performance decreases for MAX against ALL are 8.76%, 17.34%, 13.17%, and 13.76% for Precision,

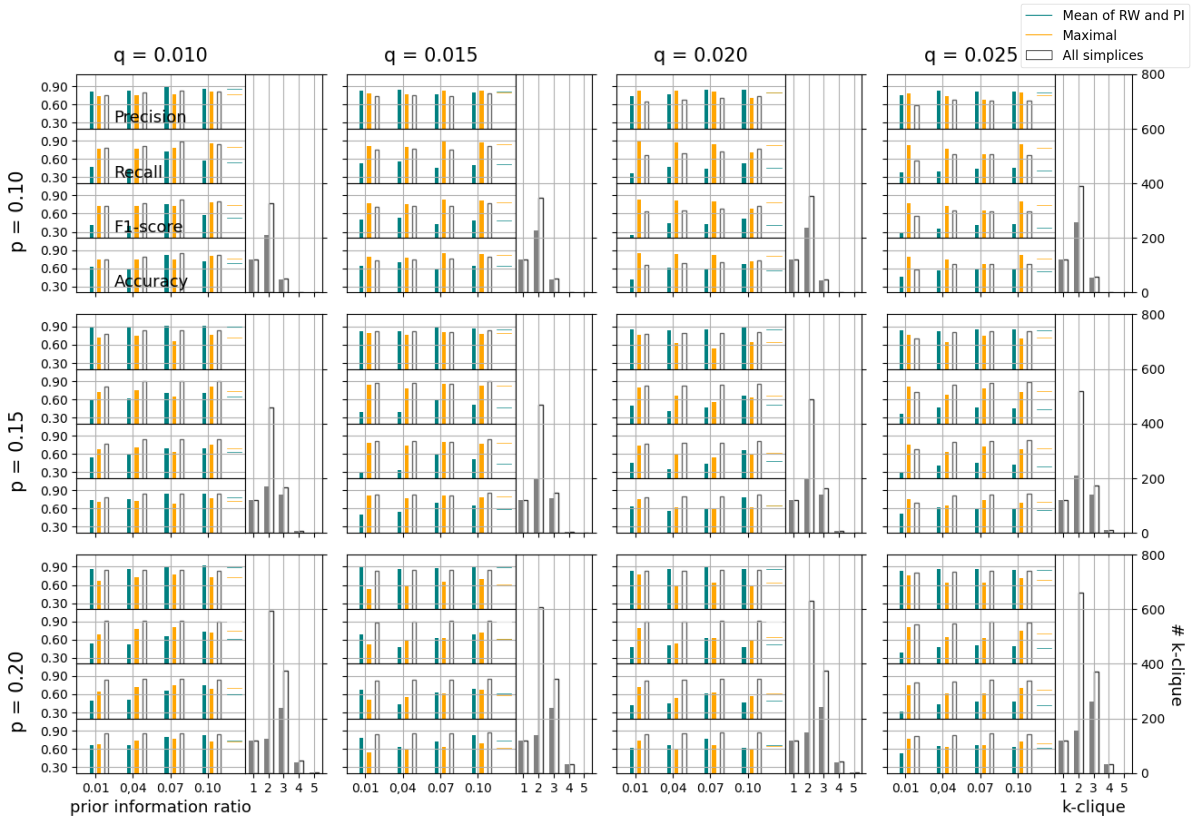


FIGURE 6. Results obtained from imbalanced graphs were generated using a random graph generation process where we set $V = C_1 \cup C_2 \cup C_3$ (i.e., there are three labels) with $|C_1| = 60$, $|C_2| = 40$, $|C_3| = 20$, resulting in $|V| = 120$. Otherwise, the information provided is identical to that described in the caption of Figure 4.

Recall, F1-score, and Accuracy, respectively. Notably, MAX continues to exhibit the positive characteristics observed in the synthetic imbalanced graph experiments, such as low Precision and high Recall and F1-score; see Figure 8. The weight W_k in the objective functions (2.3), (3.1) is set as $W_k = \alpha^k$, with the base value α ranging from 1 to 2.5. In both MAX and ALL, as the exponent base α increases, we observe that the number of true positives increases, while the number of false negatives and false positives decreases.

5.4. **Discussion.** In this section, we list some discussion about our study.

Firstly, Figure 2 illustrates the proportion of higher-order cliques in MAX as the number of nodes and the homo-connection probability change in the planted partition model. Compared to ALL, as the number of nodes increases, the total number of cliques used in training decreases, but the proportion of higher-order cliques (cliques of order 3

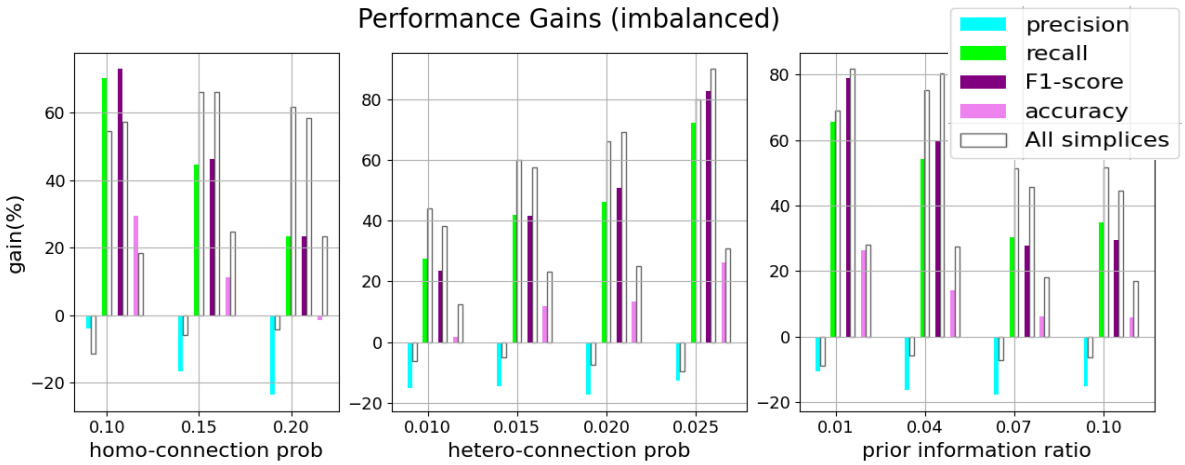


FIGURE 7. Performance gains in imbalanced experiments. Cyan, green, purple, and magenta represent the performance gains of the maximal clique strategy compared to the non-hypergraph strategy (the average performance of RW and PI) in terms of Precision, Recall, F1-Score, and Accuracy. White indicates the performance gains of the strategy using all cliques against non-hypergraph approaches. The left, middle, and right panels correspond to the performance gains concerning the homo-connection probability p, hetero-connection probability q, and prior information ratio r.

or higher) used in training increases. Furthermore, as the homo-connection probability increases, the proportion of higher-order cliques used in training also increases in MAX.

The clique-based probabilistic objective function used in this study incorporates an algorithm that imposes a penalty for cliques of higher dimensions, encouraging nodes within the same clique to have similar labels. This feature is amplified by assigning more weight to higher order cliques (increasing the value of W_k in (3.1) as k increases). As a result, MAX can yield additional performance gains, as showcased in the political book experiment. This explains why the performance of MAX is comparable to that of ALL.

Secondly, in all experiments, the hypergraph approach offers two significant advantages over the non-hypergraph approach. Generally, in classification prediction tasks, higher predictive performance is expected under conditions of higher homo-connection probability, lower hetero-connection probability, and a higher prior information ratio. However, in the balanced experiment, the hypergraph approach demonstrates a greater performance gain against the non-hypergraph approach under challenging conditions (lower homo-connection probability, higher hetero-connection probability, and lower

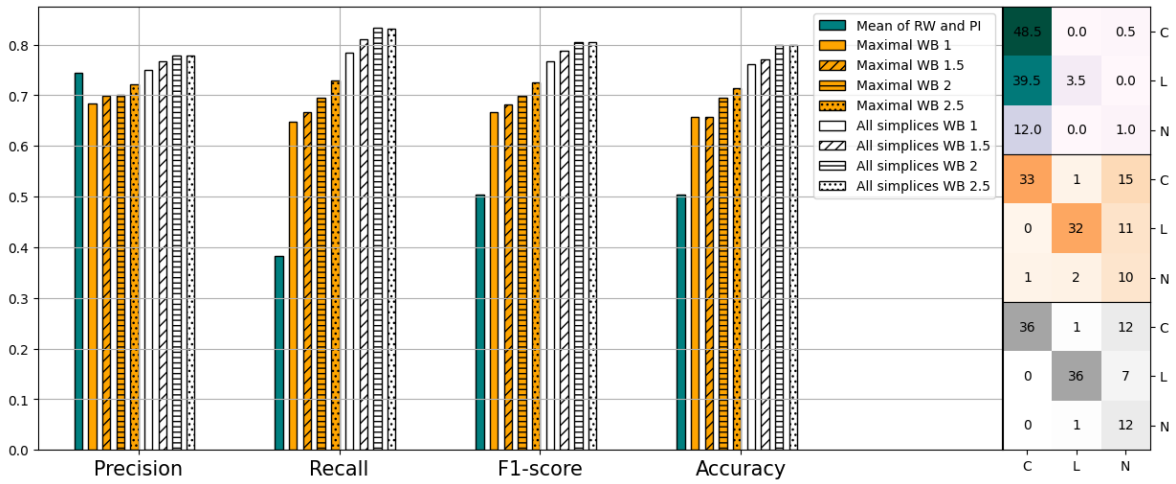


FIGURE 8. Experimental results on political book dataset. Green, orange, and white denote the performance of the non-hypergraph approach (the average of RW and PI), the maximal clique approach (MAX), and the approach using all cliques (ALL), respectively. Four orange and white bars for each error metric illustrate the performance changes as weight parameter increases (WB denotes the weight base α). The right panels display the confusion matrices corresponding to the best performance for each strategy.

prior information ratio). Furthermore, in the imbalanced experiment, although the hypergraph approach shows lower Precision compared to the non-hypergraph approach, it exhibits higher Recall and F1-score. This aspect is particularly advantageous in many real-world scenarios where accurately classifying false negatives is crucial (such as diagnosing a patient having cancer as healthy), even if it means allowing some false positives. Despite using substantially fewer cliques in training compared to ALL, experimental results have shown that MAX inherits the two crucial benefits of ALL.

Thirdly, this study generates random graphs based on the planted partition model. However, it is evident that networks generated by the PPM and real networks exhibit different clique distributions. For instance, in the political book data, the average homo and hetero connection probabilities are $p = 0.172$ and $q = 0.021$, with a clique distribution of $(105, 441, 560, 319, 81, 7)$. This indicates that the real network encompasses a variety of higher-order structures ranging from 1-clique (nodes in the network) to 6-cliques. In contrast, for the PPM model corresponding to the same p, q values, the average clique distribution is $(105, 440, 268, 9)$. This suggests that the PPM model may not adequately

replicate the structure of real data. To address this challenge, there is a need to develop methods for random graph generation that accurately capture the distribution of higher-order cliques present in real networks. Recently, there has been a shift towards treating multiple nodes as a single entity when generating graphs, aiming to understand the functions of complex systems such as co-authorship networks, brain networks, and social contact networks [KG24]. Developing new graph generation methodologies capable of mimicking the higher-order structures observed in real networks appears crucial for advancing network inference and related research.

6. Conclusion

The clique-based objective function (2.3) and its optimization (2.4) have demonstrated high effectiveness in node classification tasks. However, as discussed in [KL23], they suffer from significant computational complexity, particularly with larger networks, due to the extensive utilization of higher-order nonlinear interaction terms in the objective function. This paper aims to alleviate this computational burden while preserving classification performance. Accordingly, we propose using the reduced objective function (3.1), which focuses on maximal clique interactions in the network. In an appendix, we derive formulas for the expected number of k-cliques and the expected number of maximal k-cliques in a network generated by PPM for each positive integer k. Utilizing these formulas, we analyze the distribution of higher-order cliques based on changes in the number of nodes and the connection probabilities between nodes, as illustrated in Figure 2. This demonstrates how the number of cliques in the optimization can be dramatically reduced by focusing on the maximal cliques. Subsequently, we compare and analyze the results of training on all cliques in the network (ALL) and only on maximal cliques (MAX). Our experiments demonstrate that, although MAX utilizes a substantially smaller number of cliques for training compared to ALL, its prediction performance remains comparable. Additionally, MAX inherits the advantages of ALL, as discussed in Section 5.4. The maximal clique-based objective function (3.1) is built on the premise of inducing all nodes within a higher-order clique to have homogeneous labels, a feature that allows the objective function to reduce reliance on lower-order cliques. This heuristic elucidates why MAX, despite employing much fewer cliques, achieves comparable performance to ALL. Furthermore, we discover that assigning higher weights W_k to higher-order cliques can yield additional performance enhancements in MAX. This provides further experimental evidence that reducing reliance on lower-order cliques can decrease the number of cliques

required for the overall training process, thereby substantially reducing computational complexity while maintaining comparable classification performance.

Appendix A. Derivation of the expected number of maximal cliques

We derive explicit formulas for the expected number of all cliques and maximal cliques for arbitrary N, p, q. Figure 2 is based on this derivation. Let the label index be $I = \{1, 2, \ldots, l\}$, and the number of nodes with label i be N_i for each $i \in I$, hence, the total number of nodes $N = \sum_{i=1}^{l} N_i$. Each $k \in \mathbb{N}$ will represent the size of a clique. Let $\pi(k)$ denote the set of partitions of k. We list a partition of k in increasing order. For example, $\pi(4) = \{(1,1,1,1), (1,1,2), (1,3), (2,2), (4)\}$, thus $|\pi(4)| = 5$. Each $\eta \in \pi(k)$ represents the label composition in a k-clique. For example, $\eta = (1,1,2)$ tells us that the 4-clique under consideration has 3 distinct labels assigned to its 4 nodes, where two nodes have the same label. Let $|\eta|$ be the number of entries of η (that is, number of distinct labels in a clique), so that $\eta = (\eta_1, ..., \eta_{|\eta|})$. Let $\eta_+ = \sum_{i=1}^{|\eta|} \eta_i$, and $\eta! = \eta_1!\eta_2! \ldots \eta_{|\eta|}!$. For example, if $\eta = (2,2,3)$, then $|\eta| = 3$, $\eta_+ = 7$ and $\eta! = 2!2!3! = 24$. For $L, M \in \mathbb{N}$, let $S(L, M)$ be the set containing all possible permutations of M distinct numbers selected from $\{1, 2, ..., L\}$. For example, $S(4,2) = \{(1,2), (1,3), (1,4), (2,1), (2,3), (2,4), (3,1), (3,2), (3,4), (4,1), (4,2), (4,3)\}$. Let $S(L, M) = \emptyset$ if $L < M$. Let p, q denote the homo- and hetero-connection probabilities in the PPM.

Proposition A.1. *For each $k \geq 2$, let $\mathrm{EC}(k)$ and $\mathrm{EC}_{\mathrm{M}}(k)$ represent the expected number of the k-cliques, and of the maximal k-cliques in the PPM, respectively. Then it holds*

$$\mathrm{EC}(k) = \sum_{\eta \in \pi(k)} P(\eta) \cdot \frac{1}{\eta!} \sum_{(j_1, \ldots, j_{|\eta|}) = \xi \in S(l, |\eta|)} A(\eta, \xi), \quad \textit{and} \tag{A.1}$$

$$\mathrm{EC}_{\mathrm{M}}(k) = \sum_{\eta \in \pi(k)} P(\eta) \cdot \frac{1}{\eta!} \sum_{(j_1, \ldots, j_{|\eta|}) = \xi \in S(l, |\eta|)} A(\eta, \xi) B_1(\eta, \xi) B_2(\eta, \xi), \quad \textit{where} \tag{A.2}$$

$$P(\eta) = p^{\sum_{m=1}^{|\eta|} \binom{\eta_m}{2}} q^{\binom{\eta_+}{2} - \sum_{m=1}^{|\eta|} \binom{\eta_m}{2}}, \quad A(\eta, \xi) = \prod_{m=1}^{|\eta|} \binom{N_{j_m}}{\eta_m},$$

$$B_1(\eta, \xi) = \prod_{m=1}^{|\eta|} (1 - p^{\eta_m} q^{\eta_+ - \eta_m})^{N_{j_m} - \eta_m}, \quad B_2(\eta, \xi) = \prod_{i \in I \setminus \{j_1, \ldots, j_{|\xi|}\}} (1 - q^{\eta_+})^{N_i},$$

where we assume $\binom{n}{m} = 0$ *if* $n < m$. *Hence,* $\mathrm{EC} = \sum_{k=2}^{N} \mathrm{EC}(k)$ *and* $\mathrm{EC}_{\mathrm{M}} = \sum_{k=2}^{N} \mathrm{EC}_{\mathrm{M}}(k)$ *represent the expected number of all cliques, and of the maximal cliques, in the PPM.*

Proof. It suffices to interpret the terms P, A, B_1, and B_2. Given $\eta \in \pi(k)$ (thus $k = \eta_+$), $P(\eta)$ represents the probability of the existence of a k-clique in the graph whose node label counting is given by η. This is because the total number of edges in the k-clique connecting nodes with the same label is $\sum_{m=1}^{|\eta|} \binom{\eta_m}{2}$, and hence, the number of edges connecting nodes with different labels is $\binom{k}{2} - \sum_{m=1}^{|\eta|} \binom{\eta_m}{2}$. Meanwhile, $\dfrac{1}{\eta!} \displaystyle\sum_{(j_1,\ldots,j_{|\eta|})=\xi \in S(l,|\eta|)} A(\eta, \xi)$ represents the total number of the set of k nodes whose label counting is given by η. This yields (A.1). To derive (A.2), let us first consider the simpler case where the probability of any two nodes being connected is p. The expected number of k-cliques is then $\binom{N}{k} p^{\binom{k}{2}}$. Now observe that a k-clique is a maximal clique if and only if there exists no outside node that connects to all of the k nodes in the clique. This observation implies that the expected number of maximal k-clique in this simple case is $\binom{N}{k} p^{\binom{k}{2}} (1 - p^k)^{N-k}$. By generalizing the term $(1 - p^k)^{N-k}$ for two parameters (p, q), we can derive B_1, B_2. Given a clique σ with η its node label counts, we divide the label index $I = \{1, ..., l\}$ into two parts: labels used to label the nodes in σ (denoted as I_1) and labels not used (denoted as I_2). Then $B_j(\eta, \xi)$, $j = 1, 2$, represents the probability that any node outside σ labeled in I_j does not connect to all of the nodes in σ. This completes the proof. □

References

[Abb18] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

[AVSG10] Armen E Allahverdyan, Greg Ver Steeg, and Aram Galstyan. Community detection with and without prior information. *Europhysics Letters*, 90(1):18002, 2010.

[BCE03] Enrique Bendito, Angeles Carmona, and Andrés M Encinas. Solving dirichlet and poisson problems on graphs by means of equilibrium measures. *European Journal of Combinatorics*, 24(4):365–375, 2003.

[BCI+20] Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, 874:1–92, 2020.

[BGHS23] Christian Bick, Elizabeth Gross, Heather A Harrington, and Michael T Schaub. What are higher-order networks? *SIAM Review*, 65(3):686–731, 2023.

[BHJ04] Phillip Bonacich, Annie Cody Holdren, and Michael Johnston. Hyper-edges and multidimensional centrality. *Social networks*, 26(3):189–203, 2004.

[BSS+08] Shumeet Baluja, Rohan Seth, Dharshi Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of the 17th international conference on World Wide Web*, pages 895–904, 2008.

[EK+12] David Easley, Jon Kleinberg, et al. Networks, crowds, and markets. *Cambridge Books*, 2012.

[EM12] Eric Eaton and Rachael Mansbach. A spin-glass model for semi-supervised community detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 900–906, 2012.

[FIA11] Giuseppe Facchetti, Giovanni Iacono, and Claudio Altafini. Computing global structural balance in large-scale signed social networks. *Proceedings of the National Academy of Sciences*, 108(52):20953–20958, 2011.

[Fri91] Noah E Friedkin. Theoretical foundations for centrality measures. *American Journal of Sociology*, 96(6):1478–1504, 1991.

[GD14] Debarghya Ghoshdastidar and Ambedkar Dukkipati. Consistency of spectral partitioning of uniform hypergraphs under planted partition model. *Advances in Neural Information Processing Systems*, 27, 2014.

[Gle15] David F Gleich. Pagerank beyond the web. *SIAM Review*, 57(3):321–363, 2015.

[HLL83] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[KBG17] Chiheon Kim, Afonso S Bandeira, and Michel X Goemans. Community detection in hypergraphs, spiked tensor models, and sum-of-squares. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pages 124–128. IEEE, 2017.

[KG24] Jung-Ho Kim and K-I Goh. Higher-order components dictate higher-order contagion dynamics in hypergraphs. *Physical Review Letters*, 132(8):087401, 2024.

[KL23] Eunho Koo and Tongseok Lim. Node classification in networks via simplicial interactions. *arXiv preprint arXiv:2310.10114*, 2023.

[Kra88] Dieter Kraft. A software package for sequential quadratic programming. *Forschungsbericht-Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

[Kra94] Dieter Kraft. Algorithm 733: Tomp–fortran modules for optimal control calculations. *ACM Transactions on Mathematical Software (TOMS)*, 20(3):262–281, 1994.

[Lim20] Lek-Heng Lim. Hodge laplacians on graphs. *Siam Review*, 62(3):685–715, 2020.

[LML+17] Thibault Lesieur, Léo Miolane, Marc Lelarge, Florent Krzakala, and Lenka Zdeborová. Statistical and computational phase transitions in spiked tensor estimation. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 511–515. IEEE, 2017.

[LRK+18] John Boaz Lee, Ryan A Rossi, Xiangnan Kong, Sungchul Kim, Eunyee Koh, and Anup Rao. Higher-order graph convolutional networks. *arXiv preprint arXiv:1809.07697*, 2018.

[LW19] Clement Lee and Darren J Wilkinson. A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1):1–50, 2019.

[MGYF10] Xiaoke Ma, Lin Gao, Xuerong Yong, and Lidong Fu. Semi-supervised clustering algorithm for community structure detection in complex networks. *Physica A: Statistical Mechanics and its Applications*, 389(1):187–197, 2010.

[MP07] Sofus A Macskassy and Foster Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of machine learning research*, 8(5), 2007.

[New06] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[TLJ+24] Bohan Tang, Zexi Liu, Keyue Jiang, Siheng Chen, and Xiaowen Dong. Hypergraph node classification with graph neural networks. *arXiv preprint arXiv:2402.05569*, 2024.

[TRP+08] Partha Talukdar, Joseph Reisinger, Marius Pasca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 582–590, 2008.

[VK] Orgnet LLC V. Krebs. http://www.orgnet.com/. *unpublished data*.

[VSGA11] Greg Ver Steeg, Aram Galstyan, and Armen E Allahverdyan. Statistical mechanics of semi-supervised clustering in sparse graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(08):P08009, 2011.

[Zha13] Zhong-Yuan Zhang. Community structure detection in complex networks with partial background information. *Europhysics Letters*, 101(4):48005, 2013.

[Zhu05] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.

[ZSW13] Zhong-Yuan Zhang, Kai-Di Sun, and Si-Qi Wang. Enhanced community structure detection in complex networks with partial background information. *Scientific reports*, 3(1):3241, 2013.