論文 / 著書情報
Article / Book Information

題目(和文)	ソーシャル・ネットワーク分析に関する研究: サンプリングと高次の相互作用に焦点を当てて
Title(English)	Studies on Social Network Analysis: Sampling and Higher-order Interactions
著者(和文)	中嶋一貴
Author(English)	Kazuki Nakajima
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第12170号, 授与年月日:2022年9月22日, 学位の種別:課程博士, 審査員:三好 直人,高安 美佐子,南出 靖彦,脇田 建,村田 剛志,首藤 一幸
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第12170号, Conferred date:2022/9/22, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

A Doctoral Thesis

Studies on Social Network Analysis: Sampling and Higher-order Interactions

Kazuki Nakajima

Submitted to
the Department of Mathematical and Computing Science,
School of Computing,
Tokyo Institute of Technology
for the Degree of Doctor of Science

Thesis Supervisors:
Kazuyuki Shudo and Naoto Miyoshi

September 2022

ABSTRACT

Social network analysis is the process of investigating social structures and dynamics using the representation of networks or graphs, and it is grounded in systematic empirical data on social interactions. A growing variety of data on social interactions, including online friendships among a huge number of users, time-varying interactions between individuals, and higher-order interactions among actors (e.g., individuals, institutions, and countries), provides opportunities to better understand social structures and dynamics. In this thesis, I present four works focusing on analyses of two types of social interaction data: friendships in online social networking services and higher-order social interactions.

In the first two works, we study how to accurately estimate structural properties of online social networks by querying a small number of unique nodes using random walk. Firstly, we present a practical framework for estimating properties of a social network involving private nodes, that do not publish their own friendships, via a random walk. The proposed framework may help us to investigate properties of the entire network involving private nodes. Secondly, we introduce the social graph restoration problem, motivated by the gap between the properties that the existing methodology allows one to accurately estimate and those of interest to analysts in practical scenarios. We propose a method for restoring the original network from a small sample obtained by a random walk, which may lead to an exhaustive analysis of online social networks.

In the last two works, we study how to analyze the structure and dynamics of social networks involving higher-order interactions among more than two actors, without using the one-mode projection of the original network. Firstly, we develop a family of reference models that randomize the structure of empirical hypergraphs. The proposed model preserves properties of the node and hyperedge at fine-tunable extents, which may help us to find the dependence of a given property on the structure and dynamics of empirical hypergraphs. Secondly, we analyze bipartite networks of institutions and collaborative research grants to investigate the patterns of grant collaborations among institutions. Our analyses suggest that institutions participating in many collaborations tend to densely collaborate with each other and dense grant collaborations among such institutions have advantages on the research productivity of those institutions.

Acknowledgements

I am deeply indebted to Prof. Kazuyuki Shudo for his great supervision since April 2017, for sharing his passion for science and research with me, and for providing me with an excellent research environment. My research spirit of realizing analysis methods that are faithful to the empirical data and practical scenarios of real-world social networks, which is common to all of my works in this thesis, is mostly inspired by his research policy and passion. His valuable assistance and heart-charming encouragement were very helpful, and I would not have completed this thesis and even have become a researcher without working with him.

I deeply express Prof. Naoki Masuda (he is currently with the State University of New York at Buffalo) for his courteous supervision since August 2020. He taught me much expertise in network science, discussed research an exorbitant number of times with me, and showed me ways to achieve high-level research. I am very pleased to have worked with him because I developed a strong interest in network science through his excellent book on introduction to network science. I would not have been able to accomplish my studies on social networks involving higher-order interactions (Chapters 4 and 5) without working with him.

I would like to thank Dr. Minas Gjoka (he is currently with Google) for his kind replies to my questions on his papers and thank Prof. Naoto Miyoshi for helpful comments on the theoretical analysis of the work to be presented in Chapter 2.

I would like to express Prof. Naoto Miyoshi for his co-supervision since April 2022. I would like to appreciate the members of my thesis committee, Prof. Ken Wakita, Prof. Kazuyuki Shudo, Prof. Misako Takayasu, Prof. Yasuhiko Minamide, Prof. Naoto Miyoshi, and Prof. Tsuyoshi Murata, for their constructive feedback.

I would like to thank Mr. Kenta Iwasaki (he is currently with CyberAgent, Inc.) for his kind and nice guidance in the first year of my research career. He is a good mentor who brings out the best in his people, and without him, I would not have become a researcher.

I am thankful to the present and past members of Shudo's group for the pleasant atmosphere over the four and half years and thank Ms. Miki Kitamura (she is an administrative staff at the Department of Mathematical and Computing Science, Tokyo Institute of Technology) for her great support of my research activities. I am also thankful to the members of Masuda's group and my host family during my academic abroad from August 2021 to April 2022 for the exciting experience in daily life and research activities in Buffalo, New York.

I express Ms. Mayumi Yamada (she is an administrative staff at the Department of Mathematical and Computing Science, Tokyo Institute of Technology) and Ms. Miki Kitamura for their great support for my academic abroad in Buffalo.

I express all the members of the track and field club at the Tokyo Institute of Technology for a very valuable and exciting experience with them. My research activities were much more fulfilling when I was devoting my passion to activities in the track and field club than today when I already retired.

Lastly, I deeply thank my family and my partner for their great and kind

support, and I would not have produced my research results in this thesis without their support.

Contents

Chapter 1

Introduction

Social network analysis is the process of investigating social structures and dynamics using the representation of networks or graphs. In a standard form, a social network is composed of a set of nodes (e.g., individuals, and institutions, and countries) and a set of edges (e.g., relationships and interactions) between nodes. A well-known example of empirical social networks is Zachary's karate club network [249]. Wayne Zachary observed social interactions among 34 members of a karate club at an American university over a three-year period from 1970 to 1972, resulting in a social network consisting of 34 nodes and 78 edges between them. Other examples of social networks include collaboration networks of film actors (i.e., networks in which an edge between two actors is present if they have co-starred at least one film) [237], collaboration networks of scientists (i.e., networks in which an edge between two scientists is present if they have co-written at least one journal paper) [170, 171], and many more [36, 116, 206, 236].

Various mathematical and computational methods have enabled us to investigate the structure and dynamics of empirical social networks. Many social networks share structural patterns, such as heterogeneous distributions of the node's degree (i.e., number of other nodes to which a node is directly connected), an abundance of triangles, correlation in terms of the degree of adjacent pairs of nodes, community structure, and many more [25, 33, 128, 177]. These and other structural properties affect social dynamic processes on networks such as epidemic spreading, information spreading, and evolution of cooperation [25, 33, 128, 177].

Social network analysis is grounded in systematic empirical data on social interactions [77]. Analyses using empirical data allow us to test theories and hypotheses on social structures and dynamics. Historically, data on social interactions were collected by painstaking means, and empirical studies of social networks were limited to a few hundred nodes [224]. The wealth of data brought about by advances in high-throughput tools and technologies over the last decades has greatly improved the scale and accuracy of studies on social network analysis. In fact, a variety of large-scale and systematic datasets is now widely used for research purposes, e.g., SNAP Datasets [137], Network Data Repository [196], and a collection of datasets recorded by the SocioPatterns [210].

In this thesis, I focus on analysis methods for two types of social interaction data: friendships in online social networking services and higher-order social interactions. A growing interest in these data over the last two decades has provided opportunities for understanding social structures and dynamics with higher resolution than ever, as well as highlighting the challenges associated with analyzing each data.

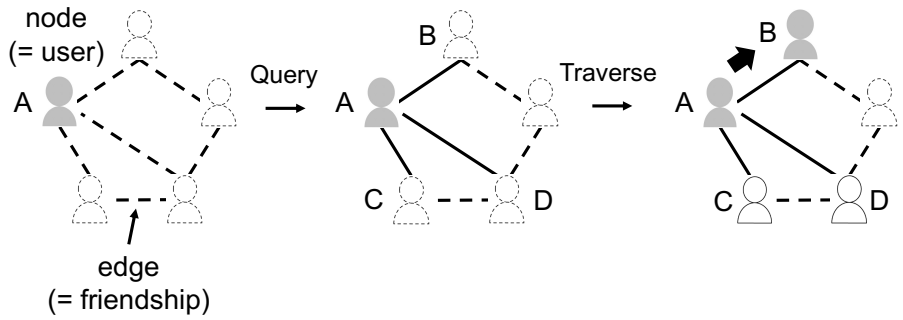


Figure 1.1: Crawling-based sampling on a social network. If we query node A, we find the neighbors of node A (suppose nodes B, C, and D). Then, we choose one of these neighbors (suppose node B), and the walker moves to node B. We repeat this process until we query a given small number of unique nodes.

1.1 Thesis Overview

This thesis is organized into two parts. The first part, which is composed of Chapters 2 and 3, discusses methods for analyzing structural properties of huge online social networks. The second part is devoted to methods for analyzing social networks involving higher-order interactions among more than two actors. Finally, I conclude this thesis in Chapter 6.

1.1.1 Huge Online Social Networks

The advent of huge online social networks (OSNs) has enabled us to investigate social structures and dynamics on a worldwide scale. Examples of OSNs include Cyworld [15], MySpace [15], Orkut [15,154], Flicker [154], LiveJournal [154], YouTube [154], Facebook [21, 85, 86, 220, 239], and Twitter [20, 90, 118, 122, 157]. Many empirical OSNs have had a huge number of users, sometimes in the hundreds of millions or more (e.g., Facebook had about 2.9 billion monthly active users as of March 31, 2022 [73]). A long series of studies have analyzed structural properties of a network where nodes represent users and edges represent friendships between users in an OSN.

In general, researchers perform a sampling of the graph data for analyses using public interfaces when complete data are not available to third parties due to privacy concerns. Many OSNs provide public interfaces to allow one to retrieve the neighbors of a user by querying the user. Crawling methods in which one repeatedly traverses a neighbor are effective for sampling the graph data in such OSNs (Fig. 1.1). Examples of crawling methods include the breadth-first search [45,121,154,197,239], snowball sampling [15,89,106,134,197], forest fire sampling [13, 65, 136, 197], and random walk [85, 86, 118]. A common challenge is how to accurately estimate properties by querying a small number of unique nodes. This is because (i) crawling methods typically induce sampling bias toward high-degree nodes and (ii) public interfaces typically limit the maximum number of queries within a particular time interval.

Re-weighted random walk is a practical framework for an unbiased estimation of properties of the OSNs [85, 86]. In this framework, one first performs a simple random walk on the underlying network (i.e., one repeatedly moves to a uniformly and randomly chosen neighbor using the public interfaces), which provides a sequence of sampled nodes that has the Markov property (i.e., a given sampled

node after the initial node depends on the previous sampled node). Then, one obtains an unbiased estimate of the property of interest by re-weighting each sampled node to correct the sampling bias derived from the Markov chain analysis. Based on this framework, a number of studies have developed algorithms that accurately estimate structural properties of OSNs using a small number of queries. Examples of structural properties that have been focused on include the network size (i.e., number of nodes) [97, 112], average degree [62, 85, 86], degree distribution [85, 86], joint degree distribution [84], clustering coefficients [32, 97, 190], motifs and graphlets [48, 95, 234], and node centrality [160, 161].

These existing algorithms typically assume that a social network comprises a set of nodes each of which publishes its own friendships. However, there is a certain percentage of *private nodes*, that do not publish their own friendships if they are queried in practical scenarios. For example, some previous studies reported that private nodes account for 27% of all the nodes on the Facebook network [45] and 34% of all the nodes on the Pokec network, which is an OSN in Slovakia [216]. When one attempts to apply such existing algorithms to real social networks, private nodes inhibit one from performing a simple random walk on the network and then induce a bias in estimators.

In Chapter 2, we present a practical framework for estimating properties via a random walk on a social network involving private nodes. First, we develop a sampling algorithm by extending a simple random walk to the case of a social network involving private nodes. Second, we propose estimators with reduced biases induced by private nodes for three network properties. Our experimental results show that the proposed estimators reduce the bias induced by private nodes in the existing estimators by up to 92.6% on empirical social network datasets involving private nodes. Some of the contents of this chapter have been published in a conference paper [162]. The full contents of this chapter will be published in a paper [164].

Analysts' interests in properties of social networks are generally diverse [36]; these properties include local structural properties (e.g., the degree distribution and clustering coefficient), global structural properties (e.g., the distributions of shortest-path lengths and betweenness centrality), and visual graph representations. However, the framework of re-weighted random walk, which we focus on in Chapter 2, is specialized in estimating local structural properties. The reasons are as follows. First, this framework forces analysts to sample most graph data to correct the sampling bias when attempting to estimate global structural properties, such as the shortest-path properties. Second, the quantity of re-weighted sample means is not sufficient to predict the structure of the original network, such as its visual representation. These observations motivate us to explore a framework that enables us to estimate various structural properties accurately on average.

Chapter 3 introduces the *social graph restoration problem*. In this problem, given a small sample of a social network obtained by a crawling method, we aim to generate a graph whose structural properties are as close as possible to the corresponding properties of the original network. To address this problem, we propose a method that generates a graph that preserves the subgraph sampled using a random walk in addition to the estimates of local structural properties obtained using the re-weighted random walk. Our experimental results show that the proposed method more accurately reproduces 12 structural properties, including both local and global structural properties, on average and the visual representation of the original network than existing methods. The contents of this chapter have been published in a conference paper [165].

(a)

Group X	Individual A, Individual B
Group Y	Individual A, Individual B, Individual C

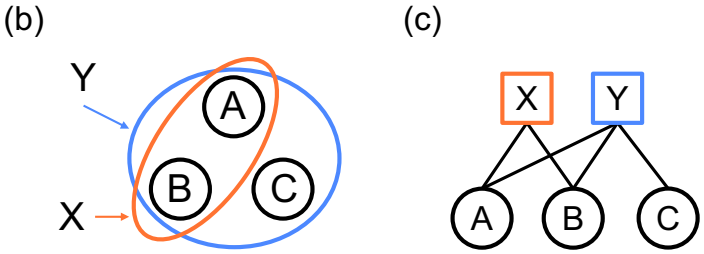


Figure 1.2: Representations of a social network involving higher-order interactions. (a) An example of two interactions among two or more individuals. (b) The corresponding hypergraph. A hypergraph is composed of nodes and hyperedges, where a hyperedge represents interaction among two or more nodes. (c) The corresponding bipartite graph. A bipartite graph is composed of a set of nodes and a set of interactions among two or more nodes, where an edge between a node and an interaction is present if the interaction involves the node.

1.1.2 Social Networks Involving Higher-order Interactions

Real-world social networks often involve higher-order interactions among more than two actors. Examples include group conversations in social contact networks [151, 212], multiple recipients of single emails [115], co-authoring in collaboration networks [171, 182, 229], and many more [28, 30]. Such networks involving higher-order interactions can be expressed as hypergraphs or bipartite graphs (Fig. 1.2).

A major method for analyzing empirical hypergraphs or bipartite graphs is to project them to dyadic networks (i.e., conventional networks, in which each edge connects a pair of nodes) and then analyze them [24, 170, 171]. However, a growing body of evidence suggests the limitations of describing the structure and dynamics of networks including higher-order interactions only using pairwise interactions [30,31,50,79,83,91,123,133,182,188,204,247]. In line with this, various measurements, dynamical process models, and theories have been developed for hypergraphs or bipartite graphs without using one-mode projection, particularly in recent years [28].

In general, a *reference model* for networks produces synthetic networks that preserve some specific properties of the given network and randomize other properties of the given network [52]. Regardless of the type of networks (e.g., dyadic networks, hypergraphs, or bipartite graphs), it is a recommended practice that one compares the structure and dynamics of a network at hand with those for randomized networks produced by reference models. Such an analysis helps us to reveal whether or not the given network has a certain structure relative to the random case and how the structural properties not preserved by the reference network model impacts dynamics on networks.

For dyadic networks, a family of standard reference model is the configuration models that preserve the degree of each node or its expectation [76, 155, 178]. The configuration models have been used for finding higher-order structural properties of various networks that the node's degree or its distribution does not imply

[33, 153, 173, 174, 237]. Furthermore, such findings have led to the development of reference models that preserve some higher-order properties of the input network, e.g., the degree correlation and the clustering coefficient of the node [26, 84, 111, 148, 150, 175, 180, 207, 211]. For hypergraphs, the properties of hyperedges as well as those of nodes are considered to affect their structure and dynamics. The existing reference models for hypergraphs preserve only up to the degree of each node and the size of each hyperedge (i.e., number of nodes that belong to each hyperedge) of a given hypergraph [37, 50, 178, 201, 202].

Chapter 4 proposes a family of reference models for hypergraphs, called the hyper dK-series. The original dK-series is a nested family of reference models that preserve local properties of nodes of the given dyadic network [84, 148, 180]. The hyper dK-series extends the dK-series to the case of hypergraphs. The hyper dK-series preserves up to the individual node's degree, node's degree correlation, node's redundancy coefficient, and/or the hyperedge's size depending on the parameter values. Then, we showcase its use in investigating epidemic spreading [63] and evolutionary game dynamics [18] models on hypergraphs. The contents of this chapter have been published in a paper [166].

One of the popular domains of social networks involving higher-order interactions may be scientific collaboration networks. In fact, scientific research has increasingly relied on teamwork over the last decades [75, 251]. For example, the fraction of scientific papers written by teams of researchers and the number of authors in a scientific paper have increased over the last century on average [241].

Various factors affect outcomes of scientific teamwork, including the team size (i.e., the number of authors of a paper) [240, 241], internationality (i.e., the number of countries involved in a paper) [54], interdisciplinarity (i.e., the number of disciplines of authors involved in a paper) [125], ethnic diversity (i.e., the number of ethnicities involved in a paper) [17], and team freshness (i.e., fraction of author pairs who co-authored at least one paper before the paper) [250]. In addition, quantitative approaches to scientific collaboration networks have contributed to the understanding of patterns of collaborations among researchers [172, 251] and their relations to research productivity (e.g., the number of published papers or the number of citations received by published papers) of researchers [9, 10, 100, 223, 233].

A universal trend in modern scientific teamwork is that researchers from different institutions increasingly collaborate with each other [12, 60, 109]. Such teams tend to produce papers with higher citation impact compared to those written by teams confined to a single institution [109]. The patterns of co-authorships between researchers from different institutions have been characterized by quantitive analyses of collaboration networks among institutions [47, 152, 243]. Grant collaboration involving multiple institutions is also a growing trend [2, 147, 158]. Ma et al. analyzed a British collaboration network among institutions in which edges represent partnerships between two institutions in funded research projects [147]. The authors found that universities with many edges tend to be densely connected to each other, forming a rich club. Analyses of such grant collaboration networks may inform the government and other stakeholders on research funding allocation among institutions [215]. Note that Ma et al. investigated a dyadic collaboration network of research grants using the one-mode projection [147].

Chapter 5 analyzes bipartite networks of institutions and collaborative grants to investigate the patterns of grant collaborations between two or more institutions. Using publicly available data from the National Science Foundation, we construct a bipartite network of institutions and collaborative grants. By extending the concept and algorithms of the rich club for dyadic networks to the case of

bipartite networks, we find rich clubs both in the entire bipartite network and the bipartite subnetwork induced by the collaborative grants involving a given number of institutions from two to five. Then, we find that the collaborative grants within rich clubs tend to be more productive in a per-dollar sense than the control. Our results highlight the advantages of grant collaborations the institutions in rich clubs. The contents of this chapter have been published online [167].

Chapter 2

Random Walk Sampling in Social Networks involving Private Nodes

2.1 Introduction

The re-weighted random walk [85, 86] does not assume *private nodes*, that do not publish their neighbors' data when they are queried in empirical social networks. Private nodes raise practical problems when one attempts to apply existing algorithms to empirical social networks. First, how do we deal with private nodes to obtain a sample sequence via a simple random walk? If a walker visits a private node, one can handle an exception wherein the neighbors' data of the node are not retrievable by jumping to some public user sampled previously. However, if one performs such exceptional processes, the sample sequence typically loses the Markov property, which prevents us from obtaining unbiased estimates of properties. There is another serious problem. A temporary solution to problems in the sampling phase is to not visit private nodes, as in the case study of random walks on the Facebook graph [85, 86]. However, if a walker does not traverse private nodes, the conventional framework [85, 86], which attempts to correct only the sampling bias, is expected to induce biases due to private nodes in estimators.

In this chapter, we aim to provide a practical framework for estimating properties based on a random walk on social networks involving private nodes. To this end, we first make three assumptions with respect to private nodes and formalize two models for accessing graph data, called the ideal model and the hidden privacy model. The assumptions and access models are based on previous studies and our observations on empirical social networks involving private nodes. Then, we design a sampling algorithm based on a random walk and develop estimators for the network size (i.e., number of nodes), average degree, and density of the node label (e.g., fraction of nodes with a given label). Our framework may help to extend random walk-based estimators of properties of a network to the case of social networks involving private nodes.

2.1.1 Our contributions

This work has three main contributions. First, we develop a sampling algorithm that practically works on social networks involving private nodes (Section 2.4). We design a procedure of neighbor selection, which is a fundamental element in the sampling phase via a random walk on the network, and derive the sampling bias of each node induced by the walk. Then, for each access model, we describe how to calculate the weight for each sampled node, which is essential to correct the sampling bias. Furthermore, we propose a method to estimate the weight using a much smaller number of queries than the exact calculation method for

the hidden privacy model.

Second, we present estimators with reduced biases induced by private nodes for the network size, average degree, and density of the node label (Section 2.5). Existing estimators are expected to induce biases due to private nodes because the conventional framework assumes the correction of only sampling bias. In our framework, we re-weight each sampled node to attempt to correct both the sampling bias and the bias induced by private nodes. Furthermore, we theoretically show that the proposed estimators have approximately no bias induced by private nodes if all public nodes form one connected component of the original network (Theorems 3, 4, and 5).

Third, we validate the theoretical results and effectiveness of the proposed estimators using empirical social network datasets (Section 2.6). We show that the proposed estimators acceptably perform on the two empirical datasets involving private nodes. Specifically, for the Pokec social network dataset [216], the proposed estimators reduce biases induced by private nodes in the existing estimators by up to 92.6%. For the Facebook dataset [120], the proposed estimators provide reasonable estimates of the network size, average degree, and cumulative degree distribution of the Facebook graph as of 2010.

2.2 Related Work

Several studies have proposed random walk algorithms to improve the estimation accuracy or the efficiency of the number of queries over a simple random walk [132, 140, 141, 168, 190, 245, 255]. Ribeiro and Towsley proposed multidimensional random walks, which improve the estimation accuracy in the presence of disconnected connected components [190]. Lee et al. proposed the non-backtracking random walk algorithm, which improves the query efficiency while preserving the Markov property of the sample sequence [132]. Yi et al. proposed the random walk-based algorithm, which reduces the bias of estimators using the bootstrapping technique [245]. These algorithms assume social networks involving no private nodes. In this work, we extend a simple random walk to the case of social networks involving private nodes. Based on our work, it is not trivial but possible to extend these improved random walks to the case of social networks involving private nodes.

Re-weighted random walk is a special case of respondent-driven sampling (RDS) [85, 86]. RDS is a random walk-based sampling method for estimating the proportion of individuals in the hard-to-reach population in social surveys (e.g., the fraction of infected individuals and the fraction of injection drug users) [98, 199, 230]. In the context of the RDS, a private node corresponds to an individual who will not respond to a survey at all. Such individuals with no response are easily present in practical scenarios [81, 87, 106, 192, 238]. Several studies numerically investigated the bias of the estimator induced by no-response individuals [145, 194, 219]. Tomas and Gile numerically showed that the estimator is biased when the response rate changes depending on the degree of the individual and the presence or absence of the infection of the individual. Lu et al. numerically investigated the bias when each individual does not respond with a given probability and showed that changes in the probability little affect the bias [145]. Rocha et al. investigated the effect of the community structure on the bias when each individual does not respond with a given probability. In this work, in the terminology of the RDS, we assume that each individual independently does not respond to a survey at all with a given probability (see Assumption 2 in Section 2.3.2 for details). Then, we propose an estimator of the density of the node label

in social networks involving private nodes (Section 2.5.4). Note that the density of the node label corresponds to the proportion of individuals with a specific characteristic (e.g., infected individual or drug user) in the context of the RDS. We theoretically and numerically show that the proposed estimator has little bias induced by private nodes.

Private nodes are regarded as missing graph data in random walk-based estimators because we are not permitted to retrieve their neighbors' data. In this work, we assume that each node becomes a private node independently at random with a given probability (see Assumption 2 in Section 2.3.2 for details). Several studies investigated the effects of completely random missing nodes on the structural properties of a network [16, 56, 103, 117, 163, 209]. Albert et al. found the robustness of networks against randomly missing nodes (i.e., given randomly missing nodes, most of the remaining nodes form the largest connected component). Kossinets showed that the bias of the average degree between the original network and the remaining largest connected component increases linearly with the proportion of randomly missing nodes. In this work, we theoretically analyze these biases and design estimators to reduce them under specific assumptions and access models. There are two important findings in this work compared with these previous studies. First, although we are not allowed to retrieve the neighbors of a private node by querying the node, we can find the private node in its neighbors that are public nodes in social networks (see Assumption 1 in Section 2.3.2 for details). Second, we can reduce the bias induced by private nodes by modifying the weight for each sampled node if each node becomes a private node independently at random with a given probability.

2.3 Preliminaries

2.3.1 Definitions and notations

We represent a social network as a connected and undirected graph $G = (V, E)$ that consists of a set of nodes $V = \{v_1, ..., v_n\}$ and a set of edges E, where n is the number of nodes. We denote by $\Gamma(i) = \{v_j \mid (v_i, v_j) \in E\}$ a set of neighbors of node v_i. Let $d_i = |\Gamma(i)|$ denote the degree (i.e., the number of neighbors) of node v_i and $D = \sum_{i=1}^{n} d_i$ denote the sum of degrees. We define the average degree of G as $d_{\text{avg}} = D/n$. Each node v_i is associated with a label $l(i)$. Examples of the node label are as follows: the degree, age, or gender of node v_i [85, 86, 190, 216]; v_i is a social bot or not [78, 92, 228]; and v_i is a drug user or not [98, 199]. Let $1_{\{cond\}}$ denote an indicator function that returns 1 if a condition *cond* holds and 0 otherwise. We define the density of node label l as $\rho(l) = \sum_{i=1}^{n} 1_{\{l(i)=l\}}/n$.

Each node v_i has a privacy label $l_{\text{pri}}(i) \in \{\text{public}, \text{private}\}$. We distinguish the privacy label $l_{\text{pri}}(i)$ from the label $l(i)$ for each node v_i. We call a node that has a private label a private node and call a node that has a public label a public node. The set of privacy labels of all the nodes is denoted by $\mathcal{L}_{\text{pri}} = \{l_{\text{pri}}(i)\}_{i=1}^{n}$.

We refer to connected subgraphs that consist of public nodes on G as *public clusters*. Let $\{C_j\}_j$ denote a set of public clusters and $C^* = (V^*, E^*)$ denote the largest public cluster. Let $n^* = |V^*|$ denote the number of nodes in C^*. We call the neighbors of a node that are public nodes the public neighbors of the node. We denote by $d_i^* = |\{v_j \in V^* \mid (v_i, v_j) \in E^*\}|$ the *public degree* (i.e., the number of public neighbors) of a node $v_i \in V^*$. Let $D^* = \sum_{v_i \in V^*} d_i^*$ denote the sum of public degrees. We define the average degree of C^* as $d_{\text{avg}}^* = D^*/n^*$. We also define the density of node label l of the largest public cluster as $\rho^*(l) = \sum_{v_i \in V^*} 1_{\{l(i)=l\}}/n^*$.

Figure 2.1 shows an example of a graph with privacy labels. Let $v_i = i$ for

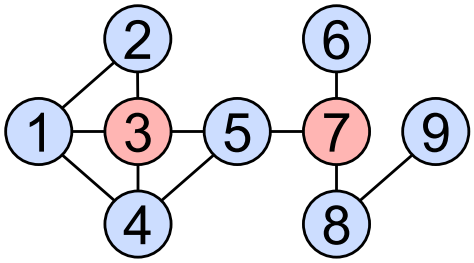


Figure 2.1: An example of a social network with privacy labels. Nodes 3 and 7 are private nodes, and all other nodes are public nodes.

$i = 1, 2, \ldots, 9$ in Fig. 2.1. Nodes v_3 and v_7 are private nodes, and all other nodes are public nodes. Public neighbors of node v_1 are nodes v_2 and v_4. There are three public clusters, C_1, C_2, and C_3, and C_1 is the largest public cluster C^*:

- $C_1 = (\{v_1, v_2, v_4, v_5\}, \{(v_1, v_2), (v_1, v_4), (v_4, v_5)\})$
- $C_2 = (\{v_8, v_9\}, \{(v_8, v_9)\})$
- $C_3 = (\{v_6\}, \{\})$.

It holds that $n^* = 4, d_1^* = 2, d_2^* = 1, d_4^* = 2, d_5^* = 1, D^* = 6, d_{\text{avg}}^* = 3/2$. Suppose that we are interested in the degree as the node label (i.e., $l(i) = d_i$ for each node v_i). Then, we have $l(1) = 3$, $l(2) = 2$, $l(3) = 4$, $l(4) = 3$, $l(5) = 3$, $l(6) = 1$, $l(7) = 3$, $l(8) = 2$, and $l(9) = 1$. In this case, we have $\rho(3) = 4/9$ and $\rho^*(3) = 3/4$.

2.3.2 Assumptions

We make the following three assumptions.

1. *If we query a public node, the indices of all its neighbors are available.* While we are not allowed to retrieve the neighbors' data of a private node by querying the node, we are allowed to retrieve the node from its neighbors that are public nodes under this assumption. For example, when one queries private node v_3 in Fig. 2.1, its neighbors are not retrievable; however, when one queries node v_5, all its neighbors, i.e., v_3, v_4, and v_7, are retrievable. We empirically find that this assumption sufficiently holds in practical scenarios: (i) the Facebook graph as of the previous study [86] satisfied this assumption; (ii) the public interfaces of Twitter as of December 2021 satisfy this assumption [221, 222]; and (iii) in the context of social surveys, even if an individual decides not to respond to the survey at all, the individual is encouraged to participate in the survey by its participating neighbors.

2. *Each node independently becomes a private node with probability p and becomes a public node otherwise, where $0 \leq p < 1$.* Intuitively, private nodes tend to have low degrees under this assumption. This is because the degree distribution of a social network is typically biased to low degrees [15, 85, 86, 122, 154]. We validate the effectiveness of our estimators designed under this assumption by using social network datasets involving real private nodes in Sections 2.6.3 and 2.6.4.

3. *We have access to some arbitrary node in the largest public cluster of G to begin our random walk.* Private nodes restrict a set of public nodes that a

walker is allowed to reach on the network. For example, if one selects node v_5 as a seed in Fig. 2.1, private node v_7 inhibits a walker from reaching public nodes v_6, v_8, and v_9. Under this assumption, a waller is allowed to traverse nodes on the largest public cluster of G. We do not consider the number of queries generated for selecting a seed of our random walk from a set of nodes on the largest public cluster. This is because we consider that this number is sufficiently small. We discuss the validity of this assumption in practical scenarios in Section 2.6.6.

2.3.3 Access models

We define access models for accessing graph G. We extend the standard model [49,85,86,190] to access models involving private nodes. Suppose we queried node v_i. If node v_i is a public node, then the neighbors' data of v_i and the label of v_i, i.e., $l(i)$, are available. If node v_i is a private node, then the neighbors' data of v_i and its node label $l(i)$ are not available[1]. We consider two models for available neighbors' data of a queried public node v_i: the *ideal model* and the *hidden privacy model*.

In the ideal model, when one queries node v_i, the indices and privacy labels of all the neighbors of v_i are available. For example, when we query a public node v_4 in Fig. 2.1, we obtain the set $\{(v_1, \text{public}), (v_3, \text{private}), (v_5, \text{public})\}$. As empirical evidence, the Facebook graph as of the previous study [85,86] corresponds to this access model.

In the hidden privacy model, when we query node v_i, the indices of all the neighbors of v_i are available but their privacy labels are not available. For example, when we query a public node v_4 in Fig. 2.1, we obtain the set $\{v_1, v_3, v_5\}$. Real-world scenarios corresponding to this access model include the public interfaces of Twitter as of December 2021 [221, 222] and real social networks in the context of social surveys [98, 199, 230].

2.3.4 Markov chain

We introduce the basics of a Markov chain for the theoretical analysis of random walk-based estimators. First, we describe the stationary distribution of a Markov chain, which serves to derive the sampling bias induced by a random walk. Let $\boldsymbol{P} = (P_{i,j})_{i,j \in S}$ denote the transition probability matrix of a Markov chain on a finite state space S. If it holds that $\pi_j = \sum_{i \in S} \pi_i P_{i,j}$ for all $j \in S$, a vector $\boldsymbol{\pi} = (\pi_i)_{i \in S}$ is the stationary distribution of the chain. If all states in an irreducible Markov chain are ergodic, then the chain is said to be ergodic (see [138] for formal definitions). The following theorem holds in regard to the stationary distribution $\boldsymbol{\pi}$ of a Markov chain.

Theorem 1. *[138] If a Markov chain is ergodic, the stationary distribution $\boldsymbol{\pi}$ uniquely exists.*

Then, we review the strong law of large numbers for a Markov chain, which ensures that an estimator converges almost surely to its expected value with respect to the stationary distribution [48, 132]:

Theorem 2. *[110, 191] Let $\{X_k\}_{k=1}^r$ be an ergodic Markov chain with the stationary distribution $\boldsymbol{\pi}$ on a finite state space S. For any function $f : S \to \mathbb{R}$, a quantity $\sum_{k=1}^r f(X_k)/r$ converges to the expected value with respect to $\boldsymbol{\pi}$, i.e.,*

[1]We assume that the response is an empty set when one queries a private node.

$\mathbb{E}_{\boldsymbol{\pi}}[f] \triangleq \sum_{i \in S} \pi_i f(i)$, *almost surely as* $r \to \infty$ *regardless of the initial distribution of the chain.*

2.4 Sampling Algorithm

In this section, we first design a sampling algorithm based on a random walk considering private nodes in each access model. First, we describe how to select a neighbor to be traversed from the current node to obtain a sample sequence that has the Markov property. Then, we derive the sampling bias of each node induced by our random walk. Finally, we describe a method for calculating the public degree of each sampled node to correct the sampling bias.

2.4.1 Neighbor selection

In a simple random walk, one repeatedly moves to a neighbor that is uniformly and randomly selected from a set of neighbors of the current node. If a walker visits a private node in this method, two main problems generally occur for the existing estimators based on a simple random walk. First, we are not allowed to continue the walk because neighbors of the private node are not retrievable. Although we can restart the walk from an arbitrary public node previously sampled, the sample sequence loses the Markov property by performing such exception handling. Second, it is difficult to correct the sampling bias of private nodes because their degrees and public degrees are unclear.

We extend a simple random walk to the case of social networks involving private nodes. We collect a sequence of indices of r sampled nodes, denoted by $(x_1, x_2, \ldots, x_r)$, as follows. We select a seed $v_{x_1} \in C^*$, which is a node on the largest public cluster, according to Assumption 3. For the k-th sampled node ($k = 1, \ldots, r-1$), we first obtain a set of neighbors of v_{x_k}, i.e., $\Gamma(x_k)$ by querying v_{x_k}. Then, we uniformly and randomly select node $u \in \Gamma(x_k)$. If u is a public node, the walker moves to u as the next sampled node $v_{x_{k+1}}$, otherwise, we uniformly and randomly select a node as u from the set $\Gamma(x_k)$ again. In the ideal model, where the privacy labels of all the neighbors of a queried node are available, we check if a selected neighbor u is public without querying node u. In the hidden privacy model, where the privacy labels of neighbors of a queried node are not available, we judge the privacy label of node u by additionally querying node u.

A walker that locates at v_{x_k} has at least one public neighbor for each $k = 1, \ldots, r$ if and only if v_{x_1} belongs to the largest public cluster comprising multiple public nodes. Specifically, v_{x_1} has a public neighbor that belongs to the largest public cluster; v_{x_k} has the public neighbor $v_{x_{k-1}}$ for each $k = 2, \ldots, r$. Therefore, the neighbor selection procedure for v_{x_k} terminates with probability 1 for each $k = 1, \ldots, r$; consequently, our random walk with length r finishes with probability 1.

2.4.2 Sampling bias

We derive the sampling bias induced by our random walk. Let the probability that an event A will occur be denoted by $\Pr[A]$. We define the distribution induced by the sequence of sampled indices as $\boldsymbol{\pi}_r = (\Pr[x_r = i])_{i=1}^n$. We show that each node on the largest public cluster is sampled in proportion to the public degree via our random walk.

Lemma 1. *The vector* $\boldsymbol{\pi}_r$ *converges to* $\boldsymbol{\pi} = (p_i)_{i=1}^n$ *after many steps of our random walk, where* $p_i = 1_{\{v_i \in V^*\}} d_i^* / D^*$.

Algorithm 1 Our random walk in the hidden privacy model.

Input: Seed $v_{x_1} \in C^*$. Sample size r.
Output: Sampling list R.

$R \leftarrow$ an empty list.
for $k = 1$ to r **do**
  Query v_{x_k} and obtain the set $\Gamma(x_k)$.
  $d_{x_k} \leftarrow |\Gamma(x_k)|$.
  $\hat{d}^*_{x_k} \leftarrow 0$.
  $R \leftarrow$ append $(x_k, d_{x_k}, \hat{d}^*_{x_k})$.
  if v_{x_k} has been visited for the first time **then**
    $a_{x_k} \leftarrow 0$.
    $b_{x_k} \leftarrow 0$.
  flag $\leftarrow$ False.
  while flag is False **do**
    $u \leftarrow$ a neighbor uniformly and randomly chosen from $\Gamma(x_k)$.
    $b_{x_k} \leftarrow b_{x_k} + 1$.
    if u is a public node **then**
      $v_{x_{k+1}} \leftarrow u$
      $a_{x_k} \leftarrow a_{x_k} + 1$
      flag $\leftarrow$ True.
for $k = 1$ to r **do**
  $\hat{d}^*_{x_k} \leftarrow d_{x_k} \frac{a_{x_k}}{b_{x_k}}$
return R

Proof. First, it holds that $\Pr[x_r = i] = 0$ for each node $v_i \in V \backslash V^*$ because our random walk never traverses nodes that do not belong to the largest public cluster C^*. Then, for each node $v_i \in V^*$, we show that $Pr[x_r = i]$ converges to d^*_i / D^* after many steps of our random walk. Our random walk has the transition probability matrix $\boldsymbol{P} = (P_{i,j})_{v_i, v_j \in V^*}$ defined as

$$P_{i,j} = \begin{cases} 1/d^*_i & \text{if } (v_i, v_j) \in E^*, \\ 0 & \text{(otherwise).} \end{cases}$$

The corresponding Markov chain is ergodic because it is equivalent to a simple random walk on the largest public cluster C^*. Note that a simple random walk on a connected graph is ergodic [138, 144]. Hence, the stationary distribution uniquely exists because of Theorem 1. The vector $(p_i)_{i=1}^n$ satisfies the definition of the stationary distribution. The probability $\Pr[x_r = i]$ converges to the corresponding stationary distribution after many steps of our random walk. □

2.4.3 Calculating the public degree of each sampled node

We calculate the public degree of each sampled node to correct the sampling bias that is attributable to the public degree. In the ideal model, we exactly calculate the public degree of each sampled node without additional queries because the privacy labels of all the neighbors of each sampled node are available. Conversely, in the hidden privacy model, the exact calculation needs a considerable number of additional queries to obtain the privacy labels of all the neighbors of each sampled node.

We propose a method to estimate the public degree of each sampled node without additional queries in the hidden privacy model. Algorithm 1 describes

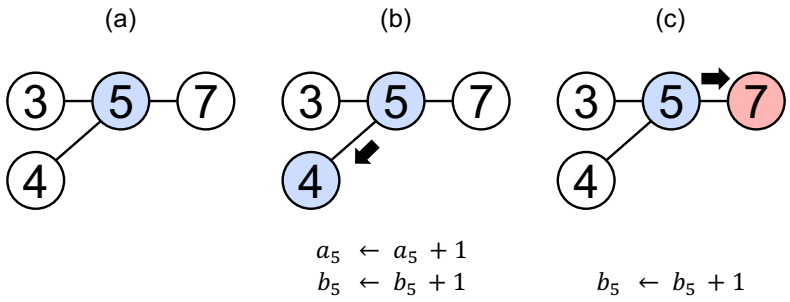


Figure 2.2: An example of the procedure for updating the quantities a_{x_k} and b_{x_k} when the walker is located at node v_{x_k} in the hidden privacy model.

our random walk using the proposed method for estimating the public degree of each sampled node. The proposed method utilizes the history of neighbor selections generated by our random walk. Specifically, we record two quantities a_{x_k} and b_{x_k} for each sampled node v_{x_k}. The quantity a_{x_k} is the total number of times public neighbors of v_{x_k} are selected. The quantity b_{x_k} is the total number of times neighbors of v_{x_k} are selected. For example, we consider the case in which a walker is located at node v_5 in the graph shown in Fig. 2.1. If we query node v_5, we obtain the set $\Gamma(5) = \{v_3, v_4, v_7\}$ (see Fig. 2.2(a)), where we note that the privacy labels of v_3, v_4, and v_7 are not included in the query response in the hidden privacy model. Then, we uniformly and randomly select a node from $\Gamma(5)$. When we select node v_4, we increase a_5 and b_5 by one each because v_4 is a public node (see Fig. 2.2(b)). When we select node v_7, we increase b_5 by one because v_7 is a private node (see Fig. 2.2(c)). After completing our random walk of length r, we calculate an estimator, $\hat{d}^*_{x_k}$, of the public degree of each sampled node v_{x_k} as

$$\hat{d}^*_{x_k} \triangleq d_{x_k} \frac{a_{x_k}}{b_{x_k}}.$$

Note that it holds that $b_{x_k} > 0$ for each $k = 1, \ldots, r$ because at least one neighbor selection is performed for each sampled node.

We ensure that the estimator $\hat{d}^*_{x_k}$ is an unbiased estimator of the public degree of v_{x_k}.

Lemma 2. *For each sampled node v_{x_k}, the estimator $\hat{d}^*_{x_k}$ converges to the true value $d^*_{x_k}$ after many steps of our random walk.*

Proof. Let $X_{x_k}(l)$ denote a random variable that returns 1 if a public neighbor of v_{x_k} is selected at the l-th trial of neighbor selections at v_{x_k} and returns 0 otherwise, where $l = 1, \ldots, b_{x_k}$ and it holds that $\sum_{l=1}^{b_{x_k}} X_{x_k}(l) = a_{x_k}$. It holds that $\Pr[X_{x_k}(l) = 1] = d^*_{x_k}/d_{x_k}$ because $X_{x_k}(l)$ follows a Bernoulli distribution for each l. Therefore, we have $\mathbb{E}[\hat{d}^*_{x_k}] = d_{x_k} d^*_{x_k}/d_{x_k} = d^*_{x_k}$. A sequence of random variables $\{X_{x_k}(l)\}_{l=1}^{b_{x_k}}$ is drawn from a process of independent and identically distributed trials. Therefore, the estimator $\hat{d}^*_{x_k}$ converges to its expected value $\mathbb{E}[\hat{d}^*_{x_k}] = d^*_{x_k}$ after many steps of our random walk because of the law of large numbers. □

Then, we show that our random walk using the proposed method theoretically generates much fewer queries than that using the exact calculation method. In the exact method, one queries all the neighbors of each sampled node to exactly calculate the public degree. When a node is visited for the first time, one can

reduce the number of queries in future steps by saving the neighbor data of the node. However, for simplicity, we do not consider this saving of the neighbors' data of nodes queried once in the theoretical analysis. We denote by $Q(k)$ the number of queries generated at the k-th sampled node v_{x_k} by the exact method. We denote by $Q'(k)$ the number of queries generated at the k-th sampled node by the proposed method. Let $Q = (\sum_{k=1}^{r} Q(k))/r$ denote the ratio of the number of queries using the exact method to the sample size. Let $Q' = (\sum_{k=1}^{r} Q'(k))/r$ denote the ratio of the number of queries using the proposed method to the sample size. We have the following lemma.

Lemma 3. *The expected value of Q with respect to $\boldsymbol{\pi}$ is given by*

$$\mathbb{E}_{\boldsymbol{\pi}}[Q] = \frac{1}{D^*} \sum_{v_i \in V^*} d_i^* d_i.$$

The expected value of Q' with respect to $\boldsymbol{\pi}$ is given by

$$\mathbb{E}_{\boldsymbol{\pi}}[Q'] = \frac{1}{D^*} \sum_{v_i \in V^*} d_i.$$

Proof. It holds that $Q(k) = d_{x_k}$ because one queries all the neighbors of v_{x_k} in the exact method. Therefore, we have

$$\mathbb{E}_{\boldsymbol{\pi}}[Q] = \mathbb{E}_{\boldsymbol{\pi}}[Q(k)] = \sum_{v_i \in V^*} \frac{d_i^*}{D^*} \mathbb{E}[Q(k)|x_k = i] = \frac{1}{D^*} \sum_{v_i \in V^*} d_i^* d_i.$$

The first equation holds because of the linearity of expectation. The second equation holds because of the law of total expectation and Lemma 1.

The quantity $Q'(k)$ follows the geometric distribution with success probability $d_{x_k}^*/d_{x_k}$ because we repeatedly query neighbors of v_{x_k} uniformly and randomly until a public neighbor of v_{x_k} is first selected in the proposed method. Thus, it holds that $\mathbb{E}[Q'(k)] = d_{x_k}/d_{x_k}^*$. Then, we have

$$\mathbb{E}_{\boldsymbol{\pi}}[Q'] = \mathbb{E}_{\boldsymbol{\pi}}[Q'(k)] = \sum_{v_i \in V^*} \frac{d_i^*}{D^*} \frac{d_i}{d_i^*} = \frac{1}{D^*} \sum_{v_i \in V^*} d_i.$$

□

Intuitively, $\sum_{v_i \in V^*} d_i^*$ is the order of $\sum_{v_i \in V} d_i$ and $\sum_{v_i \in V^*} d_i^* d_i$ is the order of $\sum_{v_i \in V} d_i^2$. The sum of squares of degrees is much larger than the sum of degrees for a large-scale network with a heavy-tailed degree distribution [177]. Therefore, Lemma 3 implies that the proposed method generates much fewer queries than the exact method.

2.5 Estimators

In the conventional framework [85, 86], one re-weights each sampled node using its public degree to correct the sampling bias. Therefore, the existing estimators converge to the quantities of the largest public cluster. When the original graph comprises only public nodes, as assumed in the conventional framework [85, 86], the expected values of the estimators are equal to the quantities of the original graph (i.e., true values). However, when there are private nodes on the original graph, the existing estimators are generally expected to have the bias induced by private nodes.

It is not trivial to reduce the bias induced by private nodes. We could easily correct the bias if we were to find the exact probability p or the proportion of private nodes. However, the quantities are typically unknown to third parties. Furthermore, it is difficult to apply existing methods for estimating the probability p or the proportion of private nodes based on the privacy labels of sampled nodes (Section 3.C.3 in Ref. [86], Section 3.2 in Ref. [112] and Section 4.2.3 in Ref. [190]). This is because private nodes are not included in the sample sequence.

We propose estimators with reduced biases induced by private nodes for the network size (i.e., the number of nodes), average degree, and density of the node label. We re-weight each sampled node using both its degree and its public degree to reduce the bias induced by private nodes. We modify the weight for each sampled node based on the mathematical property that the public degree of a node follows the binomial distribution with parameters of its degree and $1-p$ with respect to the set of privacy labels of nodes under Assumption 2. We theoretically show that the proposed estimators have approximately no bias induced by private nodes if all public nodes belong to the largest public cluster of the original network. In the following, for each of the three properties, we first introduce the existing estimator and then describe our estimator. Then, we describe heuristic estimators for the percentage of private nodes, which combine the existing and proposed estimators for the network size and average degree each.

2.5.1 Network size

Existing estimator

The node collision estimator is effective for estimating the network size [97, 112]. In the estimator, one counts the number of collisions in the indices of pairs of the sampled nodes whose ordinal numbers in the sample sequence are far away. Such pairs of sampled nodes are regarded as being sampled independently of each other from the stationary distribution [97, 112].

Formally, the existing estimator of the network size is defined as follows. Let $I = \{(k,l) \mid m \leq |k-l| \wedge 1 \leq k,\ l \leq r\}$ denote the set of integer pairs that are between 1 and r and at least a threshold m away. We set $m = 0.025r$, as in the previous study [97]. Let $\phi_{k,l}$ denote a variable that returns 1 if the indices of k-th and l-th sampled public nodes are the same, i.e., $x_k = x_l$ (this is called a collision) and returns 0 otherwise. One defines the average of the number of collisions Φ_{size}, the average of the weights to correct the sampling bias Ψ_{size}, and a size estimate $\hat{n}$ as

$$\Phi_{\text{size}} = \frac{1}{|I|} \sum_{(k,l)\in I} \phi_{k,l}, \quad \Psi_{\text{size}} = \frac{1}{|I|} \sum_{(k,l)\in I} \frac{d^*_{x_k}}{d^*_{x_l}}, \quad \hat{n} \triangleq \frac{\Psi_{\text{size}}}{\Phi_{\text{size}}}.$$

We have the following lemma, which is extended from the results shown in previous studies [97, 112] which assume that the original graph involves no private nodes.

Lemma 4. *The estimator $\hat{n}$ asymptotically converges to the size of the largest public cluster n^* after many steps of our random walk.*

Proof. First, we calculate the expected value with respect to $\boldsymbol{\pi}$ of Φ_{size}:

$$\mathbb{E}_{\boldsymbol{\pi}}\left[\Phi_{\text{size}}\right] = \mathbb{E}_{\boldsymbol{\pi}}[\phi_{k,l}] = \sum_{v_i \in V^*} \left(\frac{d^*_i}{D^*}\right)^2.$$

The first equation holds because of the linearity of expectation. The second equation holds because v_{x_k} and v_{x_l} are sampled independently of each other from the stationary distribution. Then, we calculate the expected value with respect to $\boldsymbol{\pi}$ of Ψ_{size}:

$$\mathbb{E}_{\boldsymbol{\pi}}\left[\Psi_{\text{size}}\right] = \mathbb{E}_{\boldsymbol{\pi}}\left[\frac{d^*_{x_k}}{d^*_{x_l}}\right] = \sum_{v_i \in V^*}\sum_{v_j \in V^*} \frac{d^*_i}{d^*_j}\frac{d^*_j}{D^*}\frac{d^*_i}{D^*} = n^* \sum_{v_i \in V^*}\left(\frac{d^*_i}{D^*}\right)^2.$$

Quantities Φ_{size} and Ψ_{size} intuitively converge to their respective expected values with respect to $\boldsymbol{\pi}$ after many steps of our random walk. Therefore, we conclude $\hat{n}$ asymptotically converges to n^*. □

We quantify the bias induced by private nodes of the expected value n^*. To this end, we derive the expected value of n^* with respect to the set of privacy labels $\mathcal{L}_{\text{pri}}$. Let $\mathbb{E}_{\text{pri}}[X]$ denote the expected value of a random variable X with respect to the set $\mathcal{L}_{\text{pri}}$. We approximate the expected value of n^* regarding the set $\mathcal{L}_{\text{pri}}$ under the condition that all the public nodes belong to the largest public cluster. Under this condition, it holds that $\Pr[v_i \in V^*] = \Pr[l_{\text{pri}}(i) = \text{public}] = 1 - p$ because of Assumption 2.

The following lemma holds regarding the expected value of the existing estimator.

Lemma 5. *If all the public nodes belong to the largest public cluster, we have*

$$\mathbb{E}_{pri}[n^*] = (1-p)n.$$

Proof. We define a random variable $X_{\text{size}}(i) = 1_{\{v_i \in V^*\}}$ for each node $v_i \in V$. Then, it holds that $n^* = \sum_{v_i \in V} X_{\text{size}}(i)$. The expected value of n^* with respect to $\mathcal{L}_{\text{pri}}$ under the given condition is given by

$$\mathbb{E}_{\text{pri}}[n^*] = \sum_{v_i \in V} \mathbb{E}_{\text{pri}}[X_{\text{size}}(i)] = \sum_{v_i \in V} \Pr[v_i \in V^*] = (1-p)n.$$

The first equation holds based on the linearity of expectation. The second equation holds based on the law of total expectation. □

Lemma 5 implies that the expected value of the existing estimator has the bias $1 - p$.

Proposed estimator

We modify the weight for each pair of sampled nodes (v_{x_k}, v_{x_l}) such that $(k, l) \in I$ to reduce the bias of the expected value induced by private nodes. Specifically, we define the average of the modified weights Ψ'_{size} and the proposed estimator $\hat{n}'$ as follows:

$$\Psi'_{\text{size}} = \frac{1}{|I|}\sum_{(k,l)\in I}\frac{d_{x_k}}{d^*_{x_l}}, \quad \hat{n}' \triangleq \frac{\Psi'_{\text{size}}}{\Phi_{\text{size}}}.$$

The following lemma holds in regard to the expected value of the proposed estimator.

Lemma 6. *The estimator $\hat{n}'$ asymptotically converges to*

$$\tilde{n} = n^* \frac{\sum_{v_i \in V^*} d^*_i d_i}{\sum_{v_i \in V^*} (d^*_i)^2}$$

after many steps of our random walk.

Proof. As with the proof of the Lemma 4, we have

$$\mathbb{E}_{\boldsymbol{\pi}}\left[\Psi'_{\text{size}}\right] = \mathbb{E}_{\boldsymbol{\pi}}\left[\frac{d_{x_k}}{d^*_{x_l}}\right] = n^* \sum_{v_i \in V^*} \frac{d^*_i d_i}{(D^*)^2}.$$

Quantities Φ_{size} and Ψ'_{size} intuitively converge to the respective expected values after many steps of our random walk. Therefore, $\hat{n}'$ asymptotically converges to $\tilde{n}$. □

When there are no private nodes on G, the following proposition regarding each estimator and each expected value holds.

Proposition 1. *When the original graph G involves no private nodes, two estimators, $\hat{n}$ and $\hat{n}'$, are equal, and two expected values, n^* and $\tilde{n}$, are equal to the true quantity n.*

Proof. When the original graph G involves no private nodes, it holds that $V^* = V$ and $d^*_i = d_i$ for each node $v_i \in V^*$. This is because the largest public cluster C^* is equivalent to the original graph G. Thus, it holds that $\hat{n} = \hat{n}'$ because of the definitions of the estimators. It also holds that $n^* = n$ and $\tilde{n} = n$ because of Lemmas 4 and 6. □

We show that the expected value $\tilde{n}$ of the proposed estimator reduces the bias induced by private nodes compared with the existing estimator. First, we derive the expected value with respect to the set $\mathcal{L}_{\text{pri}}$ of the public degree of a public node:

Lemma 7. *For any public node $v_i \in V^*$, we have*

$$\mathbb{E}_{pri}[d^*_i] = (1-p)d_i,$$
$$\mathbb{E}_{pri}\left[(d^*_i)^2\right] = (1-p)d_i[(1-p)d_i + p].$$

Proof. The public degree d^*_i follows the binomial distribution with parameters of the degree d_i and $1-p$ regarding the set $\mathcal{L}_{\text{pri}}$ because each neighbor of node v_i independently becomes public with the probability $1-p$ under Assumption 2. □

Then, we approximate the expected value of $\tilde{n}$ with respect to $\mathcal{L}_{\text{pri}}$ as a product of each expected value with respect to $\mathcal{L}_{\text{pri}}$ of each quantity in the denominator and numerator of $\tilde{n}$.

Theorem 3. *If all the public nodes belong to the largest public cluster, we have*

$$\mathbb{E}_{pri}[\tilde{n}] \approx \frac{\mathbb{E}_{pri}[n^*]\mathbb{E}_{pri}[\sum_{v_i \in V^*} d^*_i d_i]}{\mathbb{E}_{pri}[\sum_{v_i \in V^*} (d^*_i)^2]} = \alpha_p n,$$

where

$$\alpha_p = \frac{(1-p)\sum_{v_i \in V} (d_i)^2}{\sum_{v_i \in V} d_i[(1-p)d_i + p]}. \tag{2.1}$$

Proof. We define a random variables $X_{\text{size}}(i) = d^*_i d_i 1_{\{v_i \in V^*\}}$ and $Y_{\text{size}}(i) = (d^*_i)^2 1_{\{v_i \in V^*\}}$ for each node $v_i \in V$. Let $X_{\text{size}} = \sum_{v_i \in V^*} d^*_i d_i$ and $Y_{\text{size}} = \sum_{v_i \in V^*} (d^*_i)^2$. It holds that $X_{\text{size}} = \sum_{v_i \in V} X_{\text{size}}(i)$ and $Y_{\text{size}} = \sum_{v_i \in V} Y_{\text{size}}(i)$. We obtain the expected value of X_{size} with respect to $\mathcal{L}_{\text{pri}}$:

$$\mathbb{E}_{\text{pri}}[X_{\text{size}}] = \sum_{v_i \in V} \Pr[v_i \in V^*]\mathbb{E}_{\text{pri}}[d^*_i d_i] = (1-p)^2 \sum_{v_i \in V} (d_i)^2.$$

The second equation holds because of Lemma 7. We note that the degree d_i is constant with respect to $\mathcal{L}_{\text{pri}}$. Similarly, the expected value of Y_{size} with respect to $\mathcal{L}_{\text{pri}}$ is obtained as follows:

$$\mathbb{E}_{\text{pri}}[Y_{\text{size}}] = \sum_{v_i \in V} \Pr[v_i \in V^*] \mathbb{E}_{\text{pri}}[(d_i^*)^2] = (1-p)^2 \sum_{v_i \in V} d_i[(1-p)d_i + p].$$

Theorem 3 holds because of the above equations and Lemma 5. □

We empirically find that the coefficient α_p is almost equal to 1 for various values of p in different social networks (see Section 2.6.1 for details). This is because the sum of squares of degrees $\sum_{v_i \in V} (d_i)^2$ is considerably larger than the sum of degrees $\sum_{v_i \in V} d_i$ in large-scale networks with heavy-tailed degree distributions [177]. Therefore, Theorem 3 implies that the expected value of the proposed estimator has approximately no bias with respect to a random set of privacy labels of nodes if all the public nodes belong to the largest public cluster.

In practice, it rarely holds true that all public nodes belong to the largest public cluster of a large-scale social network. Hence, the expected values of the existing and proposed estimators typically have the biases induced by public nodes that do not belong to the largest public cluster. However, most public nodes belong to the largest public cluster of real social networks under Assumption 2. This is supported by the nature that real-world networks with heavy-tailed degree distributions have high robustness for the connected component against random removal of a set of nodes [16]. In fact, we numerically find that the proposed estimators have smaller biases induced by private nodes than the existing estimators in real social network datasets (see Section 2.6.2 for details).

2.5.2 Average degree

Existing estimator

An existing estimator of the average degree [62,85,86], denoted by $\hat{d}_{\text{avg}}$, is defined as

$$\Phi_{\text{avg}} = \frac{1}{r} \sum_{k=1}^{r} \frac{1}{d^*_{x_k}}, \quad \hat{d}_{\text{avg}} \triangleq \frac{1}{\Phi_{\text{avg}}}.$$

We have the following lemma derived from the previous study [62] which assumes that the original graph involves no private nodes.

Lemma 8. *The estimator $\hat{d}_{avg}$ converges to the average degree of the largest public cluster d^*_{avg} after many steps of our random walk.*

Proof. We calculate the expected value of Φ_{avg} with respect to $\boldsymbol{\pi}$ as follows:

$$\mathbb{E}_{\boldsymbol{\pi}}[\Phi_{\text{avg}}] = \mathbb{E}_{\boldsymbol{\pi}}\left[\frac{1}{d^*_{x_k}}\right] = \sum_{v_i \in V^*} \frac{d_i^*}{D^*} \frac{1}{d_i^*} = \frac{1}{d^*_{\text{avg}}}.$$

The quantity Φ_{avg} converges to the expected value after many steps because of Theorem 2. Therefore, $\hat{d}_{\text{avg}} = 1/\Phi_{\text{avg}}$ converges to d^*_{avg} after many steps of our random walk. □

Then, we quantify the bias of the expected value of the existing estimator induced by private nodes. We approximate the expected value of d^*_{avg} with respect to $\mathcal{L}_{\text{pri}}$ as a product of the expected value with respect to $\mathcal{L}_{\text{pri}}$ of each quantity in the denominator and numerator of d^*_{avg}.

Lemma 9. *If all the public nodes belong to the largest public cluster, we have*

$$\mathbb{E}_{pri}[d^*_{avg}] \approx \frac{\mathbb{E}_{pri}[D^*]}{\mathbb{E}_{pri}[n^*]} = (1-p)d_{avg}.$$

Proof. We define a random variable $X_{\mathrm{avg}}(i) = d^*_i 1_{\{v_i \in V^*\}}$ for each node $v_i \in V$. It holds that $D^* = \sum_{v_i \in V} X_{\mathrm{avg}}(i)$. The expected value with respect to $\mathcal{L}_{\mathrm{pri}}$ of D^* is derived as

$$\mathbb{E}_{\mathrm{pri}}[D^*] = \sum_{v_i \in V} \Pr[v_i \in V^*]\mathbb{E}_{\mathrm{pri}}[d^*_i] = (1-p)^2 D.$$

Consequently, it follows that Lemma 9 holds because of the above equation and Lemma 5. □

Lemma 9 implies that the expected value of the existing estimator has the bias $1-p$.

Proposed estimator

We modify the weight for each sampled node to reduce the bias of the expected value induced by private nodes. We define the average of the modified weights Ψ'_{size} and the proposed estimator $\hat{d}'_{\mathrm{avg}}$ as follows:

$$\Phi'_{\mathrm{avg}} = \frac{1}{r}\sum_{k=1}^{r} \frac{1}{d_{x_k}}, \quad \hat{d}'_{\mathrm{avg}} \triangleq \frac{1}{\Phi'_{\mathrm{avg}}}.$$

We have the following lemma regarding the proposed estimator.

Lemma 10. *The estimator $\hat{d}'_{avg}$ converges to*

$$\tilde{d}_{avg} = \frac{D^*}{\sum_{v_i \in V^*} d^*_i/d_i}$$

after many steps of our random walk.

Proof. We calculate the expected value of Φ'_{avg} with respect to $\boldsymbol{\pi}$ as follows:

$$\mathbb{E}_{\boldsymbol{\pi}}[\Phi'_{\mathrm{avg}}] = \mathbb{E}_{\boldsymbol{\pi}}\left[\frac{1}{d_{x_k}}\right] = \sum_{v_i \in V^*} \frac{d^*_i}{D^*}\frac{1}{d_i} = \frac{1}{\tilde{d}_{\mathrm{avg}}}.$$

The quantity Φ'_{avg} converges to the expected value because of Theorem 2, and hence, the estimator $\hat{d}'_{\mathrm{avg}}$ converges to $\tilde{d}_{\mathrm{avg}}$ after many steps of our random walk. □

The following proposition holds as well as Proposition 1.

Proposition 2. *When the original graph G involves no private nodes, two estimators $\hat{d}_{avg}$ and $\hat{d}'_{avg}$ are equal, and two expected values, d^*_{avg} and $\tilde{d}_{avg}$, are equal to the true quantity d_{avg}.*

Finally, we have the following theorem.

Theorem 4. *If all the public nodes belong to the largest public cluster, we have*

$$\mathbb{E}_{pri}[\tilde{d}_{avg}] \approx \frac{\mathbb{E}_{pri}[D^*]}{\mathbb{E}_{pri}\left[\sum_{v_i \in V^*} d^*_i/d_i\right]} = d_{avg}.$$

Proof. We define a random variable $\tilde{X}_{\text{avg}}(i) = 1_{\{v_i \in V^*\}} d_i^*/d_i$ for each node $v_i \in V$. Let $\tilde{X}_{\text{avg}} = \sum_{v_i \in V^*} d_i^*/d_i$. It holds that $\tilde{X}_{\text{avg}} = \sum_{v_i \in V} \tilde{X}_{\text{avg}}(i)$. We obtain the expected value of $\tilde{X}_{\text{avg}}$ with respect to $\mathcal{L}_{\text{pri}}$:

$$\mathbb{E}_{\text{pri}}[\tilde{X}_{\text{avg}}] = \sum_{v_i \in V} \Pr[v_i \in V^*] \mathbb{E}_{\text{pri}} \left[\frac{d_i^*}{d_i} \right] = (1-p)^2 n.$$

Theorem 4 holds because of the equation of $\mathbb{E}_{\text{pri}}[D^*] = (1-p)^2 D$ and the above equation. □

2.5.3 Node's label density

Existing estimator

An existing estimator of the density of node label l, denoted by $\hat{\rho}(l)$, is defined as follows [85, 86, 190]:

$$\Phi_{\text{label}} = \frac{1}{r} \sum_{k=1}^{r} \frac{1_{\{l(x_k)=l\}}}{d_{x_k}^*}, \quad \hat{\rho}(l) \triangleq \frac{\Phi_{\text{label}}}{\Phi_{\text{avg}}}.$$

Note that the label of interest l of the node does not include the privacy label of the node because the sample sequence contains only public nodes.

We have the following lemma derived from the previous study [190] which assumes that the original graph involves no private nodes.

Lemma 11. *The estimator $\hat{\rho}(l)$ converges to the density of node label l of the largest public cluster $\rho^*(l)$ after many steps of our random walk.*

Proof. We calculate the expected value of Φ_{label} with respect to $\boldsymbol{\pi}$ as follows:

$$\mathbb{E}_{\boldsymbol{\pi}}[\Phi_{\text{label}}] = \mathbb{E}_{\boldsymbol{\pi}} \left[\frac{1_{\{l(x_k)=l\}}}{d_{x_k}^*} \right] = \sum_{v_i \in V^*} \frac{d_i^*}{D^*} \frac{1_{\{l(i)=l\}}}{d_i^*} = \frac{1}{D^*} \sum_{v_i \in V^*} 1_{\{l(i)=l\}}.$$

The quantity Φ_{label} converges to the expected value $\mathbb{E}_{\boldsymbol{\pi}}[\Phi_{\text{label}}] = \sum_{v_i \in V^*} 1_{\{l(i)=l\}}/D^*$ after many steps of our random walk because of Theorem 2. The quantity Φ_{avg} also converges to the expected value $\mathbb{E}_{\boldsymbol{\pi}}[\Phi_{\text{avg}}] = n^*/D^*$ after many steps (see the proof of Lemma 8). Therefore, the estimator $\hat{\rho}(l)$ converges to $\rho^*(l) = \sum_{v_i \in V^*} 1_{\{l(i)=l\}}/n^*$ after many steps of our random walk. □

Then, we quantify the bias of the expected value $\rho^*(l)$ of the existing estimator.

Lemma 12. *If all the public nodes belong to the largest public cluster, we have*

$$\mathbb{E}_{pri}[\rho^*(l)] \approx \frac{\mathbb{E}_{pri}[\sum_{v_i \in V^*} 1_{\{l(i)=l\}}]}{\mathbb{E}_{pri}[n^*]} = \rho(l).$$

Proof. We define a random variable $X_{\text{label}}(i) = 1_{\{v_i \in V^* \wedge l(i)=l\}}$ for each node $v_i \in V$. Let $X_{\text{label}} = \sum_{v_i \in V^*} 1_{\{l(i)=l\}}$. It holds that $X_{\text{label}} = \sum_{v_i \in V} X_{\text{label}}(i)$. The expected value with respect to $\mathcal{L}_{\text{pri}}$ of X_{label} is derived as

$$\mathbb{E}_{\text{pri}}[X_{\text{label}}] = \sum_{v_i \in V} \Pr[v_i \in V^*] \mathbb{E}_{\text{pri}}[1_{\{l(i)=l\}}] = (1-p) \sum_{v_i \in V} 1_{\{l(i)=l\}}.$$

Note that the indicator function $1_{\{l(i)=l\}}$ is constant with respect to the set $\mathcal{L}_{\text{pri}}$. Consequently, it follows that Lemma 12 holds because of the above equation and Lemma 5. □

In contrast to the cases of the network size and average degree, Lemma 12 implies that the existing estimator $\hat{\rho}(l)$ has approximately no bias with respect to the set $\mathcal{L}_{\mathrm{pri}}$ if all public nodes belong to the largest public cluster. However, we empirically find that our estimator presented in the following further reduces the bias induced by private nodes.

Proposed estimator

We modify the weight for each sampled node to reduce the bias of the expected value induced by private nodes. We define the average of the modified weights Φ'_{label} and the proposed estimator $\hat{\rho}'(l)$ as follows:

$$\Phi'_{\mathrm{label}} = \frac{1}{r}\sum_{k=1}^{r}\frac{1_{\{l(x_k)=l\}}}{d_{x_k}}, \quad \hat{\rho}'(l) \triangleq \frac{\Phi'_{\mathrm{label}}}{\Phi'_{\mathrm{avg}}}.$$

The following lemma holds regarding the expected value of the proposed estimator.

Lemma 13. *The estimator $\hat{\rho}'(l)$ converges to*

$$\tilde{\rho}(l) = \frac{\sum_{v_i \in V^*} 1_{\{l(i)=l\}} d_i^*/d_i}{\sum_{v_i \in V^*} d_i^*/d_i}$$

after many steps of our random walk.

Proof. We calculate the expected value of Φ'_{label} with respect to $\boldsymbol{\pi}$ as follows:

$$\mathbb{E}_{\boldsymbol{\pi}}[\Phi'_{\mathrm{label}}] = \mathbb{E}_{\boldsymbol{\pi}}\left[\frac{1_{\{l(x_k)=l\}}}{d_{x_k}}\right] = \frac{1}{D^*}\sum_{v_i \in V^*}\frac{d_i^*}{d_i}1_{\{l(i)=l\}}.$$

Since Φ'_{avg} converges to the expected value $\mathbb{E}_{\boldsymbol{\pi}}[\Phi'_{\mathrm{avg}}] = (\sum_{v_i \in V^*} d_i^*/d_i)/D^*$ because of Theorem 2, $\hat{d}_{\mathrm{avg}}$ converges to $\tilde{d}_{\mathrm{avg}}$ after many steps of our random walk. □

We have the following proposition.

Proposition 3. *When the original graph G involves no private nodes, two estimators $\hat{\rho}(l)$ and $\hat{\rho}'(l)$ are equal, and two expected values, $\rho^*(l)$ and $\tilde{\rho}(l)$, are equal to the true quantity $\rho(l)$.*

Finally, we have the following theorem.

Theorem 5. *If all the public nodes belong to the largest public cluster, we have*

$$\mathbb{E}_{pri}[\tilde{\rho}(l)] \approx \frac{\mathbb{E}_{pri}[\sum_{v_i \in V^*} 1_{\{l(i)=l\}} d_i^*/d_i]}{\mathbb{E}_{pri}[\sum_{v_i \in V^*} d_i^*/d_i]} = \rho(l).$$

Proof. We define a random variable $\tilde{X}_{\mathrm{label}}(i) = 1_{\{v_i \in V^*\}} d_i^*/d_i$ for each node $v_i \in V$. Let $\tilde{X}_{\mathrm{label}} = \sum_{v_i \in V^*} 1_{\{l(i)=l\}} d_i^*/d_i$. It holds that $\tilde{X}_{\mathrm{label}} = \sum_{v_i \in V} \tilde{X}_{\mathrm{label}}(i)$. We obtain the expected value of $\tilde{X}_{\mathrm{label}}$ with respect to $\mathcal{L}_{\mathrm{pri}}$:

$$\mathbb{E}_{\mathrm{pri}}[\tilde{X}_{\mathrm{label}}] = \sum_{v_i \in V} \Pr[v_i \in V^*]\mathbb{E}_{\mathrm{pri}}\left[\frac{d_i^*}{d_i}1_{\{l(i)=l\}}\right] = (1-p)^2 \sum_{v_i \in V} 1_{\{l(i)=l\}}.$$

Theorem 5 holds because of the above equation and equation $\mathbb{E}_{\mathrm{pri}}[\sum_{v_i \in V^*} d_i^*/d_i] = (1-p)^2 n$ (see the proof of Theorem 4). □

2.5.4 Density of the private label of the node

The proposed estimator $\hat{\rho}'(l)$ is not applicable to the estimation of the density of the private label of the node (i.e., the percentage of private nodes). This is because the sample sequence does not contain private nodes. However, it is possible to intuitively estimate the probability p using the existing and proposed estimators of the network size and average degree each. Note that the probability p is almost equal to the density of the private label in a large-scale social network under Assumption 2. We denote by $\hat{p}_{\text{size}}$ the estimator of the probability p obtained by the existing and proposed estimators of the network size. We denote by $\hat{p}_{\text{avg}}$ the estimator of the probability p obtained by the existing and proposed estimators of the average degree. Based on Lemmas 5 and 9 and Theorems 3 and 4, we define estimators $\hat{p}_{\text{size}}$ and $\hat{p}_{\text{avg}}$ as

$$\hat{p}_{\text{size}} \triangleq 1 - \hat{n}/\hat{n}', \tag{2.2}$$

$$\hat{p}_{\text{avg}} \triangleq 1 - \hat{d}_{\text{avg}}/\hat{d}'_{\text{avg}}. \tag{2.3}$$

2.5.5 Estimation in the hidden privacy model

In the hidden privacy model, we calculate each estimator using the estimated public degree of each sampled node, $\hat{d}^*_{x_k}$. Even in this model, Lemmas 4, 6, 8, 10, 11, and 13 hold because of Lemma 2.

2.6 Experiments

We numerically evaluate the proposed estimators on social network datasets. We aim to answer the following questions:

1. Do the proposed estimators reduce the bias induced by private nodes of the existing estimators for the network size, average degree, and density of the node label? (Section 2.6.2)

2. Do the proposed estimators perform acceptably on social network datasets involving real private nodes? (Sections 2.6.3 and 2.6.4)

3. How does the proposed method for calculating the public degree of each sampled node affect the estimation accuracy and the number of queries in the hidden privacy model? (Section 2.6.5)

4. Is the number of additional queries generated by private nodes during seed selection small? (Section 2.6.6)

2.6.1 Datasets

Description of the datasets

We use five social network datasets, i.e., the YouTube, Pokec, Orkut, Facebook, and LiveJournal datasets. For these five datasets, we focus on undirected and connected graphs by the following pre-processing: (1) removing the directions of edges if the original graphs are directed and then (2) deleting the nodes that are not contained in the largest connected component of the original graph. The pre-processing does not affect any of the following experiments because the above pre-processing is performed before setting privacy labels of nodes and no processing is added to the graph after setting the privacy labels of the nodes. Table 2.1 shows the network size, average degree, and whether the dataset contains

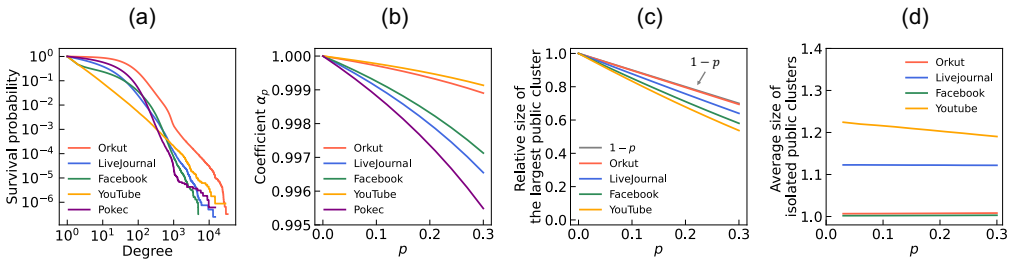


Figure 2.3: Four properties of the network datasets. (a) Cumulative degree distribution. (b) Coefficient α_p as a function of p. (c) Relative size of the largest public cluster as a function of p. (d) Average size of isolated public clusters as a function of p.

Table 2.1: Network datasets.

Network	n	d_{avg}	Privacy-label data	Reference
YouTube	1,134,890	5.27	Not contain	[119]
Pokec	1,632,803	27.32	Contain	[137]
Orkut	3,072,441	76.28	Not contain	[137]
Facebook	3,097,165	15.28	Not contain	[196]
LiveJournal	3,997,962	17.35	Not contain	[137]

the privacy-label data of nodes for the five network datasets. For the YouTube, Orkut, Facebook, and LiveJournal datasets, we independently and randomly set the privacy label of each node as private with a given probability p and otherwise public, according to Assumption 2. We vary the probability p from 0.0 to 0.30 in increments of 0.03 because there were actually tens of percentages of private nodes in social networks [41,45,66,216]. The Pokec dataset contains all the graph data of the network involving private nodes and contains real privacy labels of all the nodes [216]. Therefore, for the Pokec dataset, we apply the original privacy label to each node. Then, the Pokec network contains 552,525 real private nodes (i.e., approximately 33.8% of all the nodes).

We additionally use the dataset of the sample sequence of 1,016,275 public Facebook users obtained by Kurant et al.'s random walk in October 2010 [120]. The random walk is equivalent to our random walk in the ideal model because the Facebook graph as of October 2010 involves a certain percentage of private nodes and corresponds to the ideal model. This dataset contains the ID, the exact public degree, and the exact degree of each sampled public user, which allows us to compare the existing estimators and the proposed estimators.

Properties of the network datasets

Figure 2.3 shows four properties of each of the five network datasets. Figure 2.3(a) shows the cumulative degree distributions of the five datasets. We see that the networks have heavy-tailed degree distributions. Figure 2.3(b) shows the coefficient α_p, defined in Eq. (2.1), as a function of p for the five datasets. We find that α_p is almost equal to 1.0 for every value of p. This property is a result of the characteristic that the sum of squares of degrees is considerably larger than the sum of degrees for the networks. Figure 2.3(c) shows the relative size of the

largest public cluster, i.e., n^*/n, averaged over 1,000 independent and random sets of private nodes as a function of probability p for the four datasets. The gray solid line represents an expected upper limit, i.e., $1-p$. We observe that most public nodes belong to the largest public cluster for every probability p for the four datasets, which is qualitatively consistent with the finding in the previous study [16]. We also observe that all the public nodes belong to the largest public cluster of the Pokec network with real privacy labels of the nodes. Figure 2.3(d) shows the average size of isolated public clusters (i.e., public clusters other than the largest public cluster) averaged over 1,000 independent and random sets of private nodes as a function of probability p for the four datasets. The average size of isolated public clusters is considerably small for the four datasets, which is qualitatively consistent with the finding in the previous study [16]. We also observe that the Pokec network with real privacy labels of the nodes has no isolated public clusters.

2.6.2 Estimation accuracy of the proposed estimators when varying the percentage of private nodes

First, we compare the estimation accuracy of the existing and proposed estimators for the network size, the average degree, and the density of the node label using YouTube, Orkut, Facebook, and LiveJournal datasets. We set the density of node label of interest as a fraction of nodes with degree d or higher for each possible d (i.e., we set the indicator function used in $\hat{\rho}(l)$ and $\hat{\rho}'(l)$ as $1_{\{d_{x_k} \geq d\}}$ for node v_{x_k}). This is equivalent to estimating the cumulative degree distribution $\{P(d)\}_{d=1}^{d_{\max}}$, where $d_{\max}$ denotes the maximum degree of the node [85, 86, 132, 190]. For the network size and average degree, we evaluate the estimators using the normalized mean squared error (NRMSE) given by $\sqrt{\mathbb{E}[(\hat{x}/x-1)^2]}$, where x denotes the exact value and $\hat{x}$ denotes the estimator of x. The NRMSE has been used to evaluate both the bias and variance of a given estimator in the related literature [48,97,132, 234]. For the cumulative degree distribution, we calculate the NRMSE between the exact distribution $\{P(d)\}_{d=1}^{d_{\max}}$ and the estimated distribution $\{\hat{P}(d)\}_{d=1}^{d_{\max}}$. To this end, we use the normalized L^1 distance between $\{P(d)\}_{d=1}^{d_{\max}}$ and $\{\hat{P}(d)\}_{d=1}^{d_{\max}}$ given by $D_{P(d)} = \sum_{d=1}^{d_{\max}} |\hat{P}(d) - P(d)| / \sum_{d=1}^{d_{\max}} P(d)$. Then, the NRMSE of the estimator $\{\hat{P}(d)\}_{d=1}^{d_{\max}}$ is given by $\sqrt{\mathbb{E}[(D_{P(d)})^2]}$.

We perform the following simulations on the YouTube, Orkut, Facebook, and LiveJournal datasets. First, we independently and randomly set the privacy label of each node as private with a given probability p and otherwise public, according to Assumption 2. Second, we randomly select a seed on the largest public cluster. Third, we perform our random walk with a length r of 1% of the number of nodes. Finally, we calculate the existing and proposed estimators from the sampling list. For the given p, we estimate the NRMSE of each estimator over 1,000 independent runs.

Figure 2.4 shows the NRMSEs of the existing and proposed estimators for the network size as a function of probability p. The following observations apply to both access models. First, the NRMSEs of both estimators are exactly the same when $p = 0.0$, as shown in Proposition 1. Second, more importantly, the proposed estimator typically improves the NRMSE when $p > 0$. For example, the proposed estimator improves the NRMSE by approximately 67.7% (i.e., from 0.308 to 0.100) when $p = 0.3$ in Fig. 2.4(b). These observations qualitatively remain the same for the estimators of the average degree (see Fig. 2.5) and the cumulative degree distribution (see Fig. 2.6), except for Figs. 2.6(b) and 2.6(f).

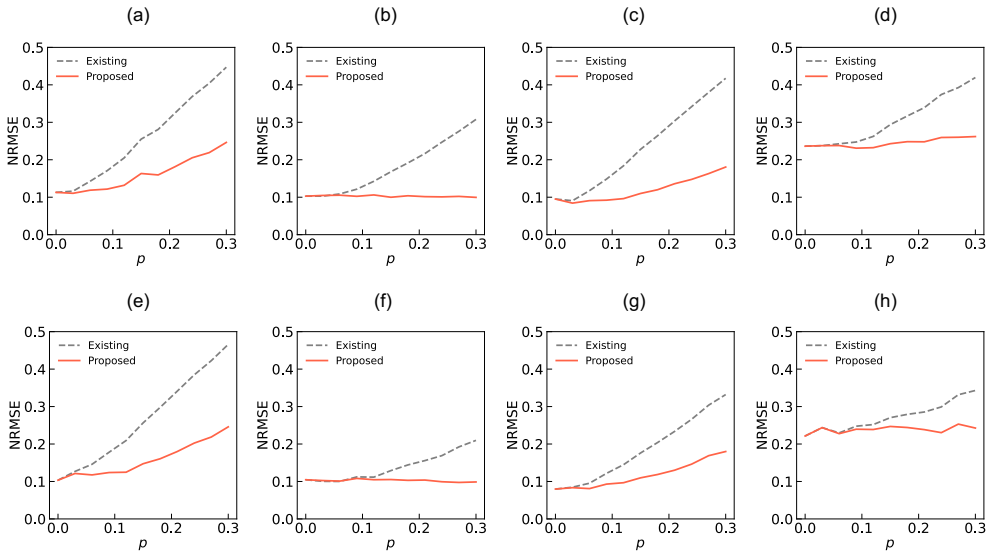


Figure 2.4: NRMSEs of the existing and proposed estimators for the network size as a function of probability p. Panels (a) and (e) show the results for the YouTube dataset; panels (b) and (f) show the results for the Orkut dataset; panels (c) and (g) show the results for the Facebook dataset; panels (d) and (h) show the results for the LiveJournal dataset. Panels (a)–(d) show the results in the ideal model, and panels (e)–(h) show the results in the hidden privacy model. We set the sample size as 1% of the number of nodes.

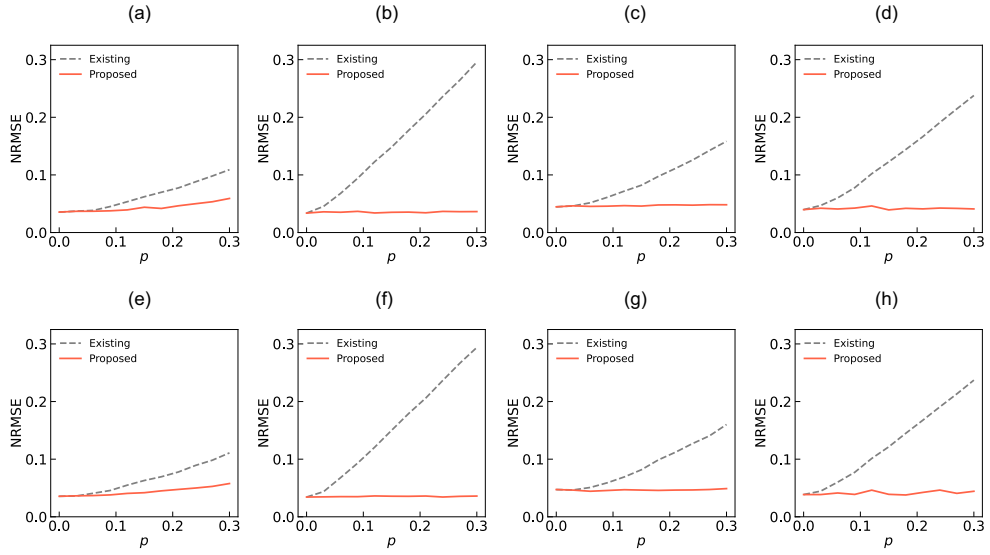


Figure 2.5: NRMSEs of the existing and proposed estimators for the average degree as a function of probability p. Panels (a) and (e) show the results for the YouTube dataset; panels (b) and (f) show the results for the Orkut dataset; panels (c) and (g) show the results for the Facebook dataset; panels (d) and (h) show the results for the LiveJournal dataset. Panels (a)–(d) show the results in the ideal model, and panels (e)–(h) show the results in the hidden privacy model. We set the sample size as 1% of the number of nodes.

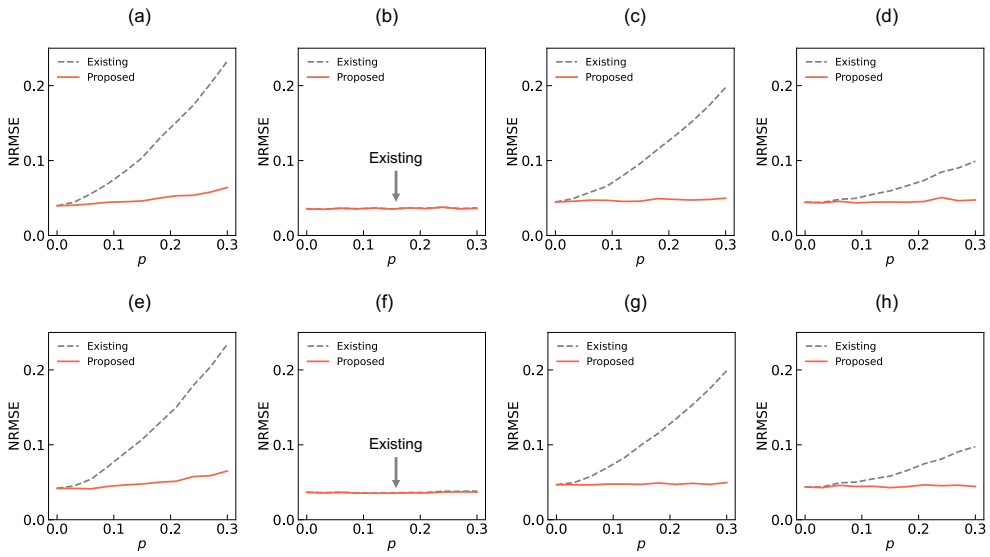


Figure 2.6: NRMSEs of the existing and proposed estimators for the cumulative degree distribution as a function of probability p. Panels (a) and (e) show the results for the YouTube dataset; panels (b) and (f) show the results for the Orkut dataset; panels (c) and (g) show the results for the Facebook dataset; panels (d) and (h) show the results for the LiveJournal dataset. Panels (a)–(d) show the results in the ideal model, and panels (e)–(h) show the results in the hidden privacy model. We set the sample size as 1% of the number of nodes. We indicate the curves by an arrow and label when two curves heavily overlap each other.

The improvement in the NRMSE results from the reduction of the bias induced by private nodes. To confirm this, we observe the NRMSEs of expected values of the existing and proposed estimators for each property (see Lemmas 4, 6, 8, 10, 11, and 13 for the expected value of each estimator). For the given p, we calculate the NRMSEs of the expected values of the existing and proposed estimators over 1,000 random sets of privacy labels of nodes. Note that the expected value of each estimator does not depend on the access model.

Figure 2.7(a)–(d) shows the NRMSEs of expected values of the existing and proposed estimators for the network size as a function of probability p. The following observations apply to the four datasets. First, the NRMSEs of both estimators are equal to zero when $p = 0$, as shown in Proposition 1. Second, the proposed estimator improves the NRMSE of the expected value when $p > 0$. These observations are qualitatively the same for the estimators of the average degree (see Fig. 2.7(e)–(h)) and the estimators of the cumulative degree distribution (see Fig. 2.7(i)–(l)). The proposed estimator for the cumulative degree distribution slightly reduces the bias induced by private nodes on the Orkut dataset (see Fig. 2.7(j)), which is qualitatively consistent with the little improvement in the NRMSE of the proposed estimator in Figs. 2.6(b) and (f).

The existing and proposed estimators have the bias induced by public nodes that do not belong to the largest public cluster. This is because our random walk samples only the nodes that belong to the largest public cluster. On the Orkut dataset, where almost all of the public nodes belong to the largest public cluster (see Fig. 2.3(c)), the proposed estimators for the network size, average degree, and cumulative degree distribution have approximately no bias for any probability p (see Figs. 2.7(b), 2.7(f), and 2.7(j)). These results support our theoretical results

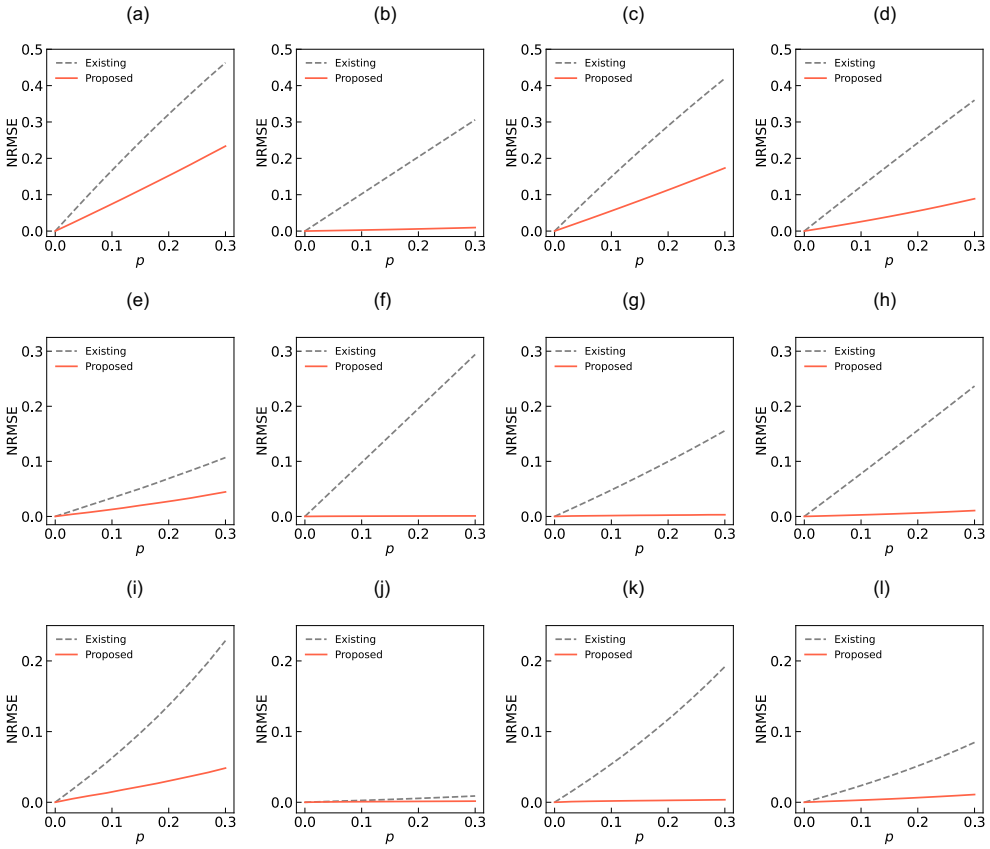


Figure 2.7: NRMSEs of the expected values of the existing and proposed estimators as a function of probability p. Panels (a), (e), and (i) show the results for the YouTube dataset; panels (b), (f), and (j) show the results for the Orkut dataset; panels (c), (g), and (k) show the results for the Facebook dataset; panels (d), (h), and (l) show the results for the LiveJournal dataset. Panels (a)–(d) show the results for the network size; panels (e)–(h) show the results for the average degree; panels (i)–(l) show the results for the cumulative degree distribution.

of Theorems 3, 4, and 5. The biases of the existing estimators for the network size and average degree approximately increase linearly with probability p, which supports Lemmas 5 and 9 (see Fig. 2.7(b) and Fig. 2.7(f)). The existing estimator for the cumulative degree distribution has approximately no bias for any p, which supports Lemma 12 (see Fig. 2.7(j)). On the other hand, on the YouTube dataset, where there are the most public nodes that do not belong to the largest public cluster among the four datasets (see Fig. 2.3(c)), the biases of the existing and proposed estimators relatively increase as the probability p increases (see Figs. 2.7(a), 2.7(e), and 2.7(i)). Nevertheless, the proposed estimators still have smaller biases induced by private nodes than the existing estimators when $p > 0$.

2.6.3 Estimation on the Pokec dataset

We evaluate the proposed estimators using the Pokec network dataset involving real private nodes [137, 216]. We perform the following simulations on the Pokec dataset. First, we apply the original privacy label contained in the dataset to each node. Second, we compute the largest public cluster of the Pokec network. Third, we randomly select a seed on the largest public cluster. Fourth, we perform

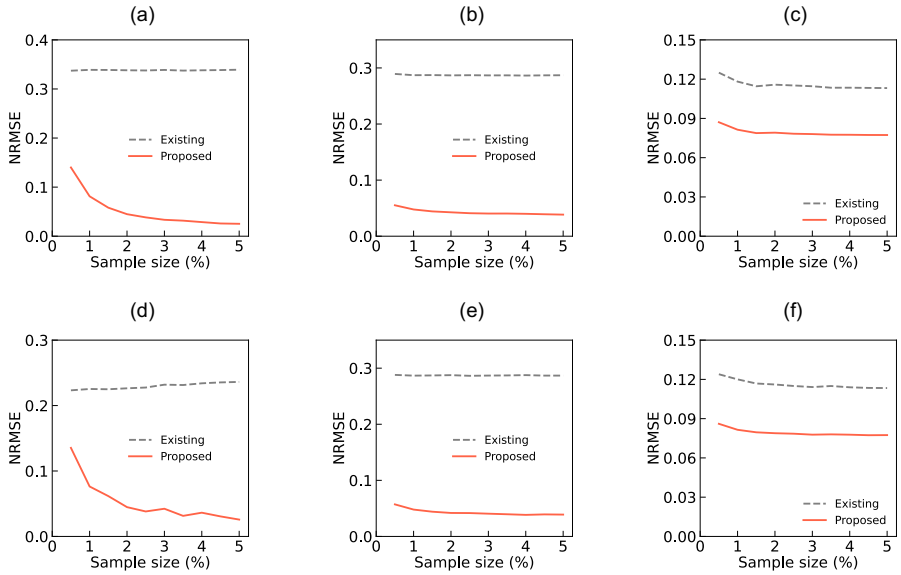


Figure 2.8: NRMSEs of the existing and proposed estimators as a function of the sample size on the Pokec dataset. Panels (a) and (d) show the results for the network size; panels (b) and (e) show the results for the average degree; panels (c) and (f) show the results for the cumulative degree distribution. Panels (a)–(c) show the results in the ideal model, and panels (d)–(f) show the results in the hidden privacy model.

our random walk with length r. Finally, we calculate the existing and proposed estimators from the sampling list. We vary the length r from 0.5% of the number of nodes to 5% of the number of nodes in increments of 0.5% of the number of nodes. For the given r, we estimate the NRMSE of each estimator over 1,000 independent runs.

Figure 2.8 shows the NRMSE of the existing and proposed estimators for the three properties as a function of the sample size. For each property, the proposed estimator improves the NRMSE for any sample size in both access models. For example, the proposed estimator for the network size improves the NRMSE by approximately 92.6% (i.e., from 0.339 to 0.025) in the case of 5% sample size in the ideal model (see Fig. 2.8(a)). The improvement in the NRMSE results from the reduction of the bias of the proposed estimators. Table 2.2 shows the errors (i.e., the relative error or the L^1 distance) of the expected values of the existing and proposed estimators for each property. The proposed estimators improve the corresponding error by 97.3% for the network size, 87.5% for the average degree, and 32.1% for the cumulative degree distribution. Furthermore, for each property, the proposed estimator has the expected value that is almost equal to the true quantity of the whole Pokec network involving real private nodes. Although Assumption 2 does not hold for the Pokec network, we found that the proposed estimators yield reasonable results as claimed in Theorems 3, 4, and 5.

2.6.4 Estimation on the Facebook sample dataset

We use the sample sequence of 1,016,275 public nodes obtained by Kurant et al.'s random walk on the Facebook graph in October 2010 [120]. The dataset contains the ID, the exact public degree, and the exact degree of each sampled public node. Therefore, we compare the existing and proposed estimators for the network size,

Table 2.2: Expected errors of the existing and proposed estimators for the three properties on the Pokec dataset. For the network size and average degree, the error shows the relative error. For the cumulative degree distribution, the error shows the L^1 distance.

Network property	Existing	Proposed
Network size	0.338	0.009
Average degree	0.287	0.036
Cumulative degree distribution	0.112	0.076

Table 2.3: Estimates of the network size and average degree obtained from the Facebook sample dataset.

Network property	Existing	Proposed
Network size	480,298,540	656,874,081
Average degree	102.07	137.03

average degree, and cumulative degree distribution of the Facebook graph as of October 2010.

Table 2.3 shows the existing and proposed estimators for the network size and average degree. It is difficult to calculate the error of each estimator because the true quantities of the Facebook graph as of October 2010 are unknown. However, we consider that the estimates are reasonable considering two findings in almost the same period. First, Facebook reported that there were 500 million active users as of July 2010 [72]. This means that there were at least 500 million users, including inactive users, at that time. Notably, the estimates of the network size shown in Table 2.3 count both active and inactive users. Our estimate, i.e., 657 million, is greater than 500 million, and we speculate that the difference (approximately 157 million) mainly comprises inactive users. Second, Catanese et al. obtained the unbiased estimate of the proportion of private nodes as 0.266 from a uniform sample of Facebook users in August 2010 [45][2]. According to Table 2.3, we obtain the estimates of the proportion of private nodes, i.e., $\hat{p}_{\text{size}}$ defined in Eq. (2.2) and $\hat{p}_{\text{avg}}$ defined in Eq. (2.3), as 0.269 and 0.255, respectively. These two estimates are considerably close to the ground truth value of 0.266. Figure 2.9 shows the existing and proposed estimators for the cumulative degree distribution. We observe that two estimates heavily overlap each other. This result is qualitatively consistent with our theoretical results of Lemma 11 and Theorem 5.

2.6.5 Effectiveness of the proposed method for calculating the public degree of each sampled node

We evaluate the proposed method for calculating the public degree of each sampled node in the hidden privacy model. The proposed estimator for the network size requires the public degree of each sampled node for re-weighting (see Section 2.5.1). Therefore, we compare the NRMSE and the proportion of queried nodes

[2]When Catanese et al. performed a uniform sampling of users on Facebook during August 2010, the user id was 32 bit. As mentioned in Refs. [85, 86], shortly after that, Facebook's user id went to 64 bit, and uniform sampling in the 64-bit space is typically infeasible.

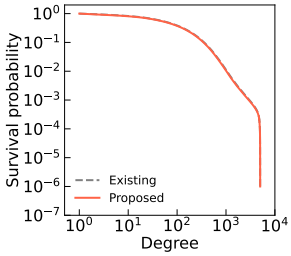


Figure 2.9: Estimates of the cumulative degree distribution obtained from the Facebook sample dataset. Two curves heavily overlap each other.

for the proposed size estimator using the proposed method and exact method, respectively. In the exact method, one queries all the neighbors of each sampled node to calculate the exact public degree in re-weighting. We perform the simulations on the YouTube, Orkut, Facebook, and LiveJournal datasets using the same procedure followed in Section 2.6.2.

Figure 2.10 shows the results for the four datasets. The following observations apply to all the four datasets. First, the proposed method achieves almost the same NRMSE as the exact method (see Fig. 2.10(a)–(d)). Second, although the exact method queries tens of percent of nodes which are much greater than the 1% sample size, the proposed method queries approximately 1% nodes (see Fig. 2.10(e)–(h)). These results support our theoretical result of Lemma 3. Therefore, the proposed method reduces the proportion of queried nodes by tens of percent while maintaining almost the estimation accuracy compared with the case of using the exact method.

2.6.6 Selection of a seed on the largest public cluster

Thus far, we have assumed that we have access to some arbitrary node in the largest public cluster of the original graph to begin our random walk (see Assumption 3). In practical scenarios, we require additional queries in the seed-selection phase by restarting a random walk from another seed in the following two cases. The first case is when a given seed is a private node. This is the case, for example, when one selects nodes v_3 or v_7 as a seed in the graph shown in Fig. 2.1. The second case is when a given public seed is on an isolated public cluster (i.e., a public cluster other than the largest public cluster). This is the case, for example, when one selects nodes v_6, v_8, or v_9 as a seed in the graph shown in Fig. 2.1.

We consider that the number of queries generated in each case is sufficiently small. We generate a small number of queries in the first case because (i) the proportion of private nodes is generally smaller than that of public nodes in real social networks (e.g., 27% on Facebook [45] and 34% on Pokec [216]) and (ii) one query is enough to check the privacy label of a node. In the second case, we generate a small number of queries under Assumption 2 owing to the following two natures of real-world networks having heavy-tailed degree distributions [16]. First, most public nodes belong to the largest public cluster. In our simulations, even if $p = 0.3$, 99.1% on Orkut, 91.4% on LiveJournal, 82.8% on Facebook, and 76.7% on YouTube of public nodes belong to the largest public cluster (see Fig. 2.3(c)). Therefore, a selected public seed belongs to the largest public cluster with high probability. Second, the average size of isolated public clusters is considerably smaller than the size of the largest public cluster. In our simulations, the average

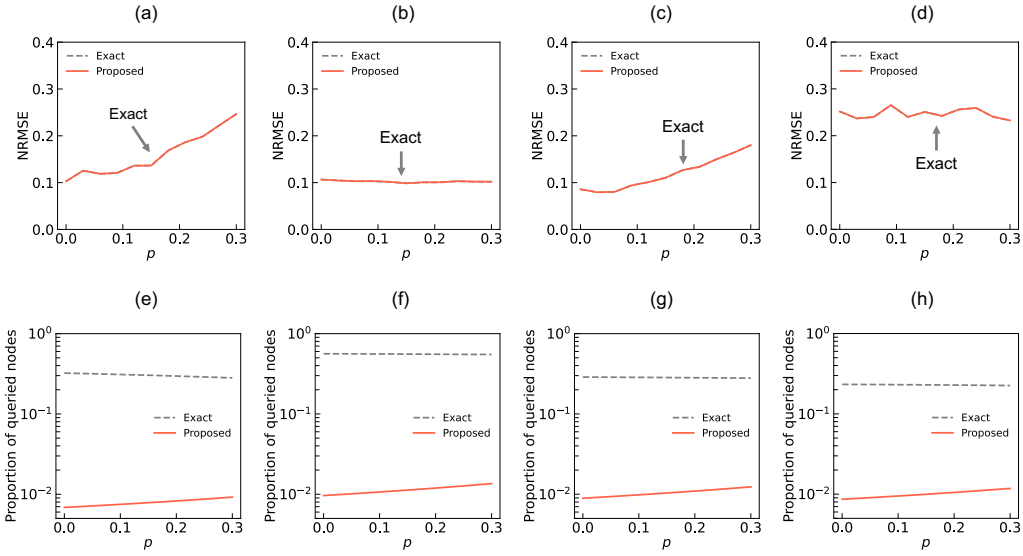


Figure 2.10: Effects of the proposed method for calculating the public degree of each sampled node in the hidden privacy model. Panels (a) and (d) show the results for the YouTube dataset; panels (b) and (e) show the results for the Orkut dataset; panels (c) and (f) show the results for the Facebook dataset; panels (d) and (h) show the results for the LiveJournal dataset. Panels (a)–(d) show the NRMSEs of the existing and proposed estimators for the network size as a function of probability p. Panels (e)–(h) show the proportion of queried nodes using the exact and proposed methods for calculating the public degree of each sampled node as a function of probability p. We set the sample size as 1% of the number of nodes. We indicate the curves by an arrow and label when two curves heavily overlap each other.

size is only approximately one for every probability p on the four datasets (see Fig. 2.3(d)). Therefore, even if a selected seed belongs to an isolated public cluster, we will generate a small number of queries there. Finally, we consider that two real-world datasets support Assumption 3. There are no isolated public clusters on the Pokec network; therefore, the second case does not occur on this network. The Facebook sample dataset yields an estimate of 657 million users and contains one million unique public users; therefore, we consider that the sample was obtained from the largest public cluster of the Facebook graph as of October 2010.

2.7 Conclusion

In this chapter, we proposed a framework for estimating properties based on a random walk on social networks involving private nodes. Social networks typically involve a certain percentage of private nodes that do not publish their neighbors' data when they are queried. However, previous studies have ignored the effects of private nodes on random walk-based estimators because private nodes inhibit the performing of a simple random walk on the network. We extended a simple random walk and the existing estimators for the three properties to the case of social networks involving private nodes based on the three assumptions with respect to private nodes. Although Assumption 2 oversimplifies the distribution of private nodes in social networks, the proposed estimators based on the assumption

realized reasonable estimates on the two social network datasets involving real private nodes. We expect that this work will lead to the accurate estimation of the properties of social networks and finally to the understanding of social characteristics such as human connections and behaviors.

Chapter 3

Social Graph Restoration via Random Walk Sampling

3.1 Introduction

Analysts are often interested in various characteristics of social networks, such as local structural properties (e.g., the degree distribution and clustering coefficient), global structural properties (e.g., the distributions of shortest-path lengths and betweenness centrality), and visual graph representations [36]. Throughout the work presented in Chapter 2, we raise the following question: will only such improvement of the re-weighted random walk realizes methods to exhaustively analyze properties of online social networks? The answer may be no because the re-weighted random walk is specialized in estimating local structural properties in principle.

In this chapter, we study the social graph restoration problem: given a small sample of a social graph obtained by crawling, we aim to generate a graph whose structural properties are as close as possible to the corresponding properties of the original graph. To address this problem, we propose a method that generates a graph that preserves the estimates of local structural properties and the structure of the subgraph sampled by a random walk. Our experimental results show that the proposed method outperforms existing methods in terms of the average accuracy of 12 structural properties and the visual representation of generated graphs. The source code for our method is available at `https://github.com/kazuibasou/social-graph-restoration`.

3.2 Related Work

In the past decade, a number of algorithms based on the re-weighted random walk have been developed for accurately estimating structural properties using a small number of queries. Examples of such structural properties include the network size [97,112], average degree [62,85,86], degree distribution [85,86], joint degree distribution [84], clustering coefficients [32,97,190], motifs and graphlets [48,95,234], and node centrality [160,161]. Most of these existing algorithms focus on estimating local structural properties.

We regard subgraph sampling as a baseline method for the social graph restoration problem. In subgraph sampling, one constructs the subgraph induced from a set of edges obtained using a crawling method [13, 134, 136]. In early studies, the subgraph was implicitly assumed to be a representative sample of the original graph [15, 45, 122, 154, 239]. However, Gjoka et al. demonstrated that crawling methods typically introduce a significant sampling bias toward high-degree nodes [85, 86]. In this work, we compare the proposed method with sub-

graph sampling using each of the well-used crawling methods (i.e., breadth-first search [45, 121, 154, 197, 239], snowball sampling [15, 89, 106, 134, 197], forest fire sampling [13, 65, 136, 197], and random walk). We confirm that subgraph sampling using a small sample typically introduces bias in structural properties on average and misses the surrounding structure that consists of low-degree nodes in the graph visualization.

Gjoka et al. proposed a method for generating a graph that preserves the estimates of the joint degree distribution and degree-dependent clustering coefficient obtained by re-weighted random walk [84]. The authors showed that the generated graph accurately reproduces not only local structural properties but also global structural properties that are not intended to be preserved. In this work, we propose a method for generating a graph that preserves both the estimates of local structural properties (i.e., the number of nodes, average degree, degree distribution, joint degree distribution, and degree-dependent clustering coefficient) and the structure of the subgraph sampled by a random walk. Specifically, we add nodes and edges to the subgraph sampled by a random walk to ensure that the final graph preserves the estimates of local structural properties. Our underlying idea is to optimally use the raw structural information of the subgraph in the generation process. Our experimental results show that the proposed method more accurately reproduces an average of 12 structural properties and the visual representation of the original graph and has a generation time that is several times faster than that of Gjoka et al.'s method.

Several studies have developed random walk algorithms to improve the accuracy of estimators or the efficiency of the number of queries [132, 140, 141, 162, 168,190,245,255]. Ribeiro and Towsley proposed multidimensional random walks, which improve the estimation accuracy over a simple random walk (i.e., one repeatedly moves to a neighbor chosen uniformly and randomly on the graph) in the presence of disconnected connected components [190]. Lee et al. proposed the non-backtracking random walk algorithm, which improves the query efficiency while preserving the Markov property of the sample sequence [132]. Nakajima and Shudo recently proposed a random walk algorithm to reduce the bias caused by private nodes whose neighbors' data are not retrievable in social networks [162]. In this work, we propose a method for restoring the original social graph via a simple random walk. However, while it is not trivial, it is possible to combine the above improved random walks with the proposed method.

Graph-generative models have been developed to reproduce the structural properties of a given graph [35,46,88,135,148,180,193,208,227,248]. In this work, we extend a family of generative models called the dK-series [84, 148, 180] to the generation of a graph that preserves the estimates of local structural properties and the structure of the subgraph sampled by a random walk. It is not trivial to extend any generative model including the dK-series, which assumes that all graph data are available, to the social graph restoration problem because of the following three reasons. First, we need to estimate the input parameters of the model from a sample of the original graph. Second, we need to construct the input parameters from their estimates so that those parameters meet all conditions required to realize the desired graph. Third, although one adds nodes and edges in an empty graph in most generative models, we add nodes and edges to the sampled subgraph to restore the original graph.

Social graph restoration is related to matrix completion [44], in which one complements entries in a given matrix, and is also related to link prediction [142], network completion [114], and network inference [96, 103, 176], in which one complements nodes or edges in a given graph. However, the subgraph sampled by a

random walk omits nodes biased toward low degrees and their edges, which is a special case of the assumption of these problems regarding missing nodes or edges. The proposed method is specialized in the case of complementing the nodes and edges in the subgraph sampled by a random walk.

3.3 Preliminaries

3.3.1 Problem definition

We represent a connected and undirected social graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is a set of nodes (users) and $\mathcal{E}$ is a set of edges (friendships). We allow multiple edges and loops. Let n denote the number of nodes and m denote the number of edges. We denote the adjacency matrix for $\mathcal{G}$ by $\boldsymbol{A}$. We assume that A_{ij} is the number of edges between v_i and v_j $(i \neq j)$. We assume that A_{ii} is equal to twice the number of loops of v_i by convention [177]. Let $\mathcal{N}(i)$ denote a set of edges connected to v_i. Let $d_i = |\mathcal{N}(i)|$ be the degree of v_i and $k_{\max}$ be the maximum degree of the node. Let $1_{\{cond\}}$ denote a function that returns 1 if a condition *cond* holds and 0 otherwise.

We assume the standard model for accessing graph $\mathcal{G}$ as in Refs. [15, 85, 86, 122, 154, 239]: (i) if one queries node v_i, the set $\mathcal{N}(i)$ is available; (ii) completely or randomly accessing $\mathcal{G}$ is not feasible; and (iii) the graph $\mathcal{G}$ is static. Crawling methods (e.g., breadth-first search and random walk) are effective for sampling the nodes and edges of a graph in this access model.

We study the following problem: given a sampling list of the indices and connected edges of a small fraction of nodes queried using a crawling method, we generate a graph whose structural properties are as close as possible to the corresponding structural properties of the original graph $\mathcal{G}$.

3.3.2 Random walk

We obtain a sequence of sampled nodes via a simple random walk as follows. We select a seed node v_{x_1}, where x_i denotes the index of the i-th sampled node. For the i-th sampled node $(i = 1, \ldots, r-1)$, we select an edge uniformly at random from the set $\mathcal{N}(x_i)$ and then pass through the edge. Finally, we obtain a list of the indices and connected edges of r sampled nodes, as denoted by $\mathcal{L} = ((x_i, \mathcal{N}(x_i)))_{i=1}^{r}$.

3.3.3 dK-series

We use the concept of a family of graph-generative models called the dK-series [84, 148, 180]. The dK-series defines a series of random graphs called dK-graphs that preserve all the joint degree distributions of the nodes in the subgraphs of size d or less in a given graph. $0K$-graphs preserve the number of nodes n and the average degree $\bar{k}$ of a given graph, where we define

$$\bar{k} = \frac{2m}{n}. \tag{3.1}$$

$1K$-graphs preserve n, $\bar{k}$, and the degree distribution $\{P(k)\}_k$ of a given graph. We define

$$P(k) = \frac{n(k)}{n} \tag{3.2}$$

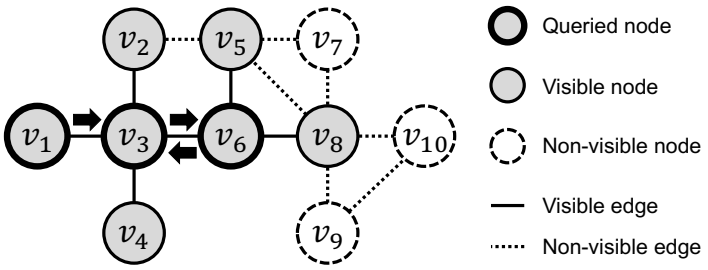


Figure 3.1: An example of a random walk on a graph.

for $k = 1, \ldots, k_{\max}$, where $n(k)$ is the number of nodes with degree k. Preserving n, $\bar{k}$, and $\{P(k)\}_k$ is identical to preserving $\{n(k)\}_k$. We refer to $\{n(k)\}_k$ as a *degree vector*, as in Ref. [211]. $2K$-graphs preserve n, $\bar{k}$, $\{P(k)\}_k$, and the joint degree distribution $\{P(k,k')\}_{k,k'}$ of a given graph. We define

$$P(k,k') = \frac{\mu(k,k')m(k,k')}{2m} \tag{3.3}$$

for $k = 1, \ldots, k_{\max}$, $k' = 1, \ldots, k_{\max}$, where $m(k,k')$ is the number of edges between nodes with degree k and nodes with degree k'. We define $\mu(k,k') = 1$ if $k \neq k'$ and $\mu(k,k) = 2$ otherwise such that $P(k,k')$ is normalized; i.e., $\sum_{k=1}^{k_{\max}} \sum_{k'=1}^{k_{\max}} P(k,k') = 1$. Preserving n, $\bar{k}$, $\{P(k)\}_k$, and $\{P(k,k')\}_{k,k'}$ is identical to preserving $\{m(k,k')\}_{k,k'}$. We refer to $\{m(k,k')\}_{k,k'}$ as a *joint degree matrix*, as in Ref. [211]. $2.5K$-graphs preserve n, $\bar{k}$, $\{P(k)\}_k$, $\{P(k,k')\}_{k,k'}$, and the degree-dependent clustering coefficient $\{\bar{c}(k)\}_k$ of a given graph. For $k = 1, \ldots, k_{\max}$, we define

$$\bar{c}(k) = \frac{1}{n(k)} \sum_{i=1,\ d_i=k}^{n} \frac{2t_i}{k(k-1)},$$

where $t_i = \sum_{j=1,\ j\neq i}^{n-1} \sum_{l=j+1,\ l\neq i}^{n} A_{ij}A_{il}A_{jl}$ is the number of triangles to which v_i belongs and $\bar{c}(1) = 0$.

dK-graphs more accurately reproduce the structural properties of a given graph as the value of d increases [148, 180]. Gjoka et al. demonstrated that $2.5K$-graphs successfully reproduce not only local structural properties but also global structural properties (e.g., shortest path properties) that are not intended to be preserved [84].

3.3.4 Subgraph sampling

In our method, we first construct the subgraph induced from a set of edges obtained using a random walk. We find a subset of edges in $\mathcal{G}$ obtained through a random walk as

$$\mathcal{E}' = \bigcup_{v_i \in \mathcal{V}'_{\text{qry}}} \mathcal{N}(i),$$

where $\mathcal{V}'_{\text{qry}}$ denotes a set of queried nodes. In other words, $\mathcal{E}'$ is a union set of edges connected to each of queried nodes. Then, we construct the subgraph, $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, which is induced from the subset $\mathcal{E}'$. The subset $\mathcal{V}'$ consists of two disjoint sets, i.e., $\mathcal{V}'_{\text{qry}}$ and $\mathcal{V}'_{\text{vis}}$, where $\mathcal{V}'_{\text{vis}}$ denotes a set of nodes visible as neighbors of the queried nodes.

Figure 3.1 shows an example in which we traverse nodes in the order v_1, v_3, v_6, and v_3 via a random walk on a graph. In this example, we obtain $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, where $\mathcal{V}'_{\mathrm{qry}} = \{v_1, v_3, v_6\}$, $\mathcal{V}'_{\mathrm{vis}} = \{v_2, v_4, v_5, v_8\}$, $\mathcal{V}' = \{v_1, v_2, v_3, v_4, v_5, v_6, v_8\}$ and $\mathcal{E}' = \{(v_1, v_3), (v_2, v_3), (v_3, v_4), (v_3, v_6), (v_5, v_6), (v_6, v_8)\}$.

3.3.5 Unbiased estimators of local structural properties

Then, we estimate the number of nodes, average degree, degree distribution, joint degree distribution, and degree-dependent clustering coefficient of the original graph from the sampling list $\mathcal{L}$. For this purpose, we use existing estimators based on re-weighted random walk as follows.

Let $\mathcal{I} = \{(i, j) \mid M \leq |i - j| \wedge 1 \leq i,\ j \leq r\}$ denote a set of integer pairs that are between 1 and r and are located at least a threshold M away. An unbiased estimator of the number of nodes [97, 112] is given by

$$\hat{n} \triangleq \frac{\sum_{(i,j)\in\mathcal{I}} d_{x_i}/d_{x_j}}{\sum_{(i,j)\in\mathcal{I}} 1_{\{x_i = x_j\}}}.$$

We set $M = 0.025r$, as in the previous study [97].

An unbiased estimator of the average degree [62, 86] is given by $\hat{\bar{k}} \triangleq 1/\bar{\Phi}$, where we define

$$\bar{\Phi} = \frac{1}{r}\sum_{i=1}^{r} 1/d_{x_i}.$$

An unbiased estimator of the degree distribution [85, 86, 190] is given by $\hat{P}(k) \triangleq \Phi(k)/\bar{\Phi}$, where we define

$$\Phi(k) = \frac{1}{kr}\sum_{i=1}^{r} 1_{\{d_{x_i}=k\}}.$$

An unbiased estimator of the joint degree distribution is given by combining the following two methods [84]: induced edges (IE) and traversed edges (TE). The unbiased estimator of the joint degree distribution using IE is defined as $\hat{P}_{\mathrm{IE}}(k, k') \triangleq \hat{n}\hat{\bar{k}}\Phi(k, k')$, where we define

$$\Phi(k, k') = \frac{1}{kk'|\mathcal{I}|}\sum_{(i,j)\in\mathcal{I}} 1_{\{d_{x_i}=k \wedge d_{x_j}=k'\}} A_{x_i x_j}.$$

Then, the unbiased estimator of the joint degree distribution using TE is defined as

$$\begin{aligned}&\hat{P}_{\mathrm{TE}}(k, k')\\ \triangleq&\frac{1}{2(r-1)}\sum_{i=1}^{r-1} (1_{\{d_{x_i}=k \wedge d_{x_{i+1}}=k'\}} + 1_{\{d_{x_i}=k' \wedge d_{x_{i+1}}=k\}}).\end{aligned}$$

Finally, the hybrid unbiased estimator $\hat{P}(k, k')$ is defined with $\hat{\bar{k}}$ as a threshold:

$$\hat{P}(k, k') \triangleq \begin{cases} \hat{P}_{\mathrm{IE}}(k, k') & (\text{if } k + k' \geq 2\hat{\bar{k}}),\\ \hat{P}_{\mathrm{TE}}(k, k') & (\text{if } k + k' < 2\hat{\bar{k}}).\end{cases}$$

The original paper [84] did not prove that $\hat{P}(k, k')$ is an unbiased estimator of $P(k, k')$. Therefore, we prove that in Section 3.8.

An unbiased estimator of the degree-dependent clustering coefficient [97] is given by $\hat{\bar{c}}(k) \triangleq \Phi_{\bar{c}}(k)/\Phi(k)$, where we define

$$\Phi_{\bar{c}}(k) = \frac{1}{(k-1)(r-2)} \sum_{i=2}^{r-1} 1_{\{d_{x_i}=k\}} A_{x_{i-1}x_{i+1}}.$$

3.4 Proposed Method

3.4.1 Overview

In this section, we propose a method for restoring the original graph $\mathcal{G}$ based on the subgraph $\mathcal{G}'$ and the estimates of five local structural properties (i.e., the number of nodes $\hat{n}$, average degree $\hat{\bar{k}}$, degree distribution $\{\hat{P}(k)\}_k$, joint degree distribution $\{\hat{P}(k,k')\}_{k,k'}$, and degree-dependent clustering coefficient $\{\hat{\bar{c}}(k)\}_k$). Our idea is to add nodes and edges to the subgraph to generate a graph that preserves these estimates of local structural properties. We intend to reproduce the global structural properties of the original graph by preserving local structural properties, as in the underlying idea of the dK-series [84, 148, 180]. Furthermore, we intend to reproduce the structural properties and the visual representation of the original graph more accurately by preserving the structure of the subgraph.

The proposed method consists of four phases (see also Fig. 3.2). We denote the graph to be generated by the proposed method as $\tilde{\mathcal{G}}$ throughout this section. In the first phase, we construct the target degree vector, as denoted by $\{n^*(k)\}_k$ (Section 3.4.2). This vector determines the number of nodes with degree k in $\tilde{\mathcal{G}}$. We construct the target degree vector based on the subgraph $\mathcal{G}'$ and the estimates $\hat{n}$ and $\{\hat{P}(k)\}_k$. In the second phase, we construct the target joint degree matrix, as denoted by $\{m^*(k,k')\}_{k,k'}$ (Section 3.4.3). This matrix determines the number of edges between nodes with degree k and nodes with degree k' in $\tilde{\mathcal{G}}$. We construct the target joint degree matrix based on the subgraph $\tilde{\mathcal{G}}$, the estimates $\hat{n}$, $\hat{\bar{k}}$, and $\{\hat{P}(k,k')\}_{k,k'}$, and the target degree vector $\{n^*(k)\}_k$. In the third phase, we add nodes and edges to the subgraph to ensure that the generated graph $\tilde{\mathcal{G}}$ preserves $\{n^*(k)\}_k$ and $\{m^*(k,k')\}_{k,k'}$ (Section 3.4.4). In the fourth phase, we repeatedly rewire edges in the generated graph $\tilde{\mathcal{G}}$ so that $\tilde{\mathcal{G}}$ also preserves the estimate $\{\hat{\bar{c}}(k)\}_k$ (Section 3.4.5). In the following sections, we describe each phase of the proposed method in detail.

3.4.2 Constructing a target degree vector

In the first phase, we construct a target degree vector based on the subgraph $\mathcal{G}'$ and the estimates of the number of nodes $\hat{n}$ and the degree distribution $\hat{P}(k)$. The target degree vector, as denoted by $\{n^*(k)\}_k$, determines the number of nodes with degree k in the graph to be generated $\tilde{\mathcal{G}}$.

We denote by $k^*_{\max}$ the target maximum degree of the graph to be generated, $\tilde{\mathcal{G}}$. In general, a target degree vector, $\{n^*(k)\}_k$, needs to satisfy the following two conditions to realize a graph that preserves it [178]:

(DV-1) $n^*(k)$ is a nonnegative integer for each $k = 1, \ldots, k^*_{\max}$.

(DV-2) $\sum_{k=1}^{k^*_{\max}} k n^*(k)$ is an even number.

However, the immediate estimate of the number of nodes with degree k obtained by $\hat{n}(k) = \hat{n}\hat{P}(k)$ (see Eq. (3.2)) typically does not satisfy these realization conditions. For example, $\hat{n}(k)$ is not typically an integer for each degree k.

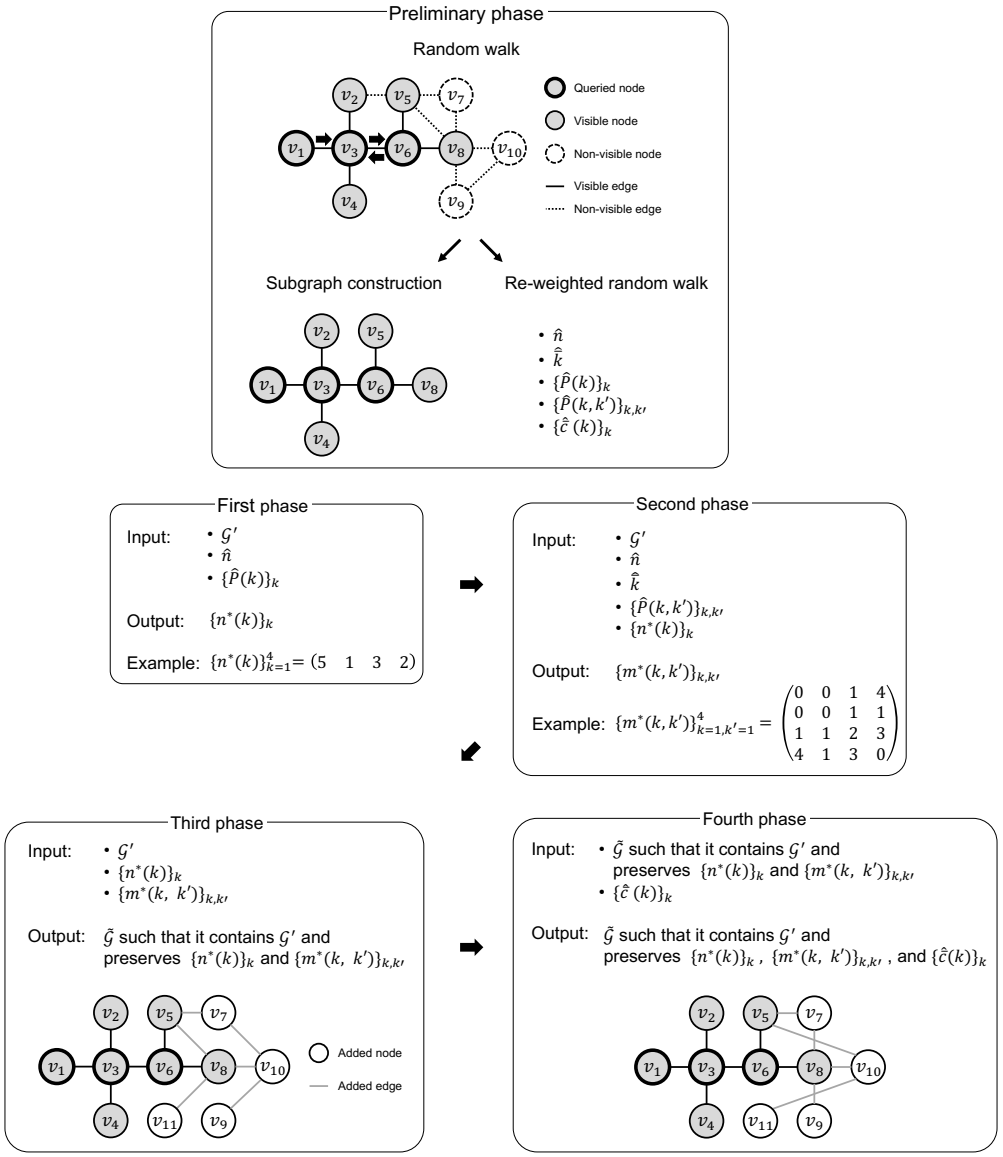


Figure 3.2: Workflow of the proposed method. $\mathcal{G}'$ represents the subgraph obtained by a random walk; $\hat{n}$ represents the estimate of the number of nodes; $\hat{\bar{k}}$ represents the estimate of the average degree; $\{\hat{P}(k)\}_k$ represents the estimate of the degree distribution; $\{n^*(k)\}_k$ represents the target degree vector; $\{\hat{P}(k,k')\}_{k,k'}$ represents the estimate of the joint degree distribution; $\{m^*(k,k')\}_k$ represents the target joint degree matrix; $\{\hat{\bar{c}}(k)\}_k$ represents the estimate of the degree-dependent clustering coefficient; and $\tilde{\mathcal{G}}$ represents the graph to be generated by the proposed method.

Algorithm 2 Adjust the target degree vector to ensure that it satisfies condition (DV-2).

Input: Estimates: $\hat{n}$ and $\{\hat{P}(k)\}_k$.
Input: Target maximum degree: $k^*_{\max}$.
Input: Target degree vector: $\{n^*(k)\}_k$.
1: **if** $\sum_{k=1}^{k^*_{\max}} kn^*(k)$ is an odd number **then**
2: Select degree k such that k is an odd number and $\Delta_+(k)$ is the smallest.
3: $n(k) \leftarrow n(k) + 1$.
4: **return** $\{n^*(k)\}_k$.

Therefore, we construct $\{n^*(k)\}_k$, which satisfies the realization conditions while minimizing the error of $\{n^*(k)\}_k$ relative to the original estimate $\{\hat{n}(k)\}_k$.

Initialization step

We initialize $n^*(k)$ for each degree k using $\hat{n}$ and $\hat{P}(k)$ such that $\{n^*(k)\}_k$ satisfies condition (DV-1). Let $\mathrm{NearInt}(a)$ denote a function that returns the nearest integer to real value a and $\max(b, c)$ denote a function that returns the larger of the two integers b and c.

First, we set the target maximum degree $k^*_{\max}$ as the larger value between the maximum degree k such that $\hat{P}(k) > 0$ and the maximum degree of the node in the subgraph $\mathcal{G}'$. Then, for each degree $k = 1, \ldots, k^*_{\max}$, we set

$$n^*(k) = \begin{cases} \max(\mathrm{NearInt}(\hat{n}\hat{P}(k)), 1) & \text{if } \hat{P}(k) > 0, \\ 0 & \text{if } \hat{P}(k) = 0. \end{cases}$$

Note that we initialize $n^*(k)$ with a positive integer for each degree k such that $\hat{P}(k) > 0$. For example, if $\hat{n}\hat{P}(1) = 0.1$, we set $n^*(1) = 1$ and not $n^*(1) = 0$. This is because if we obtain a positive estimate $\hat{P}(k) > 0$ then there must be at least one node with degree k in the original graph $\mathcal{G}$ based on the definition of $\hat{P}(k)$.

Adjustment step

Then, we adjust the target degree vector $\{n^*(k)\}_k$ such that $\{n^*(k)\}_k$ satisfies condition (DV-2) as follows (see also Algorithm 2). If and only if the sum of degrees, i.e., $\sum_{k=1}^{k^*_{\max}} kn^*(k)$, is an odd number, we increase $n^*(k)$ by one for degree k ($1 \leq k \leq k^*_{\max}$) such that k is an odd number and the increase in the error of $n^*(k)$ relative to the original estimate $\hat{n}(k) = \hat{n}\hat{P}(k)$ upon increasing $n^*(k)$ by one, denoted by $\Delta_+(k)$, is the smallest value. We define $\Delta_+(k)$ as

$$\Delta_+(k) = \begin{cases} \frac{|\hat{n}(k) - (n^*(k)+1)|}{\hat{n}(k)} - \frac{|\hat{n}(k) - n^*(k)|}{\hat{n}(k)} & \text{if } \hat{P}(k) > 0, \\ \infty & \text{if } \hat{P}(k) = 0. \end{cases}$$

If there are two or more candidates for degree k with the same increase $\Delta_+(k)$, we choose the smallest degree k to minimize the increase in the number of edges in the graph to be generated, i.e., $\sum_{k=1}^{k^*_{\max}} kn^*(k)$. The target degree vector does not break condition (DV-1) because the adjustment step comprises only increasing $n^*(k)$ for some degree k.

Modification step

In general, to generate a graph that preserves a given target degree vector, we assign a target degree to each node, i.e., the degree of each node in the generated

Algorithm 3 Modify the target degree vector to ensure that it satisfies condition (DV-3).

Input: Subgraph: $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$.
Input: Estimates: $\hat{n}$ and $\{\hat{P}(k)\}_k$.
Input: Target maximum degree: $k^*_{\max}$.
Input: Target degree vector: $\{n^*(k)\}_k$.
Calculate degree d'_i of each node in the subgraph $v'_i \in \mathcal{V}'$.
for each $v'_i \in \mathcal{V}'_{\mathrm{qry}}$ in arbitrary order **do**
  $d^*_i \leftarrow d'_i$.
Calculate the present $n'(k)$ for each $k = 1, \ldots, k^*_{\max}$.
for $k = 1, \ldots, k^*_{\max}$ **do**
  $n^*(k) \leftarrow \max(n^*(k), n'(k))$.
for each $v'_i \in \mathcal{V}'_{\mathrm{vis}}$ in decreasing order of d'_i **do**
  Construct the degree sequence $\mathcal{D}_{\mathrm{seq}}(i)$.
  if $\mathcal{D}_{\mathrm{seq}}(i)$ is not empty **then**
    Select degree k uniformly randomly from $\mathcal{D}_{\mathrm{seq}}(i)$.
  else
    Select degree k such that $d'_i \le k \le k^*_{\max}$ and $\Delta_+(k)$ is the smallest.
  $d^*_i \leftarrow k$.
  $n'(k) \leftarrow n'(k) + 1$.
  $n^*(k) \leftarrow \max(n^*(k), n'(k))$.
return $\{n^*(k)\}_k$.

graph [148, 178]. Therefore, we next assign the target degree, denoted by d^*_i, of each node v'_i in the subgraph $\mathcal{G}'$ whose target degree is constrained by the degree of the subgraph. In parallel with this assignment process, we modify the target degree vector to ensure that it also satisfies another condition to realize a generated graph $\tilde{\mathcal{G}}$ that contains the subgraph $\mathcal{G}'$. Specifically, the target number of nodes with degree k, i.e., $n^*(k)$, needs to be no less than the number of nodes with the target degree k in the subgraph, which is defined as $n'(k) = \sum_{v'_i \in \mathcal{V}'} 1_{\{d^*_i = k\}}$, for each degree k:

(DV-3) $n^*(k) \ge n'(k)$ for each degree $k = 1, \ldots, k^*_{\max}$.

We modify the target degree vector to ensure that it satisfies all conditions (i.e., (DV-1), (DV-2), and (DV-3)) while minimizing the error of $\{n^*(k)\}_k$ relative to the original estimate $\{\hat{n}(k)\}_k$.

Before we assign the target degree of each node v'_i in the subgraph $\mathcal{G}'$, we clarify the relationship between the degree d'_i of v'_i in $\mathcal{G}'$ and the degree d_i of v'_i in the original graph $\mathcal{G}$:

Lemma 14. *For each node in the subgraph $v'_i \in \mathcal{V}'$, the degree d'_i in $\mathcal{G}'$ and the degree d_i in $\mathcal{G}$ satisfy*

$$d_i = d'_i \quad \text{if } v'_i \in \mathcal{V}'_{qry},$$
$$d_i \ge d'_i \quad \text{if } v'_i \in \mathcal{V}'_{vis}.$$

Proof. The first equation holds because all edges incident to a queried node are contained in $\mathcal{G}'$ based on the problem definition. The second inequality holds because a visible node is connected only to queried neighbors in $\mathcal{G}'$. □

We modify the target degree vector as follows (see also Algorithm 3). First, for each queried node $v'_i \in \mathcal{V}'_{\mathrm{qry}}$ in arbitrary order, we assign a target degree $d^*_i = d'_i$

that is the same as the degree of the node in the subgraph $\mathcal{G}'$, according to Lemma 14 (lines 2–3 in Algorithm 3). Second, we calculate the present number of nodes with target degree k in the subgraph, i.e., $n'(k)$, for each degree $k = 1, \dots, k^*_{\max}$ (line 4 in Algorithm 3). Then, we modify the target number $n^*(k)$ to $n^*(k) = n'(k)$ if and only if $n^*(k) < n'(k)$ for each degree $k = 1, \dots, k^*_{\max}$ to satisfy condition (DV-3) (lines 5–6 in Algorithm 3).

Next, for each visible node $v'_i \in \mathcal{V}'_{\text{vis}}$, we assign the target degree d^*_i such that $d^*_i \geq d'_i$, according to Lemma 14. For this purpose, we first select the visible node $v'_i \in \mathcal{V}'_{\text{vis}}$ that is not assigned the target degree and has the largest degree in the subgraph. Second, we construct a sequence of target degrees that can be assigned to v'_i, as denoted by $\mathcal{D}_{\text{seq}}(i)$, in which degree k appears $n^*(k) - n'(k)$ times for each $k = d'_i, \dots, k^*_{\max}$ (line 8 in Algorithm 3). If $\mathcal{D}_{\text{seq}}(i)$ is not empty, we choose degree k uniformly and randomly from $\mathcal{D}_{\text{seq}}(i)$ (lines 9–10 in Algorithm 3). Otherwise, we select degree k such that $d'_i \leq k \leq k^*_{\max}$ and the increase in the error $\Delta_+(k)$ is the smallest (lines 11–12 in Algorithm 3). If two or more candidates exist for degree k with the same increase $\Delta_+(k)$, we choose the smallest. Then, we assign the target degree of v'_i as $d^*_i = k$ and increase $n'(k)$ by one (lines 13–14 in Algorithm 3). We modify the target number $n^*(k)$ to $n^*(k) = n'(k)$ if and only if $n^*(k) < n'(k)$ (line 15 in Algorithm 3). We continue this procedure until we assign a target degree to all visible nodes.

Note that we assign target degrees to visible nodes in decreasing order of the degrees in the subgraph. This is because a node with a larger degree in the subgraph tends to have fewer candidate target degrees in social graphs with heavy-tailed degree distributions [15, 85, 86, 122, 154].

The target degree vector, $\{n^*(k)\}_k$, does not break condition (DV-1) if we execute the modification algorithm on the target degree vector. This is because the algorithm comprises only increasing $n^*(k)$ values for multiple degrees of k. On the other hand, this modification step may make $\{n^*(k)\}_k$ break condition (DV-2). In this case, we perform the adjustment process (Algorithm 2) again. If we execute the adjustment algorithm on the target degree vector, $\{n^*(k)\}_k$ does not break conditions (DV-1) and (DV-3). This is because the algorithm comprises only increasing $n^*(k)$ for some degree k. Therefore, $\{n^*(k)\}_k$ finally satisfies all conditions, i.e., (DV-1), (DV-2), and (DV-3).

3.4.3 Constructing a target joint degree matrix

In the second phase, we construct the target joint degree matrix based on the subgraph $\mathcal{G}'$, the target degree vector $\{n^*(k)\}_k$, and the estimates of the number of nodes $\hat{n}$, the average degree $\hat{\bar{k}}$, and the joint degree distribution $\hat{P}(k, k')$. The target joint degree matrix, as denoted by $\{m^*(k, k')\}_{k,k'}$, determines the number of edges between nodes with degree k and nodes with degree k' in the graph to be generated $\tilde{\mathcal{G}}$.

As in the case of constructing the target degree vector, the target joint degree matrix, $\{m^*(k, k')\}_{k,k'}$, needs to satisfy the following three conditions to realize a graph that preserves it:

(JDM-1) $m^*(k, k')$ is a nonnegative integer for each degree $k = 1, \dots, k^*_{\max}$ and $k' = 1, \dots, k^*_{\max}$.

(JDM-2) $m^*(k, k') = m^*(k', k)$ for each degree $k = 1, \dots, k^*_{\max}$ and each $k = 1, \dots, k^*_{\max}$ such that $k \neq k'$.

(JDM-3) $\sum_{k'=1}^{k^*_{\max}} \mu(k, k') m^*(k, k') = k n^*(k)$ for each degree $k = 1, \dots, k^*_{\max}$.

These conditions are obtained by relaxing the conditions required to realize a graph that preserves $\{m^*(k,k')\}_{k,k'}$ and contains no multiple edges or self-loops [211].

However, as in the case of constructing the target degree vector, the immediate estimate of the number of edges between nodes with degree k and nodes with degree k' obtained by $\hat{m}(k,k') = \hat{n}\hat{\bar{k}}\hat{P}(k,k')/\mu(k,k')$ (see Eqs. (3.1) and (3.3)) typically does not satisfy the realization conditions. Therefore, we construct $m^*(k,k')$ for each k and k' that satisfies the realization conditions while minimizing the error of $\{m^*(k,k')\}_{k,k'}$ relative to the original estimate $\{\hat{m}(k,k')\}_{k,k'}$.

Initialization step

First, we initialize $m^*(k,k')$ for each degree k and k' using $\hat{n}$, $\hat{\bar{k}}$, and $\hat{P}(k,k')$ such that it satisfies conditions (JDM-1) and (JDM-2). Specifically, for each degree $k = 1,\dots,k^*_{\max}$ and $k' = 1,\dots,k^*_{\max}$, we set

$$m^*(k,k') = \begin{cases} \max(\mathrm{NearInt}(\hat{n}\hat{\bar{k}}\hat{P}(k,k')/\mu(k,k'),1) & \text{if } \hat{P}(k,k') > 0, \\ 0 & \text{if } \hat{P}(k,k') = 0. \end{cases}$$

It holds that $m^*(k,k') = m^*(k',k)$ because $\hat{P}(k,k') = \hat{P}(k',k)$ holds for $k \neq k'$. Note that we initialize $m^*(k,k')$ as a positive integer for each degree k and k' such that $\hat{P}(k,k') > 0$. This is because if we obtain a positive estimate $\hat{P}(k,k') > 0$ then there must be at least one edge between nodes with degree k and nodes with degree k' in the original graph $\mathcal{G}$ based on the definition of $\hat{P}(k,k')$.

Adjustment step

Then, we adjust $m^*(k,k')$ for each k and k' to ensure that it satisfies condition (JDM-3) (see also Algorithm 4). We denote the present sum of $m^*(k,k')$ for degree $k' = 1,\dots,k^*_{\max}$ as $s(k) = \sum_{k'=1}^{k^*_{\max}} \mu(k,k')m^*(k,k')$. We also denote the target sum for degree k as $s^*(k) = kn^*(k)$. We denote the set of degrees k by which we adjust the present sum $s(k)$ as $\mathcal{D}$.

For each degree $k \in \mathcal{D}$, we repeatedly increase or decrease $m^*(k,k')$ by one for multiple degrees k' until the present sum $s(k)$ is equal to the target sum $s^*(k)$. We define

$$\mathcal{D} = \{k \mid 1 \leq k \leq k^*_{\max} \wedge s(k) \neq s^*(k)\} \cup \{1\}.$$

We include degree $k = 1$ in the set $\mathcal{D}$ to enable us to finely adjust the target joint degree matrix to ensure that it satisfies condition (JDM-3). We adjust $n^*(k)$ if and only if $s(k)$ cannot reach $s^*(k)$ only with the adjustment of $m^*(k,k')$ for multiple degrees k'.

We impose three constraints in adjusting the target joint degree matrix. First, we ensure that $m^*(k,k')$ is not less than the input lower limit denoted by $m_{\min}(k,k')$ for each k and k'. We assume that $m_{\min}(k,k') \geq 0$. The first constraint prevents $m^*(k,k')$ from violating condition (JDM-1) and is also used in the modification step, which is described in the next section. Second, if we increase (decrease) $m^*(k,k')$ by one for $k' \neq k$, we also increase (decrease) $m^*(k',k)$ by one. The second constraint prevents $m^*(k,k')$ and $m^*(k',k)$ such that $k \neq k'$ from violating condition (JDM-2). The second constraint makes it difficult to adjust the target joint degree matrix because both present sums $s(k)$ and $s(k')$ change if we change

Algorithm 4 Adjust the target joint degree matrix to ensure that it satisfies condition (JDM-3).

Input: Estimates: $\hat{n}$, $\hat{\bar{k}}$, and $\{\hat{P}(k,k')\}_{k,k'}$.
Input: Target maximum degree: $k^*_{\max}$.
Input: Target degree vector: $\{n^*(k)\}_k$.
Input: Target joint degree matrix: $\{m^*(k,k')\}_{k,k'}$.
Input: Lower limits: $\{m_{\min}(k,k')\}_{k,k'}$.
1: **for** each $k \in \mathcal{D}$ in decreasing order of k **do**
2: **if** $k = 1$ and $|s(1) - s^*(1)|$ is an odd number **then**
3: $n^*(1) \leftarrow n^*(1) + 1$.
4: **while** $s(k) \neq s^*(k)$ **do**
5: **if** $s(k) < s^*(k)$ **then**
6: Select degree $k' \in \mathcal{D}'_+(k)$ with the smallest $\Delta_+(k,k')$.
7: $m^*(k,k') \leftarrow m^*(k,k') + 1$.
8: **if** $k \neq k'$ **then**
9: $m^*(k',k) \leftarrow m^*(k',k) + 1$.
10: **else**
11: **if** $\mathcal{D}'_-(k)$ is not empty **then**
12: Select degree $k' \in \mathcal{D}'_-(k)$ with the smallest $\Delta_-(k,k')$.
13: $m^*(k,k') \leftarrow m^*(k,k') - 1$.
14: **if** $k \neq k'$ **then**
15: $m^*(k',k) \leftarrow m^*(k',k) - 1$.
16: **else**
17: **if** $k = 1$ **then**
18: $n^*(1) \leftarrow n^*(1) + 2$.
19: **else**
20: $n^*(k) \leftarrow n^*(k) + 1$.
21: **return** $\{m^*(k,k')\}_{k,k'}$.

$m^*(k,k')$ for $k' \neq k$. We address this difficulty by imposing the following third constraint: when we attempt to adjust the present sum $s(k)$ for degree k, we do not change $m^*(k,k')$ for any degree k' such that $s(k') = s^*(k')$ already holds true before adjusting the present sum $s(k)$. The third constraint prevents the sum $s(k')$ for any degree k' already been adjusted before adjusting the sum $s(k)$ from violating condition (JDM-3).

We adjust the present sum $s(k)$ for degree $k \in \mathcal{D}$ in descending order of k. This ordering is based on the following two observations: (i) the later the adjustment order of $s(k)$ is, the fewer the elements of $m^*(k,k')$ that can be changed, and (ii) the smaller degree k is, the fewer the edges that need to be added to make $s(k)$ equal to $s^*(k)$. Accordingly, when we adjust $s(k)$, we are allowed to change only $m^*(k,k')$ for degree $k' \in \mathcal{D}$ such that $k' \leq k$.

When we adjust the present sum of degree 1, i.e., $s(1)$, we first need to ensure that the absolute difference between the present and target sums, i.e., $|s(1) - s^*(1)|$, is an even number. This reason is as follows. The sum $s(1)$ increases or decreases only by an even number because we are allowed to increase or decrease only $m^*(1,1)$ due to our third constraint. Thus, if $|s(1) - s^*(1)|$ is an odd number, $s(1)$ will not reach $s^*(1)$. Therefore, in this case, we make the absolute difference an even number by increasing $n^*(1)$ by one (lines 2–3 in Algorithm 4).

If $s(k) < s^*(k)$, we increase $m^*(k,k')$ by one for degree k' (lines 5–9 in Algorithm 4). For the given k, we define the set of degrees k' for which $m^*(k,k')$ is

increased by one as

$$\mathcal{D}'_{+}(k) = \begin{cases} \{k' \mid k' \in \mathcal{D} \wedge k' \leq k\} & \text{if } s(k) \neq s^*(k) - 1, \\ \{k' \mid k' \in \mathcal{D} \wedge k' < k\} & \text{if } s(k) = s^*(k) - 1. \end{cases}$$

We exclude degree k if $s(k) = s^*(k) - 1$ to avoid increasing $s(k)$ by two, where we recall that $s(k)$ is increased by two if we increase $m^*(k,k)$ by one because of the factor $\mu(k,k) = 2$. The set $\mathcal{D}'_{+}(k)$ always contains at least one degree k'. This is because if $k > 1$, the set $\mathcal{D}'_{+}(k)$ contains at least degree $k' = 1$; otherwise, it holds that $\mathcal{D}'_{\text{inc}}(1) = \{1\}$ because our adjustment algorithm maintains $|s^*(1) - s(1)|$ as an even number. Then, we increase $m^*(k,k')$ by one for degree $k' \in \mathcal{D}'_{+}(k)$ such that the increase in the error of $m^*(k,k')$ relative to the original estimate $\hat{m}(k,k') = \hat{n}\hat{\bar{k}}\hat{P}(k,k')/\mu(k,k')$ upon increasing $m^*(k,k')$ by one, as denoted by $\Delta_{+}(k,k')$, is the smallest. We define $\Delta_{+}(k,k')$ as

$$\Delta_{+}(k,k') = \begin{cases} \frac{|\hat{m}(k,k') - (m^*(k,k')+1)|}{\hat{m}(k,k')} - \frac{|\hat{m}(k,k') - m^*(k,k')|}{\hat{m}(k,k')} & \text{if } \hat{P}(k,k') > 0, \\ \infty & \text{if } \hat{P}(k,k') = 0. \end{cases}$$

If two or more candidates for degree k' exist with the same increase $\Delta_{+}(k,k')$, we uniformly and randomly choose from among the k' values. This random selection is based on our preliminary observation.

If $s(k) > s^*(k)$, we attempt to decrease $m^*(k,k')$ by one for degree k' (lines 10–20 in Algorithm 4). We strictly adhere to the lower limit of $m^*(k,k')$, i.e., $m_{\min}(k,k')$, for each k and k'. Unless we state otherwise, we set the lower limit as

$$m_{\min}(k,k') = 0$$

for any k and k'. For the given k, we define the set of degrees k' for which $m^*(k,k')$ is decreased by one as

$$\mathcal{D}'_{-}(k) = \begin{cases} \{k' \mid k' \in \mathcal{D} \wedge k' \leq k \wedge m^*(k,k') > m_{\min}(k,k')\} \\ \qquad\qquad \text{if } s(k) \neq s^*(k) + 1, \\ \{k' \mid k' \in \mathcal{D} \wedge k' < k \wedge m^*(k,k') > m_{\min}(k,k')\} \\ \qquad\qquad \text{if } s(k) = s^*(k) + 1. \end{cases}$$

If $\mathcal{D}'_{-}(k)$ is not an empty set, we decrease $m^*(k,k')$ by one for degree $k' \in \mathcal{D}'_{-}(k)$ such that the increase in the error of $m^*(k,k')$ relative to the original estimate $\hat{m}(k,k')$ upon decreasing $m^*(k,k')$ by one, as denoted by $\Delta_{-}(k,k')$, is the smallest. We define $\Delta_{-}(k,k')$ as

$$\Delta_{-}(k,k') = \begin{cases} \frac{|\hat{m}(k,k') - (m^*(k,k')-1)|}{\hat{m}(k,k')} - \frac{|\hat{m}(k,k') - m^*(k,k')|}{\hat{m}(k,k')} & \text{if } \hat{P}(k,k') > 0, \\ \infty & \text{if } \hat{P}(k,k') = 0. \end{cases}$$

If two or more candidates for degree k' exist, we uniformly and randomly choose k' between among the k' values.

The set $\mathcal{D}'_{-}(k)$ may be an empty set due to the constraint of the lower limits. In this case, we increase the target sum $s^*(k)$ to shift the adjustment process

Algorithm 5 Modify the target joint degree matrix to ensure that it satisfies condition (JDM-4).

Input: Subgraph: $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$.
Input: Estimates: $\hat{n}$, $\hat{\bar{k}}$, and $\{\hat{P}(k,k')\}_{k,k'}$.
Input: Target maximum degree: $k^*_{\max}$.
Input: Target degree of each node in the subgraph: $\{d^*_i\}_{v_i \in V^*}$
Input: Target joint degree matrix: $\{m^*(k,k')\}_{k,k'}$.

1: Calculate $m'(k,k')$ for each $k = 1, \ldots, k^*_{\max}$ and $k' = 1, \ldots, k^*_{\max}$.
2: **for** $k_1 = 1, \ldots, k^*_{\max}$ **do**
3: **for** $k_2 = k_1, \ldots, k^*_{\max}$ **do**
4: **while** $m^*(k_1,k_2) < m'(k_1,k_2)$ **do**
5: $m^*(k_1,k_2) \leftarrow m^*(k_1,k_2) + 1$.
6: **if** $k_1 \neq k_2$ **then**
7: $m^*(k_2,k_1) \leftarrow m^*(k_2,k_1) + 1$.
8: **if** $\mathcal{D}''_-(k_1)$ is not empty **then**
9: Select degree $k_3 \in \mathcal{D}''_-(k_1)$ with the smallest $\Delta_-(k_1,k_3)$.
10: $m^*(k_1,k_3) \leftarrow m^*(k_1,k_3) - 1$.
11: **if** $k_3 \neq k_1$ **then**
12: $m^*(k_3,k_1) \leftarrow m^*(k_3,k_1) - 1$.
13: **if** $\mathcal{D}''_-(k_2)$ is not empty **then**
14: Select degree $k_4 \in \mathcal{D}''_-(k_2)$ with the smallest $\Delta_-(k_2,k_4)$.
15: $m^*(k_2,k_4) \leftarrow m^*(k_2,k_4) - 1$.
16: **if** $k_4 \neq k_2$ **then**
17: $m^*(k_4,k_2) \leftarrow m^*(k_4,k_2) - 1$.
18: **if** both degrees k_3 and k_4 have been found **then**
19: $m^*(k_3,k_4) \leftarrow m^*(k_3,k_4) + 1$.
20: **if** $k_3 \neq k_4$ **then**
21: $m^*(k_4,k_3) \leftarrow m^*(k_4,k_3) + 1$.
22: **return** $\{m^*(k,k')\}_{k,k'}$.

toward the adjustment in which we increase the present sum $s(k)$. Specifically, if $k > 1$, we increase $s^*(k)$ by k by increasing $n^*(k)$ by one; otherwise, we increase $s^*(1)$ by two while maintaining $|s^*(1) - s(1)|$ as an even number by increasing $n^*(1)$ by two (lines 16–20 in Algorithm 4).

For any degree $k \in \mathcal{D}$, the present sum $s(k)$ will reach the target sum $s^*(k)$ in a finite number of steps. The reason is as follows. If we increase $m(k,k')$ by one for degree $k' \neq k$, the present sum $s(k)$ increases by one. If we increase $m(k,k)$ by one, the present sum $s(k)$ increases by two. When $k > 1$, there is at least one degree k' such that the sum $s(k)$ is increased by one because the set $\mathcal{D}'_+(k)$ always contains degree $k' = 1$. When $k = 1$, we are allowed to increase the sum $s(1)$ by only two because $\mathcal{D}'_+(1) = \{1\}$. However, we maintain $|s(1) - s^*(1)|$ as an even number throughout the process. Therefore, $s(k)$ will reach $s^*(k)$ in the case of an adjustment by increasing. In the case of an adjustment by decreasing, although there is the case in which the set $\mathcal{D}'_-(k)$ is an empty set, we recall that we shift the process toward the adjustment by increasing in that case.

Modification step

Next, we modify the target joint degree matrix to ensure that it also satisfies the required condition for realizing a generated graph $\tilde{\mathcal{G}}$ that contains the subgraph $\mathcal{G}'$. Specifically, the target number of nodes between nodes with degree k and nodes

with degree k', i.e., $m^*(k,k')$, should be no less than the number of edges between nodes with target degree k and nodes with target degree k' in the subgraph, which is defined as $m'(k,k') = \sum_{(v'_i,v'_j)\in\mathcal{E}'} 1_{\{(d^*_i=k\wedge d^*_j=k')\vee(d^*_i=k'\wedge d^*_j=k)\}}$, for each k and k':

(JDM-4) $m^*(k,k') \geq m'(k,k')$ for each $k = 1,\ldots,k^*_{\max}$ and $k' = 1,\ldots,k^*_{\max}$.

We modify the target joint degree matrix to ensure that it satisfies all conditions, i.e., (JDM-1), (JDM-2), (JDM-3), and (JDM-4), while minimizing the error of $\{m^*(k,k')\}_{k,k'}$ relative to the original estimate $\{\hat{m}(k,k')\}_{k,k'}$.

The basic idea behind our modification algorithm for the target joint degree matrix is as follows. Suppose that $m^*(k_1,k_2) < m'(k_1,k_2)$ for a pair of degrees k_1 and k_2; hence, we need to modify $m^*(k_1,k_2)$ to ensure that $m^*(k_1,k_2) \geq m'(k_1,k_2)$ holds. A simple modification involves forcibly increasing $m^*(k_1,k_2)$ to $m^*(k_1,k_2) = m'(k_1,k_2)$. However, if one performs this forced increase on multiple pairs of k_1 and k_2, the sum $s(k)$ will violate condition (JDM-3) for multiple degrees k, and the target number of edges in the graph to be generated, i.e., $\sum_{k=1}^{k^*_{\max}}\sum_{k'=k}^{k^*_{\max}} m^*(k,k')$, will cumulatively increase. Therefore, if we increase $m^*(k_1,k_2)$ by one, we attempt to decrease $m^*(k_1,k_3)$ by one for degree k_3 such that $k_3 \neq k_2$ to ensure that both sums $s(k_1)$ and $s(k_2)$ are retained, minimizing the violation of condition (JDM-3) and the increase in the target number of edges.

Specifically, we modify the target joint degree matrix as follows (see also Algorithm 5). For each pair of degrees k_1 and k_2 ($1 \leq k_1 \leq k^*_{\max},\ k_1 \leq k_2 \leq k^*_{\max}$), we repeat the following procedure until $m^*(k_1,k_2)$ is not less than $m'(k_1,k_2)$. First, we increase $m^*(k_1,k_2)$ by one (line 5 in Algorithm 5). If $k_1 \neq k_2$, we also increase $m^*(k_2,k_1)$ by one (lines 6–7 in Algorithm 5). Second, we attempt to find a degree k_3 such that $k_3 \neq k_2$ and $m^*(k_1,k_3)$ can be decreased by one to ensure that the sum $s(k_1)$ is retained. We define a set of such degrees for the given degree k as

$$\mathcal{D}''_-(k) = \{k' \mid 1 \leq k' \leq k^*_{\max} \wedge k' \neq k \wedge m^*(k,k') > m'(k,k')\}.$$

If $\mathcal{D}''_-(k_1)$ is not empty, we select $k_3 \in \mathcal{D}''_-(k_1)$ with the smallest $\Delta_-(k_1,k_3)$ and decrease $m^*(k_1,k_3)$ by one (lines 8–10 in Algorithm 5). If there are two or more candidates for k_3 with the same increase $\Delta_-(k_1,k_3)$, we uniformly and randomly choose k_3 from among those candidates. If $k_3 \neq k_1$, we also decrease $m^*(k_3,k_1)$ by one (lines 11–12 in Algorithm 5). Third, since $m^*(k_2,k_1)$ has been increased by one, we attempt to decrease $m(k_2,k_4)$ by one for degree k_4 such that $k_4 \neq k_2$ to ensure that the sum $s(k_2)$ is retained. If the set $\mathcal{D}''_-(k_2)$ is not empty, we select the $k_4 \in \mathcal{D}''_-(k_2)$ with the smallest $\Delta_-(k_2,k_4)$ and decrease $m^*(k_2,k_4)$ by one (lines 13–15 in Algorithm 5). If two or more candidates for k_4 exist with the same increase $\Delta_-(k_2,k_4)$, we uniformly and randomly choose from among the k_4 values. If $k_2 \neq k_4$, we also decrease $m^*(k_4,k_2)$ by one (lines 16–17 in Algorithm 5). Finally, if and only if both k_3 and k_4 have been found, we increase $m^*(k_3,k_4)$ by one and increase $m^*(k_4,k_3)$ by one if $k_3 \neq k_4$ to ensure that both sums $s(k_3)$ and $s(k_4)$ are retained (lines 18–21 in Algorithm 5).

If we find either of degrees k_3 and k_4, $m^*(k_1,k_2)$ is increased by one while preserving the target number of edges, i.e., $\sum_{k=1}^{k^*_{\max}}\sum_{k'=k}^{k^*_{\max}} m^*(k,k')$. This is because $m^*(k_1,k_2)$ is increased by one, and either $m^*(k_1,k_3)$ or $m^*(k_2,k_4)$ is decreased by one. Furthermore, if both k_3 and k_4 have been found, $m^*(k_1,k_2)$ is increased by one while preserving the target number of edges and the sum $s(k)$ for any degree k because all sums $s(k_1), s(k_2), s(k_3)$, and $s(k_4)$ are retained.

The target joint degree matrix, i.e., $\{m^*(k,k')\}_{k,k'}$, does not break conditions (JDM-1) and (JDM-2) if we execute the modification algorithm on the target joint

Algorithm 6 Construct a graph that preserves the target degree vector and the target joint degree matrix from the subgraph.

Input: Subgraph: $\mathcal{G}'$.
Input: Target degree vector: $\{n^*(k)\}_k$.
Input: Target joint degree matrix: $\{m^*(k,k')\}_{k,k'}$.
1: $\tilde{\mathcal{G}} \leftarrow \mathcal{G}'$.
2: Add $(\sum_{k=1}^{k^*_{\max}} n^*(k)) - n'$ nodes to a set of nodes in $\tilde{\mathcal{G}}$.
3: Construct the degree sequence $\mathcal{D}_{\text{seq}}$ in which degree k appears $n^*(k) - n'(k)$ times for $k = 1, \ldots, k^*_{\max}$.
4: Randomly shuffle $\mathcal{D}_{\text{seq}}$.
5: **for** each added node $\tilde{v}_i \in \mathcal{V}_{\text{add}}$ in arbitrary order **do**
6: $k \leftarrow$ the last element in $\mathcal{D}_{\text{seq}}$.
7: Remove the last element in $\mathcal{D}_{\text{seq}}$.
8: $d^*_i \leftarrow k$.
9: **for** each node $\tilde{v}_i \in \mathcal{V}_{\text{qry}} \cup \mathcal{V}_{\text{vis}}$ **do**
10: Attach $d^*_i - d'_i$ half-edges to $\tilde{v}_i$.
11: **for** each node $\tilde{v}_i \in \mathcal{V}_{\text{add}}$ **do**
12: Attach d^*_i half-edges to $\tilde{v}_i$.
13: **for** $k = 1, \ldots, k^*_{\max}$ **do**
14: **for** $k' = k, \ldots, k^*_{\max}$ **do**
15: **for** $i = 1$ to $m^*(k,k') - m'(k,k')$ **do**
16: Uniformly and randomly select a free half-edge of the nodes with the target degree k and a free half-edge of the nodes with the target degree k' and connect them.
17: **return** $\tilde{\mathcal{G}}$.

degree matrix. On the other hand, $\{m^*(k,k')\}_{k,k'}$ may break condition (JDM-3). In this case, we again execute the adjustment algorithm on the target joint degree matrix (Algorithm 4), with the lower limit $m_{\min}(k,k')$ set as

$$m_{\min}(k,k') = m'(k,k')$$

for each degree k and k' to ensure that $\{m^*(k,k')\}_{k,k'}$ retains condition (JDM-4). If we perform the adjustment algorithm again, $\{m^*(k,k')\}_{k,k'}$ still satisfies conditions (JDM-1), (JDM-2), and (JDM-4), and hence, it finally satisfies all conditions, i.e., (JDM-1), (JDM-2), (JDM-3), and (JDM-4).

3.4.4 Adding nodes and edges to the subgraph

In the third phase, we add nodes and edges to the subgraph $\mathcal{G}'$ to ensure that the graph to be generated, $\tilde{\mathcal{G}}$, preserves the target degree vector $\{n^*(k)\}_k$ and the target joint degree matrix $\{m^*(k,k')\}_{k,k'}$. It is trivial to construct a graph that preserves the given $\{n^*(k)\}_k$ and $\{m^*(k,k')\}_{k,k'}$ from an empty graph [76, 148, 211]. We extend the existing construction procedure to the case of constructing the graph $\tilde{\mathcal{G}}$ that preserves $\{n^*(k)\}_k$ and $\{m^*(k,k')\}_{k,k'}$ from the subgraph $\mathcal{G}'$ (see also Algorithm 6).

First, we set the graph $\tilde{\mathcal{G}}$ as the subgraph $\mathcal{G}'$ (line 1 in Algorithm 6). Second, we add $(\sum_{k=1}^{k^*_{\max}} n^*(k)) - n'$ nodes to a set of nodes in the subgraph such that $\tilde{\mathcal{G}}$ contains $\sum_{k=1}^{k^*_{\max}} n^*(k)$ nodes, where n' is the number of nodes in the subgraph and $\sum_{k=1}^{k^*_{\max}} n^*(k)$ is the target number of nodes in $\tilde{\mathcal{G}}$ (line 2 in Algorithm 6). We denote the set of added nodes as $\mathcal{V}_{\text{add}}$. We denote the set of nodes in $\tilde{\mathcal{G}}$ as $\tilde{\mathcal{V}}$. It holds

Algorithm 7 Rewire edges to ensure that the generated graph preserves the estimate of the degree-dependent clustering coefficient.

Input: Generated graph: $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$.
Input: Estimate: $\{\hat{\bar{c}}(k)\}_k$.
Input: Coefficient of the number of rewiring attempts: R_{C}.
1: $\tilde{\mathcal{E}}_{\mathrm{rew}} \leftarrow$ a set of candidate edges to be rewired in $\tilde{\mathcal{G}}$.
2: $R \leftarrow R_{\mathrm{C}}|\tilde{\mathcal{E}}_{\mathrm{rew}}|$ // the number of rewiring attempts.
3: $\{\tilde{\bar{c}}(k)\}_k \leftarrow$ the present degree-dependent clustering coefficient of $\tilde{\mathcal{G}}$.
4: $D \leftarrow L^1$ distance between $\{\tilde{\bar{c}}(k)\}_k$ and $\{\hat{\bar{c}}(k)\}_k$.
5: **for** $r' = 1$ to R **do**
6: $(\tilde{v}_i, \tilde{v}_j), (\tilde{v}_a, \tilde{v}_b) \leftarrow$ random edge pair in $\tilde{\mathcal{E}}_{\mathrm{rew}}$.
7: $\{\tilde{\bar{c}}_{\mathrm{rew}}(k)\}_k \leftarrow$ degree-dependent clustering coefficient when the selected edge pair is rewired.
8: $D_{\mathrm{rew}} \leftarrow L^1$ distance between $\{\tilde{\bar{c}}_{\mathrm{rew}}(k)\}_k$ and $\{\hat{\bar{c}}(k)\}_k$.
9: **if** $D_{\mathrm{rew}} < D$ **then**
10: Remove edges $(\tilde{v}_i, \tilde{v}_j)$ and $(\tilde{v}_a, \tilde{v}_b)$.
11: Add edges $(\tilde{v}_i, \tilde{v}_b)$ and $(\tilde{v}_a, \tilde{v}_j)$.
12: Update $\tilde{\mathcal{E}}_{\mathrm{rew}}$.
13: $D \leftarrow D_{\mathrm{rew}}$.
14: **return** $\tilde{\mathcal{G}}$.

that $\tilde{\mathcal{V}}$ is a union of three disjoint sets, i.e., $\mathcal{V}_{\mathrm{qry}}$, $\mathcal{V}_{\mathrm{vis}}$, and $\mathcal{V}_{\mathrm{add}}$. Third, for each degree $k = 1, \ldots, k^*_{\max}$ we arbitrarily assign a target degree k to the $n^*(k) - n'(k)$ nodes that are not assigned a target degree in $\mathcal{V}_{\mathrm{add}}$ (lines 3–8 in Algorithm 6). Note that $n^*(k) - n'(k) \geq 0$ always holds true because $\{n^*(k)\}_k$ satisfies condition (DV-3). Fourth, we ensure that each node in the subgraph $\tilde{v}_i \in \mathcal{V}_{\mathrm{qry}} \cup \mathcal{V}_{\mathrm{vis}}$ has $d^*_i - d'_i$ half-edges, where d^*_i is the target degree of $\tilde{v}_i$ and d'_i is the degree of $\tilde{v}_i$ in the subgraph (lines 9–10 in Algorithm 6). Fifth, we ensure that each added node $\tilde{v}_i \in \mathcal{V}_{\mathrm{add}}$ has d^*_i half-edges (lines 11–12 in Algorithm 6). Finally, for each degree $k = 1, \ldots, k^*_{\max}$ and $k' = k, \ldots, k^*_{\max}$, we repeat the following procedure $m^*(k, k') - m'(k, k')$ times: we randomly connect a free half-edge of the nodes with the target degree k and a free half-edge of the nodes with the target degree k' (lines 13–15 in Algorithm 6).

3.4.5 Rewiring edges in the generated graph

In general, it is practically difficult to generate a graph that exactly preserves a given degree-dependent clustering coefficient because the clustering coefficients of multiple nodes simultaneously change if an edge is added or removed [180, 218]. In practice, one performs a large number of rewiring attempts of edges in a given graph to ensure that the graph approximately preserves the given degree-dependent clustering coefficient [84, 148, 180].

We perform the following process of rewiring edges in the generated graph $\tilde{\mathcal{G}}$ to ensure that $\tilde{\mathcal{G}}$ approximately preserves $\{\hat{\bar{c}}(k)\}_k$ (see also Algorithm 7). We first uniformly and randomly select an edge pair $(\tilde{v}_i, \tilde{v}_j) \in \tilde{\mathcal{E}}_{\mathrm{rew}}$ and $(\tilde{v}_{i'}, \tilde{v}_{j'}) \in \tilde{\mathcal{E}}_{\mathrm{rew}}$ such that the degrees of $\tilde{v}_i$ and $\tilde{v}_{i'}$ are equal, where $\tilde{\mathcal{E}}_{\mathrm{rew}}$ is a set of candidate edges to be rewired. We define $\tilde{\mathcal{E}}_{\mathrm{rew}}$ as

$$\tilde{\mathcal{E}}_{\mathrm{rew}} = \tilde{\mathcal{E}} \setminus \mathcal{E}', \tag{3.4}$$

where $\tilde{\mathcal{E}}$ represents a set of edges in $\tilde{\mathcal{G}}$. Then, we replace $(\tilde{v}_i, \tilde{v}_j)$ and $(\tilde{v}_{i'}, \tilde{v}_{j'})$ with $(\tilde{v}_i, \tilde{v}_{j'})$ and $(\tilde{v}_{i'}, \tilde{v}_j)$ if and only if the normalized L^1 distance between the

estimated and present degree-dependent clustering coefficients, denoted by D, decreases when we rewire the edges. We define D as

$$D = \frac{\sum_{k=1}^{k^*_{\max}} |\tilde{\bar{c}}(k) - \hat{\bar{c}}(k)|}{\sum_{k=1}^{k^*_{\max}} \hat{\bar{c}}(k)},$$

where $\{\tilde{\bar{c}}(k)\}_k$ represents the present degree-dependent clustering coefficient of $\tilde{\mathcal{G}}$. If the rewiring is accepted, we update the set $\tilde{\mathcal{E}}_{\mathrm{rew}}$. We repeat this rewiring attempt a sufficiently large number of $R = R_{\mathrm{C}}|\tilde{\mathcal{E}}_{\mathrm{rew}}|$ times, where R_{C} is a coefficient of the number of rewiring attempts.

The rewiring process exactly preserves both the degree vector $\{n^*(k)\}_k$ and the joint degree matrix $\{m^*(k,k')\}_{k,k'}$ of the generated graph $\tilde{\mathcal{G}}$ for the following two reasons. First, the rewiring process preserves the degree of each node and hence preserves $\{n^*(k)\}_k$. Second, the rewiring preserves $m^*(k,k')$ for each k and k' because the degrees of $\tilde{v}_i$ and $\tilde{v}_j$ are equal [84, 148, 180].

We empirically require the rewiring of several hundred times the number of candidate edges [84, 180]. Thus, the exact recalculation of $\{\tilde{\bar{c}}(k)\}_k$ per rewiring attempt is not practical. In practice, it is sufficient to update the difference in the number of triangles to which only nodes that are involved in the rewiring, i.e., $v_i, v_j, v_{i'}, v_{j'}$, and their neighbors, belong. Updating the number of triangles to which a node involved in one rewiring attempt requires an average time of $O(\tilde{\bar{k}}^2)$, where $\tilde{\bar{k}}$ represents the average degree of $\tilde{\mathcal{G}}$. In total, the rewiring algorithm requires an average time of $O(\tilde{\bar{k}}^2 R_{\mathrm{C}}|\tilde{\mathcal{E}}_{\mathrm{rew}}|)$.

The rewiring process exactly preserves the structure of the subgraph $\mathcal{G}'$ of $\tilde{\mathcal{G}}$ because we exclude the edges in the subgraph from the candidate edges to be rewired, as shown in Eq. (3.4). In contrast, every edge in a given graph is a candidate edge to be rewired in Gjoka et al.'s procedure (i.e., $\tilde{\mathcal{E}}_{\mathrm{rew}} = \tilde{\mathcal{E}}$) [84]. This is because Gjoka et al.'s rewiring process does not use any structure of the subgraph sampled by a random walk. Our rewiring procedure has two advantages over Gjoka et al.'s procedure because of the reduction in the number of candidate edges to be rewired: (i) our method is more likely to succeed in the rewiring of edges such that the generated graph approximately preserves $\{\hat{\bar{c}}(k)\}_k$, and (ii) the proposed procedure reduces the rewiring time to $O(\tilde{\bar{k}}^2 R_{\mathrm{C}}(|\tilde{\mathcal{E}}| - |\mathcal{E}'|))$ from $O(\tilde{\bar{k}}^2 R_{\mathrm{C}}|\tilde{\mathcal{E}}|)$ in Gjoka et al.'s procedure.

3.5 Experimental Design

We evaluate the proposed method in terms of the accuracy of structural properties, the visual representation of generated graphs, and the generation time. We conduct all experiments on a Linux server with an Intel Xeon E5-2698 (2.20 GHz) processor and 503 GB of main memory. All code is implemented in C++. The datasets and source code used in our experiments are available at Ref. [159].

3.5.1 Datasets

We use seven datasets of social graphs that are publicly available at Refs. [137, 196]. We preprocessed each dataset by first removing multiple edges and the directions of edges from the original graph and then by extracting the largest connected component. Table 3.1 lists the numbers of nodes and edges in all the graphs used in our experiments.

Table 3.1: Datasets.

Dataset	Number of nodes	Number of edges
Anybeat [196]	12,645	49,132
Brightkite [196]	56,739	212,945
Epinions [137]	75,877	405,739
Slashdot [196]	77,360	469,180
Gowalla [196]	196,591	950,327
Livemocha [196]	104,103	2,193,083
YouTube [137]	1,134,890	2,987,624

3.5.2 Structural properties of interest

We focus on 12 structural properties of a given graph.

1. Number of nodes, n.
2. Average degree, $\bar{k}$.
3. Degree distribution, $\{P(k)\}_k$.
4. Neighbor connectivity [33], as denoted by $\{\bar{k}_{\mathrm{nn}}(k)\}_k$. We define

$$\bar{k}_{\mathrm{nn}}(k) = \frac{1}{n(k)} \sum_{i=1,\ d_i=k}^{n} \frac{1}{k} \sum_{j=1}^{n} A_{i,j} d_j$$

   for each k. This property measures the average degree of neighbors of nodes with degree k, which is a coarse-grained version of the joint degree distribution [148, 180].

5. Network clustering coefficient [33], as defined by

$$\bar{c} = \frac{1}{n} \sum_{i=1}^{n} \frac{2t_i}{d_i(d_i-1)}.$$

6. Degree-dependent clustering coefficient, $\{\bar{c}(k)\}_k$.
7. Edgewise shared partner distribution [104], as denoted by $\{P(s)\}_s$. We define

$$P(s) = \frac{1}{m} \sum_{(v_i,v_j)\in\mathcal{E}, i<j} 1_{\{\mathrm{sp}(i,j)=s\}}$$

   for each s, where $\mathrm{sp}(i,j) = \sum_{k=1,\ k\neq i,\ k\neq j}^{n} A_{i,k}A_{j,k}$. This property measures the proportion of edges that have s common neighbors.

8. Average shortest-path length, as denoted by $\bar{l}$. We define

$$\bar{l} = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} l_{i,j},$$

   where $l_{i,j}$ denotes the shortest-path length between v_i and v_j.

9. Shortest-path length distribution, as denoted by $\{P(l)\}_l$. We define

$$P(l) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1_{\{l_{i,j}=l\}}.$$

10. Diameter, which is the longest shortest-path length between two nodes and is denoted by $l_{\max}$.

11. Degree-dependent betweenness centrality, as denoted by $\{\bar{b}(k)\}_k$. We define

$$\bar{b}(k) = \frac{1}{n(k)} \sum_{i=1,\ d_i=k}^{n} b_i,$$

where b_i is the betweenness centrality of v_i. We define

$$b_i = \sum_{j=1,\ j\neq i}^{n} \sum_{k=1,\ k\neq i,\ k\neq j}^{n} \frac{\sigma_{j,k}(i)}{\sigma_{j,k}},$$

where $\sigma_{j,k}(i)$ is the number of shortest paths between node v_j and node v_k that pass through node v_i and $\sigma_{j,k}$ is the number of shortest paths between node v_j and node v_k. This property measures the average betweenness centrality of the nodes with degree k [148, 180].

12. Largest eigenvalue of an adjacency matrix $\boldsymbol{A}$, as denoted by λ_1.

We regard properties (1)–(7) as local structural properties and properties (8)–(12) as global structural properties. For the properties involving shortest paths (i.e., $\bar{l}$, $\{P(l)\}_l$, $l_{\max}$, and $\{\bar{b}(k)\}_k$), we calculate those of the largest connected component of a given graph. To reduce the simulation time, we use the parallel algorithms presented in Ref. [22] to calculate $\bar{l}, \{P(l)\}_l, l_{\max}$, and $\{\bar{b}(k)\}_k$ of a given graph. Note that the use of these parallel algorithms does not affect the performance of each method.

3.5.3 Accuracy measure

To measure the accuracy of the structural properties of a generated graph, we calculate the normalized L^1 distance for each of the 12 structural properties between the original and generated graphs, as in Ref. [84]. For each structural property, we denote the vector representing the property of the original graph as $\boldsymbol{x}$ and the vector representing that of the generated graph as $\tilde{\boldsymbol{x}}$. We define the normalized L^1 distance between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ as $\sum_i |\tilde{x}_i - x_i| / \sum_i x_i$. For example, the L^1 distance between the degree distribution of the original graph, $\{P(k)\}_k$, and that of the generated graph, as denoted by $\{\tilde{P}(k)\}_k$, is given by $\sum_k |\tilde{P}(k) - P(k)| / \sum_k P(k)$. For the scalar properties (i.e., n, $\bar{k}$, $\bar{c}$, $l_{\max}$, and λ_1), the L^1 distance is equivalent to the relative error. For example, the L^1 distance between the number of nodes in the original graph, n, and that in the generated graph, as denoted by $\tilde{n}$, is given by $|\tilde{n} - n|/n$.

3.5.4 Methods to be compared

We compare our method with two existing methods.

- **Subgraph sampling** [13, 15, 122, 136, 154, 239]. One constructs a subgraph induced from a set of edges obtained using a crawling method. The crawling method is arbitrary. Therefore, we consider three well-used crawling methods in addition to a random walk (RW).
  - Breadth-first search (BFS) [45, 121, 154, 197, 239]. One selects a seed node and explores all of its neighbors. Then, one traverses the earliest explored node, and explores all of its neighbors that have not been traversed. One repeats this procedure.
  - Snowball sampling [15, 89, 106, 134, 197]. All the neighbors are not explored, unlike the BFS procedure, and at most k neighbors are chosen randomly at every iteration.
  - Forest fire sampling (FF) [13, 65, 136, 197]. FF is a stochastic version of snowball sampling. At every iteration, one explores a random proportion of neighbors. The proportion is sampled from a geometric distribution with the mean $p_f/(1-p_f)$, where p_f is a parameter. Note that this process can finish before a target fraction of nodes is sampled. In this case, we uniformly randomly select a node from the sampled nodes and revive the process from the node, as in Ref. [121].
- **Gjoka et al.'s method** [84]. This method generates a graph that preserves the estimates of local structural properties. This method does not use any structure of the subgraph sampled by a random walk. Unfortunately, we found that it is difficult to reproduce the original method based on their paper and the source code [84]. We describe how to implement the reproducible version of Gjoka et al.'s method in Section 3.9.

We apply each method in a single run as follows. We first uniformly and randomly select a seed node from a set of nodes. Then, we start BFS, snowball sampling, FF, and RW from the same seed. We continue each sampling procedure until the percentage of queried nodes reaches a given value. For subgraph sampling using RW, Gjoka et al's method, and the proposed method, we perform these methods for the same RW to achieve a fair comparison.

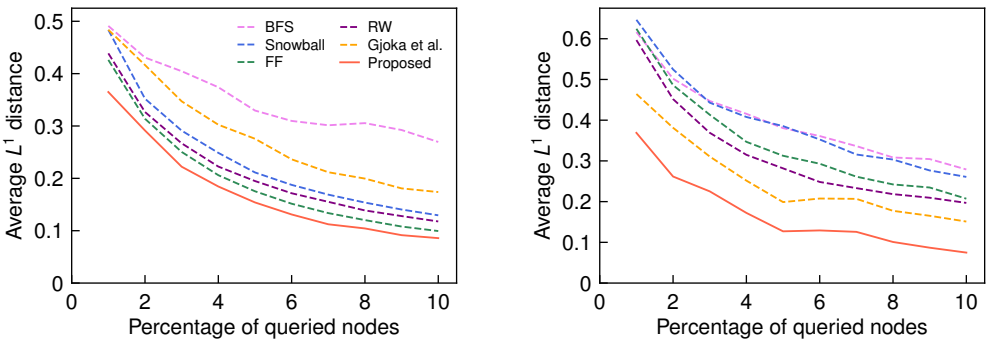
3.5.5 Parameters

In snowball sampling, we set $k = 50$, as in Ref. [197]. In FF, we set $p_f = 0.7$, as in Ref. [13]. In the proposed and Gjoka et al.'s methods, we set the coefficient of the number of rewiring attempts as $R_{\mathrm{C}} = 500$, based on Ref. [180].

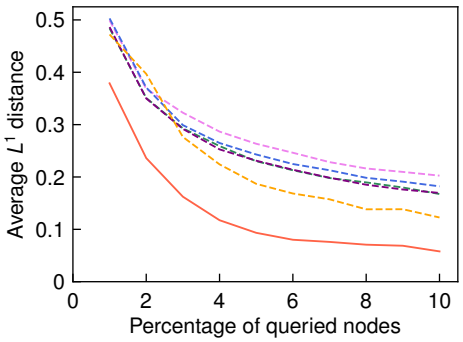
3.6 Experimental Results

3.6.1 Accuracy of structural properties

Figure 3.3 shows the average L^1 distance over the 12 structural properties from different methods for the Anybeat, Brightkite, and Epinions datasets, with the percentage of queried nodes varying from 1% to 10%. We observe that the proposed method outperforms the compared methods in terms of the average L^1 distance for all the percentages of queried nodes. Specifically, the proposed method improves the lowest average L^1 distance among the compared methods by 13.1%, 50.5%, and 52.8% (i.e., from 0.099 to 0.086, from 0.151 to 0.075, and from 0.123 to 0.058, respectively) using 10% queried nodes on the Anybeat, Brightkite, and Epinions graphs, respectively.



(a) Anybeat (b) Brightkite



(c) Epinions

Figure 3.3: Average L^1 distance over the 12 structural properties from different methods. We vary the percentage of queried nodes from 1% to 10% in increments of 1%. All results are the average over 10 runs.

Table 3.2 shows the L^1 distance for each structural property from different methods using 10% queried nodes on the Slashdot, Gowalla, and Livemocha datasets. We first compare the proposed method with subgraph sampling using BFS, snowball, FF, and RW. First, the proposed method typically improves the L^1 distance for n, $\bar{k}$, $\{P(k)\}_k$, and $\{\bar{k}_{\mathrm{nn}}(k)\}_k$. Second, the proposed method typically worsens the L^1 distance for $\bar{c}$ and $\{\bar{c}(k)\}_k$. This is because the generated graph does not exactly preserves the estimate of the node clustering due to the rewiring process. Third, in many cases, the proposed method improves the L^1 distance for global properties, i.e., $\bar{l}$, $\{P(l)\}_l$, $l_{\max}$, $\{\bar{b}(k)\}_k$, and λ_1.

We then compare the proposed method with Gjoka et al.'s method. First, the proposed method achieves comparable or sometimes better L^1 distances for n, $\bar{k}$, $\{P(k)\}_k$, and $\{\bar{k}_{\mathrm{nn}}(k)\}_k$. This stems from our design of the algorithms for constructing the target degree vector and the target joint degree matrix while minimizing their errors relative to the original estimates. Second, the proposed method improves the L^1 distance for $\{\bar{c}(k)\}_k$. This is because the proposed method is more likely to succeed in the rewiring of edges to approach the estimate of $\{\hat{\bar{c}}(k)\}_k$ because the edges in the sampled subgraph are excluded from the candidate edges to be rewired. Third, the proposed method improves the L^1 distance for $\{P(s)\}_s$ but worsens that for $\bar{c}$. Fourth, the proposed method often improves the L^1 distance for global properties, i.e., $\bar{l}$, $\{P(l)\}_l$, $l_{\max}$, $\{\bar{b}(k)\}_k$, and λ_1.

Table 3.3 shows the average and standard deviation of the L^1 distance for the 12 properties from different methods using 10% queried nodes. The proposed method achieves the lowest average and standard deviation for the six datasets.

Table 3.2: L^1 distance of each property from different methods using 10% queried nodes. All results are the average over 10 runs. The lowest value is shown in bold.

Dataset	Method	n	$\bar{k}$	$P(k)$	$\bar{k}_{\text{nn}}(k)$	$\bar{c}$	$\bar{c}(k)$	$P(s)$	$\bar{l}$	$P(l)$	$l_{\max}$	$\bar{b}(k)$	λ_1
Slashdot	BFS	0.272	0.032	0.082	0.126	0.050	0.172	0.092	0.088	0.368	0.475	0.210	0.017
	Snowball	0.248	0.043	0.074	0.102	0.057	0.152	0.092	0.086	0.356	0.392	0.108	0.013
	FF	0.237	0.042	0.073	0.102	0.029	0.164	0.095	0.083	0.349	0.300	0.094	0.014
	RW	0.242	0.042	0.072	0.102	**0.023**	**0.150**	0.099	0.084	0.352	0.225	0.100	**0.011**
	Gjoka et al.	**0.026**	**0.024**	0.057	0.100	0.097	0.708	0.353	**0.018**	**0.091**	**0.033**	0.258	0.016
	Proposed	0.029	0.027	**0.056**	**0.042**	0.097	0.205	**0.034**	0.025	0.101	0.058	**0.068**	**0.011**
Gowalla	BFS	0.432	0.100	0.356	0.324	0.234	0.153	0.098	0.203	0.851	0.556	0.337	0.009
	Snowball	0.442	0.038	0.323	0.191	0.154	0.095	**0.080**	0.168	0.627	0.413	0.244	0.007
	FF	0.408	0.032	0.280	0.149	0.102	0.072	0.087	0.140	0.511	0.256	0.176	0.008
	RW	0.395	0.040	0.273	0.133	0.072	**0.064**	0.099	0.137	0.500	0.244	**0.154**	0.006
	Gjoka et al.	0.034	0.017	**0.040**	0.102	**0.029**	0.350	0.539	0.038	0.160	**0.106**	0.767	0.469
	Proposed	**0.032**	**0.015**	0.041	**0.047**	0.273	0.110	0.142	**0.031**	**0.116**	**0.106**	0.250	**0.001**
Livemocha	BFS	0.062	0.405	0.500	0.295	0.322	0.343	0.040	0.037	0.204	**0.000**	0.614	0.111
	Snowball	0.037	0.286	0.352	0.209	0.273	0.262	**0.033**	0.018	0.106	0.017	0.301	0.057
	FF	**0.025**	0.262	0.324	0.179	0.279	**0.233**	0.039	0.015	0.088	**0.000**	0.212	0.044
	RW	0.027	0.271	0.335	0.181	0.287	0.244	0.041	0.015	0.091	**0.000**	0.228	0.045
	Gjoka et al.	0.033	0.021	**0.126**	0.142	**0.097**	0.883	0.619	0.012	0.114	0.783	0.386	**0.012**
	Proposed	0.048	**0.020**	0.132	**0.092**	0.129	0.388	0.150	**0.006**	**0.032**	0.050	**0.118**	0.020

Table 3.3: Average and standard deviation of the L^1 distance for the 12 properties from different methods using 10% queried nodes. Each result is shown as the average ± standard deviation. All results are the average over 10 runs. The lowest value is shown in bold.

Dataset	BFS	Snowball	FF	RW	Gjoka et al.	Proposed
Anybeat	0.270 ± 0.144	0.129 ± 0.098	0.099 ± 0.071	0.118 ± 0.082	0.174 ± 0.143	**0.086 ± 0.062**
Brightkite	0.279 ± 0.205	0.261 ± 0.189	0.207 ± 0.165	0.197 ± 0.159	0.151 ± 0.159	**0.075 ± 0.061**
Epinions	0.203 ± 0.192	0.182 ± 0.173	0.167 ± 0.156	0.170 ± 0.164	0.123 ± 0.132	**0.058 ± 0.055**
Slashdot	0.165 ± 0.140	0.144 ± 0.118	0.132 ± 0.105	0.125 ± 0.098	0.148 ± 0.198	**0.063 ± 0.057**
Gowalla	0.305 ± 0.223	0.232 ± 0.180	0.185 ± 0.147	0.176 ± 0.145	0.221 ± 0.242	**0.097 ± 0.089**
Livemocha	0.244 ± 0.199	0.162 ± 0.126	0.142 ± 0.113	0.147 ± 0.118	0.269 ± 0.320	**0.099 ± 0.105**

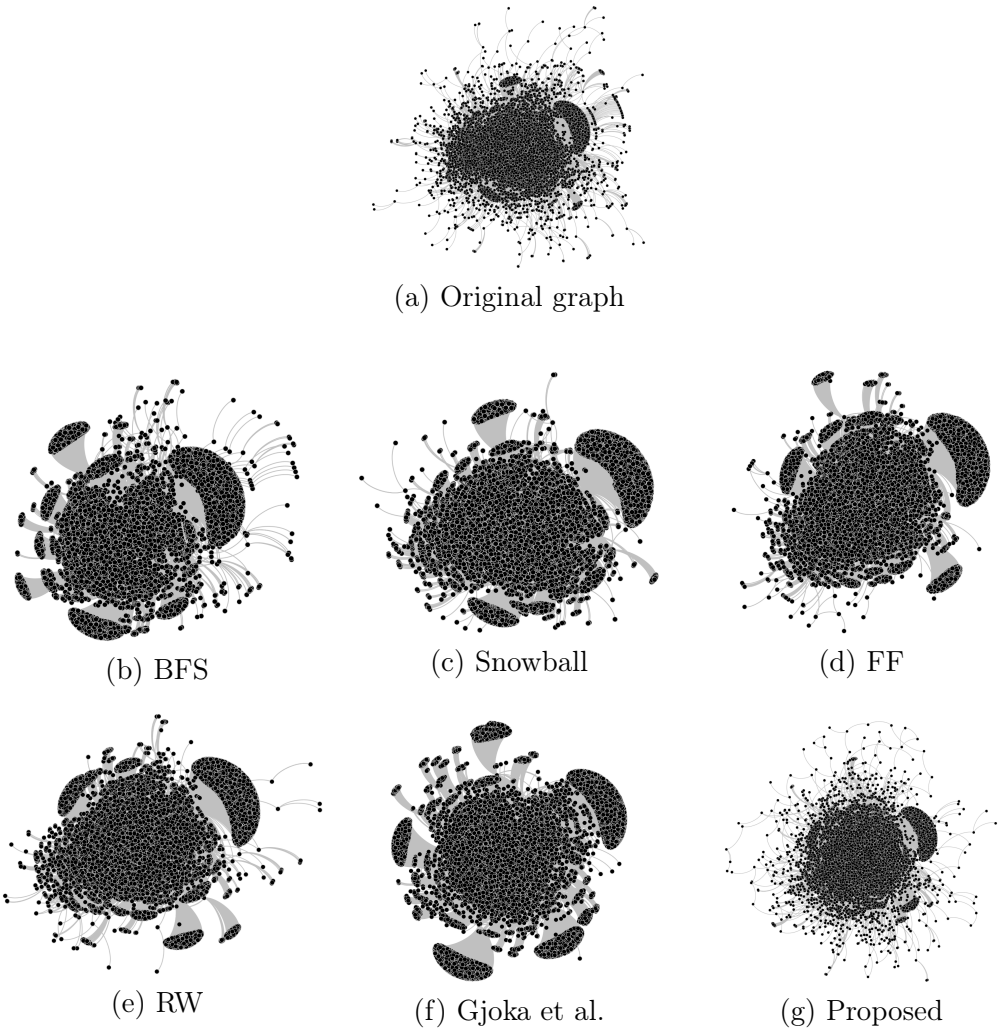
3.6.2 Graph visualization

We compare the visual representations of graphs generated by different methods.

Figure 3.4(a) shows the original graph and Figs. 3.4(b)–(g) show the visual representations of the graphs generated by each method using 10% queried nodes for the Anybeat dataset. We make the following observations. First, all subgraphs constructed using the BFS, snowball, FF, and RW methods capture the core structure consisting of high-degree nodes but not the peripheral structure consisting of low-degree nodes (see Figs. 3.4(b)–3.4(e)). This is because crawling methods typically collect samples biased toward high-degree nodes [85, 86]. Second, Gjoka et al.'s method hardly reproduces the visual representation of the original graph (see Fig. 3.4(f)) because their method does not use any structures of the sampled subgraph. Third, the proposed method successfully reproduces the original structure in the visualization (see Fig. 3.4(g)) because the generated graph preserves the structure of the sampled subgraph. Fourth, the proposed method successfully reproduces not only the core structure but also the peripheral structure, which subgraph sampling does not reproduce.

3.6.3 Generation time

We compare the generation times of different methods. Table 3.4 shows the generation times (in seconds) of the different methods using 10% queried nodes for six datasets. For the proposed and Gjoka et al.'s methods, we show both the total generation time and the running time of the rewiring process. Subgraph sampling is much faster because the construction time of the subgraph is linearly proportional to the number of edges in the subgraph. The proposed and Gjoka et al.'s methods require much longer generation times than subgraph sampling, mainly due to the process of rewiring edges. However, interestingly, the proposed method is several times faster than Gjoka et al.'s method for all six datasets, e.g., 9.0 times faster for the Anybeat dataset and 10.4 times faster for the Epinions dataset. This is because the proposed method reduces the running time of the process of rewiring edges, which is a bottleneck in the generation time, because our rewiring procedure excludes the edges in the sampled subgraph from the candidate edges to be rewired. For the proposed and Gjoka et al.'s methods, although the rewiring time is reduced upon decreasing the coefficient of the number of rewiring attempts R_{C}, we note that the reproducibility of the structural properties, including clustering coefficients, of the generated graphs is also reduced.



(a) Original graph

(b) BFS (c) Snowball (d) FF

(e) RW (f) Gjoka et al. (g) Proposed

Figure 3.4: Graph visualization for the Anybeat dataset. (a) Original graph. (b)–(g) Graphs generated by the different methods using 10% queried nodes. The black circles represent nodes and the gray curves represent edges. We used Yifan Hu's algorithm (so-called Scalable Force Directed Placement, SFDP) implemented in Gephi software [27, 101] to visualize each graph. In addition, for the graphs generated by the subgraph sampling and the proposed method (i.e., (b), (c), (d), (e), and (g)), we manually rotated each graph to ensure that the sampled subgraph approximately overlaps the corresponding part of the original graph.

Table 3.4: Generation times (in seconds) of the different methods using 10% queried nodes. For the proposed and Gjoka et al.'s methods, the total time and rewiring time are shown. All results are an average of 10 runs. The lowest value is shown in bold.

Dataset	BFS	Snowball	FF	RW	Gjoka et al.		Proposed	
					Total	Rewiring	Total	Rewiring
Anybeat	**0.008**	0.014	0.014	0.018	547	546	60.6	56.4
Brightkite	**0.120**	0.121	0.142	0.148	703	694	192	163
Epinions	**0.235**	0.248	0.243	0.259	2,914	2,878	280	168
Slashdot	**0.292**	0.305	0.307	0.310	3,086	3,064	362	296
Gowalla	**0.954**	1.08	1.19	1.30	15,907	15,735	4,255	3,679
Livemocha	**1.71**	2.22	2.07	2.29	59,114	59,054	8,645	8,331

3.6.4 Performance on the YouTube dataset

Finally, we compare the performance of the different methods using 1% queried nodes on the YouTube dataset. Table 3.5 shows the L^1 distance for each property, the average and standard deviation of the L^1 distance for the 12 properties, and the generation time for each method. First, the proposed method achieves the lowest L^1 distance for most of the 12 properties. Second, the proposed method improves the average and standard deviation of the L^1 distance over the 12 properties by 47.3% (from 0.262 to 0.138) and 41.1% (from 0.236 to 0.139), respectively, compared with the lowest value among the existing methods. Third, the generation time of the proposed method is reduced by 43.7% compared to that of Gjoka et al.'s method. As expected, the main factor in this increased speed is the reduction in the rewiring time; the proposed method requires 37,990 seconds for rewiring, whereas Gjoka et al.'s method requires 77,271 seconds. Although subgraph sampling is considerably faster, the reproducibility of the structural properties of the subgraphs is much worse than that of graphs generated by the proposed method.

Table 3.5: Performance of different methods using 1% queried nodes for the YouTube graph. The L^1 distance for each property, the average (AVG) and standard deviation (SD) of the L^1 distance for 12 properties, and the generation time (in seconds) are shown. All results are an average of 5 runs. The lowest value is shown in bold.

Method	n	$\bar{k}$	$P(k)$	$\bar{k}_{\text{nn}}(k)$	$\bar{c}$	$\bar{c}(k)$	$P(s)$	$\bar{l}$	$P(l)$	$l_{\max}$	$\bar{b}(k)$	λ_1	AVG ± SD	Time (sec)
BFS	0.752	0.039	0.191	0.661	0.630	0.531	0.173	0.296	1.31	0.700	0.796	0.058	0.512 ± 0.363	**0.724**
Snowball	0.749	0.060	0.180	0.608	0.620	0.593	0.131	0.263	1.11	0.625	0.792	0.074	0.484 ± 0.323	0.819
FF	0.642	0.088	0.160	0.450	0.500	0.471	0.196	0.237	1.02	0.508	0.487	0.036	0.400 ± 0.264	1.30
RW	0.637	0.051	0.166	0.514	0.536	0.532	0.190	0.233	1.00	0.417	0.491	0.036	0.400 ± 0.268	1.22
Gjoka et al.	**0.062**	**0.025**	**0.033**	0.255	0.025	0.707	0.361	0.067	0.232	0.250	0.566	0.563	0.262 ± 0.236	77,334
Proposed	**0.062**	**0.025**	**0.033**	**0.196**	**0.022**	**0.409**	**0.106**	**0.042**	**0.191**	**0.142**	**0.412**	**0.014**	**0.138 ± 0.139**	43,567

3.7 Conclusion

In this chapter, we proposed a method for restoring the original social graph from a small sample obtained by a random walk. The proposed method generates a graph that preserves estimates of local structural properties and the structure of the subgraph sampled by a random walk. We compared the proposed method with subgraph sampling and Gjoka et al.'s method in terms of the accuracy of 12 structural properties, the visual representation, and the generation time for generated graphs. We showed that the proposed method generates graphs that more accurately reproduce the structural properties on average and the visual representation of the original graph than the compared methods. Furthermore, the generation time of the proposed method is several times faster than that of Gjoka et al.'s method. If most of the graph data could be sampled (e.g., if 50% or more of the nodes could be queried), subgraph sampling is more effective than the proposed method because the subgraph should be almost structurally equivalent to the original graph and its construction time is fast. However, it is often difficult to collect a large sample of social graphs in practical scenarios. For example, the percentage of queried nodes was less than 1% in a case study of crawling the Facebook graph [85, 86]. Based on these results, we suggest investing in methods to complement the nodes and edges in the subgraph sampled by a random walk to realize exhaustive analyses of social graphs with limited data access.

There are several future directions for this research. The first is to study a method with theoretical guarantees for restoring the social graph. The proposed method enables us to estimate various structural properties with good accuracy on average but has one limitation; i.e., there is no guarantee of error bounds in the structural properties for the generated graphs. This is mainly because the dK-series [84, 148, 180], which is the family of generative models underlying the proposed method, does not guarantee error bounds in the structural properties of the generated graphs. The second is to study a scalable restoration method to deal with large-scale social graphs. The proposed method suffers from a considerably high computational overhead compared to subgraph sampling, although the proposed method is several times faster than Gjoka et al.'s method [84]. This is mainly due to the process of rewiring edges in a generated graph and existing studies on the dK-series [84, 148, 180] also faced high computational costs due to the rewiring process. Studying a restoration method based on scalable graph generative models that accurately reproduce the structural properties of a given graph could improve this limitation. Finally, it would be interesting to use or extend the dissimilarity [205] of a given graph to investigate how well the proposed method restores the original social graph.

3.8 Unbiasedness of an estimator of the joint degree distribution

In this section, we prove that the estimator $\hat{P}(k,k')$ proposed in [84] is an unbiased estimator of the joint degree distribution $P(k,k')$.

Lemma 15. *$\hat{P}(k,k')$ is an asymptotically unbiased estimator of $P(k,k')$.*

Proof. We show that both $\hat{P}_{\mathrm{IE}}(k,k')$ and $\hat{P}_{\mathrm{TE}}(k,k')$ are asymptotically unbiased estimators of $P(k,k')$. First, we calculate the expectation of $\Phi(k,k')$ with respect to the stationary distribution of a simple random walk.

$$
\begin{aligned}
&\mathbb{E}[\Phi(k,k')] \\
=&\mathbb{E}\left[\frac{1}{kk'}1_{\{d_{x_{i'}}=k\wedge d_{x_{j'}}=k'\}}A_{x_{i'},x_{j'}}\right] \\
=&\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{d_i}{2m}\frac{d_j}{2m} \\
&\qquad\times\mathbb{E}\left[\frac{1}{kk'}1_{\{d_{x_{i'}}=k\wedge d_{x_{j'}}=k'\}}A_{x_{i'},x_{j'}}\middle| x_{i'}=i, x_{j'}=j\right] \\
=&\frac{1}{4m^2}\sum_{i=1,\ d_i=k}^{n}\sum_{j=1,\ d_j=k}^{n}A_{i,j} \\
=&\frac{1}{2m}P(k,k').
\end{aligned}
$$

The first equation holds because of the linearity of expectation. The second equation holds for the following reasons: (i) the stationary distribution of a simple random walk in a state space of a set of nodes is given by $\pi_{\mathcal{V}}=(d_i/2m)_{i=1}^{n}$ [138]; (ii) we have the law of total expectation; and (iii) a node pair $v_{x_i'}$ and $v_{x_j'}$ is regarded as being independently sampled if those ordinal numbers in the sample sequence (i.e., $x_{i'}$ and $x_{j'}$) are sufficiently far apart [97, 112]. Then, we conclude that $\hat{P}_{\mathrm{IE}}(k,k')$ is an asymptotically unbiased estimator of $P(k,k')$ because it holds that

$$
\begin{aligned}
\mathbb{E}[\hat{n}]\mathbb{E}[\hat{d}_{\mathrm{avg}}]\mathbb{E}[\Phi(k,k')] &= nd_{\mathrm{avg}}\frac{1}{2m}P(k,k') \\
&= P(k,k').
\end{aligned}
$$

The second equation holds because of the handshaking lemma.

Next, we calculate the expectation of $\hat{P}_{\mathrm{TE}}(k,k')$:

$$
\begin{aligned}
&\mathbb{E}[\hat{P}_{\mathrm{TE}}(k,k')] \\
=&\sum_{(v_i,v_j)\in\mathcal{E}}\frac{1}{2m}\left(1_{\{d_i=k\wedge d_j=k'\}}+1_{\{d_i=k'\wedge d_j=k\}}\right) \\
=&P(k,k').
\end{aligned}
$$

The first equation holds for the following reasons; (i) we have the linearity of expectation; (ii) the stationary distribution of a simple random walk in a state space of a set of edges is given by $\pi_{\mathcal{E}}=(1/2m)_{(v_i,v_j)\in\mathcal{E}}$ [138]; and (iii) we have the law of total expectation. Therefore, we obtain the desired result. □

3.9 Implementation of Gjoka et al.'s method

In this section, we describe the implementation of Gjoka et al.'s method [84]. The underlying idea of Gjoka et al.'s method is to generate a graph that preserves

the estimates of local structural properties obtained by re-weighted random walk. However, we found that it is difficult to reproduce the original method from their paper [84] and the source code. Therefore, we implemented a reproducible version of this method by utilizing the algorithms of the proposed method as follows.

First, we obtain the estimates of local structural properties (i.e., the number of nodes $\hat{n}$, average degree $\hat{\bar{k}}$, degree distribution $\{\hat{P}(k)\}_k$, joint degree distribution $\{\hat{P}(k,k')\}_{k,k'}$, and degree-dependent clustering coefficient $\{\hat{\bar{c}}(k)\}_k$) using re-weighted random walk (see Section III. E). Second, we construct the target degree vector. To this end, we perform the initialization step described in Section IV. B and then perform the adjustment step described in Section IV. B. We do not perform the modification step for the target degree vector because Gjoka et al.'s method does not use any structural information of the subgraph sampled by a random walk. Third, we construct the target joint degree matrix. To this end, we perform the initialization step described in Section IV. C and then perform the adjustment step described in Section IV. C. We do not perform the modification step for the target joint degree matrix for the same reason given in the construction of the target degree vector. Fourth, we construct a graph that preserves the target degree vector and the target joint degree matrix from an empty graph by using the existing procedure [148, 211]. Finally, we perform a process of rewiring edges so that the final graph approximately preserves the estimate of the degree-dependent clustering coefficient $\{\hat{\bar{c}}(k)\}_k$. The rewiring procedure is the same as that described in Section IV. E, except that all the edges in the generated graph are candidates of edges to be rewired (i.e., $\tilde{\mathcal{E}}_{\mathrm{rew}} = \tilde{\mathcal{E}}$).

Chapter 4

Randomizing Hypergraphs Preserving Degree Correlation and Local Clustering

4.1 Introduction

Many social networks involve direct interactions among more than two actors and can be represented by hypergraphs, in which hyperedges encode higher-order interactions among an arbitrary number of nodes. To analyze structures and dynamics of given hypergraphs, a solid practice is to compare them with those for randomized hypergraphs that preserve some specific properties of the original hypergraphs. The existing models for randomized hypergraphs, however, preserve only up to the degree of each node and the size of each hyperedge of a given hypergraph [37, 50, 178, 201, 202].

In this chapter, we propose a family of reference models for hypergraphs, called the hyper dK-series. The original dK-series is a nested family of reference models that preserve local properties of nodes of the given dyadic network [84, 148, 180]. The hyper dK-series preserves up to the individual node's degree, node's degree correlation, node's redundancy coefficient, and/or the hyperedge's size depending on the parameter values, i.e., $d_v \in \{0, 1, 2, 2.5\}$ and $d_e \in \{0, 1\}$. Furthermore, we numerically find that the hyper dK-series with $d_v = 2.5$ more accurately preserves the shortest path length and degree distribution of the one-mode projection of the original hypergraph, which the method does not intend to preserve, than the other d_v-values. We also apply the hyper dK-series to numerical simulations of epidemic spreading and evolutionary game dynamics on empirical social hypergraphs. We find that the hyperedge's size affects these dynamics more than any of the node's properties and that the node's degree correlation (i.e., the property with $d_v = 2$) and redundancy (i.e., the property with $d_v = 2.5$) in the empirical hypergraphs promote cooperation. Our code for the hyper dK-series is available at `https://github.com/kazuibasou/hyper-dk-series`.

4.2 Preliminaries

4.2.1 Hypergraph and bipartite graph

We represent a network including higher-order interactions among two or more entities as an unweighted hypergraph that consists of a set of nodes $V = \{v_1, \ldots, v_N\}$ and a set of hyperedges $E = \{e_1, \ldots, e_M\}$, where N is the number of nodes, and M is the number of hyperedges. We assume that the original hypergraph, for which we generate sample hypergraphs using reference models, contains no multiple edges. Each hyperedge $e_j \in E$ is a subset of V with arbitrary cardinality $|e_j|$.

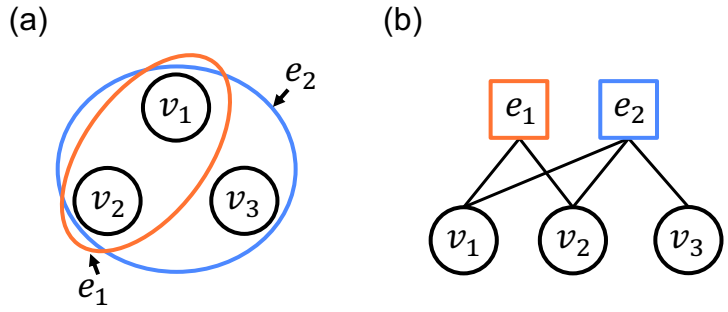


Figure 4.1: Hypergraph and the corresponding bipartite graph. (a) A hypergraph that consists of $V = \{v_1, v_2, v_2\}$ and $E = \{e_1, e_2\}$, where $e_1 = \{v_1, v_2\}$ and $e_2 = \{v_1, v_2, v_3\}$. (b) The corresponding bipartite graph, which consists of V, E, and $\mathcal{E} = \{(v_1, e_1), (v_1, e_2), (v_2, e_1), (v_2, e_2), (v_3, e_2)\}$.

We denote by $G = (V, E, \mathcal{E})$ the bipartite graph that corresponds to the given hypergraph, where $\mathcal{E}$ is a set of edges in the bipartite graph. An edge (v_i, e_j) exists between each node v_i and each hyperedge e_j if and only if v_i belongs to the hyperedge e_j in the hypergraph. We denote by $\mathcal{M} = |\mathcal{E}|$ the number of edges in G. We show in Fig. 4.1 a hypergraph and its bipartite-graph representation.

4.2.2 Local properties of nodes and hyperedges

In this section, we describe local properties of bipartite graph G some of which our reference models preserve. We denote the incidence matrix of G by $B = (B_{ij})$, where $i = 1, \ldots, N$, $j = 1, \ldots, M$, $B_{ij} = 1$ if $(v_i, e_j) \in \mathcal{E}$, and $B_{ij} = 0$ otherwise. Let $k_i = \sum_{j=1}^{M} B_{ij}$ be the degree of node v_i. We denote the size of hyperedge e_j, i.e., the number of nodes that belong to hyperedge e_j, by $s_j = \sum_{i=1}^{N} B_{ij}$.

We define the joint degree distribution of two nodes that share at least one hyperedge, which extends the joint degree distribution for dyadic networks defined in Refs. [148, 180]. Let $m(k, k')$ denote the number of hyperedges that nodes with degree $k = 1, \ldots, M$ and nodes with degree $k' = k, \ldots, M$ share. For example, in a bipartite graph shown in Fig. 4.1(b), one obtains $m(1, 2) = 2$ because node v_1 with degree $k_1 = 2$ and node v_3 with degree $k_3 = 1$ share a hyperedge e_2, and node v_2 with degree $k_2 = 2$ and node v_3 share a hyperedge e_2. Similarly, one obtains $m(1, 1) = 0$ and $m(2, 2) = 2$. We define the pairwise joint degree distribution of the node, denoted by $P(k, k')$, as

$$P(k, k') = \frac{2m(k, k')}{\sum_{j=1}^{M} s_j(s_j - 1)}. \tag{4.1}$$

Note that $P(k, k')$ is normalized, i.e., $\sum_{k=1}^{M} \sum_{k'=k}^{M} P(k, k') = 1$. We also define the average degree of the nearest neighbors of nodes with degree k, which extends the definition for dyadic networks defined in Refs. [33, 181], by

$$k_{\mathrm{nn}}(k) = \frac{\sum_{k'=1}^{M} k' P(k, k')}{\sum_{k'=1}^{M} P(k, k')}. \tag{4.2}$$

Equations (4.1) and (4.2) are consistent with the corresponding definitions for dyadic networks when $s_j = 2$ for each hyperedge $e_j \in E$.

We also examine quadruple relationships around a node in a bipartite graph, which is similar to local clustering (i.e., abundance of triangles) in dyadic networks. The redundancy coefficient of node v_i, denoted by r_i, quantifies the amount

Table 4.1: Properties of nodes and hyperedges corresponding to each d_v and d_e value. The hyper dK-series with $(d_v, d_e) = (2, 1)$, for example, preserves the quantities for $d_v = 0, 1, 2$, and $d_e = 0, 1$ shown in this table.

Parameter value	Properties to be preserved
$d_v = 0$	Average degree of the node
$d_v = 1$	Degree of each node
$d_v = 2$	Pairwise joint degree distribution of the node
$d_v = 2.5$	Degree-dependent redundancy coefficient of the node
$d_e = 0$	Average size of the hyperedge
$d_e = 1$	Size of each hyperedge

of quadratic relationships around the node in a bipartite graph [127]. It is the fraction of pairs of hyperedges to which v_i belongs such that at least one different node also belongs to both hyperedges. Formally, if $k_i > 1$, we define

$$r_i = \frac{2}{k_i(k_i - 1)} \sum_{j=1}^{M} \sum_{j'=1}^{j-1} B_{i,j} B_{i,j'} 1_{\{|\Gamma| > 0\}} \tag{4.3}$$

where we define $\Gamma = \{v_{i'} \in V \backslash \{v_i\} \mid B_{i',j} B_{i',j'} > 0\}$ and $1_{\{cond\}}$ denotes an indicator function that returns 1 if a condition *cond* holds, and it returns 0 otherwise. We define $r_i = 0$ if $k_i \in \{0, 1\}$. The degree-dependent redundancy coefficient of the node is the average of the redundancy coefficient over the nodes with degree k, i.e.,

$$r(k) = \frac{1}{n(k)} \sum_{i=1,\ k_i=k}^{N} r_i, \tag{4.4}$$

where $n(k)$ is the number of nodes with degree k.

One can also define the pairwise joint size distribution of the hyperedge and the redundancy coefficient of the hyperedge in the same way as for the node. However, we do not introduce them because we construct reference models that preserve up to the size distribution of the hyperedge. This choice stands on our observation that it is practically difficult to generate randomized bipartite graphs preserving up to pairwise correlation and quadratic relationships for both nodes' degrees and hyperedges' sizes. If one is interested in preserving the size correlation and redundancy for hyperedges instead of the corresponding quantities for nodes, one can apply our algorithm described in the following text after interchanging the nodes and hyperedges in the bipartite-graph representation of the hypergraph.

4.3 Reference Models for Hypergraphs — Hyper dK-series

In this section, we propose a family of reference models for hypergraphs that preserve local properties of nodes and hyperedges in the given hypergraph to different extents. We extend a class of reference models for dyadic networks called the dK-series [84, 148, 180] to the case of hypergraphs. The dK-series preserves some local properties of nodes (i.e., degree distribution, joint degree distribution, or degree-dependent clustering coefficient) of a given dyadic network.

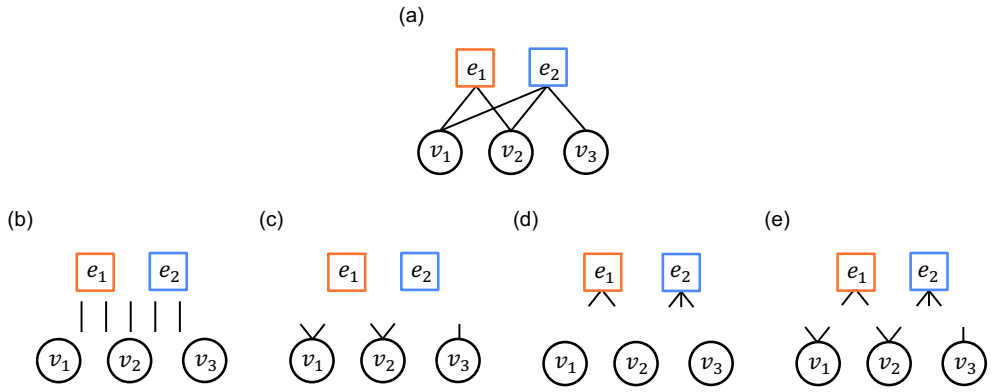


Figure 4.2: An example schematically showing the algorithm of the hyper dK-series with $d_v \in \{0,1\}$ and $d_e \in \{0,1\}$. (a) A bipartite graph. (b) $(d_v, d_e) = (0,0)$. (c) $(d_v, d_e) = (1,0)$. (d) $(d_v, d_e) = (0,1)$. (e) $(d_v, d_e) = (1,1)$.

The proposed model, which we refer to as hyper dK-series, produces a bipartite graph that preserves the joint degree distributions of the node in the subgraphs of size $d_v \in \{0,1,2,2.5\}$ or less and the size distributions of the hyperedge in the subgraphs of size $d_e \in \{0,1\}$ or less in the given bipartite graph G. We list the quantity corresponding to each d_v and d_e value in Table 4.1. By definition, the hyper dK-series with $d_v = 0$ preserves the numbers of edges in G, or equivalently, the average degree of the node. The hyper dK-series with $d_v = 1$ preserves the degree of each node. With $d_e = 0$ and $d_e = 1$, the hyper dK-series similarly preserves the average size of hyperedges and the size of each hyperedge, respectively. With $d_v = 2$, it preserves the degree of each node and aims to preserve the pairwise joint degree distribution of the node. With $d_v = 2.5$, it intends to preserve the joint degree distributions of nodes in the subgraphs of size between $d_v = 2$ and $d_v = 3$. By definition, this means that the hyper dK-series preserves the degree of each node, approximately preserves the pairwise joint degree distribution of the node, and approximately preserves the degree-dependent redundancy coefficient of the node. Like the dK-series for dyadic networks [84, 148, 180], the hyper dK-series have an inclusiveness property. In other words, the hyper dK-series with given values of d_v and d_e preserve quantities that any hyper dK-series with (d'_v, d'_e), where $d'_v \leq d_v$ and $d'_e \leq d_e$, preserve.

4.3.1 $d_v \in \{0,1\}$

In this section, we describe generation of bipartite graphs using the hyper dK-series with $d_v \in \{0,1\}$ and $d_e \in \{0,1\}$. We distinguish between the original bipartite graph, denoted by $G = (V, E, \mathcal{E})$, and the bipartite graph produced by the hyper dK-series, denoted by $\tilde{G} = (V, E, \tilde{\mathcal{E}})$. We allow $\tilde{G}$ to have multiple edges between nodes and hyperedges and to have multiple connected components, which are allowed in previous studies as well [127, 178]. We define a component of a bipartite graph as any of its maximal subgraphs in which any two nodes are connected to each other by a path within the subgraph. Our algorithm of the hyper dK-series starts with a bipartite graph with N nodes, M hyperedges, and no edge.

When $(d_v, d_e) = (0,0)$, we sequentially add edges to construct $\tilde{G}$ as follows. We select a node uniformly randomly, i.e., with probability $1/N$ and a hyperedge uniformly at random, i.e., with probability $1/M$, and connect them (Fig. 4.2(b)).

We repeat this procedure $\mathcal{M}$ times. The generated bipartite graph has N nodes, M hyperedges, and $\mathcal{M}$ edges, and hence preserves the average nodal degree and the average size of the hyperedge. When $(d_v, d_e) = (1, 0)$, we first attach k_i half-edges to each node v_i (Fig. 4.2(c)). Then, we connect each of the $\mathcal{M}$ half-edges to a hyperedge chosen uniformly at random, i.e., with probability $1/M$. The case of $(d_v, d_e) = (0, 1)$ is parallel to that of $(d_v, d_e) = (1, 0)$. Specifically, we first attach s_j half-edges to each hyperedge e_j (Fig. 4.2(d)) and then connect each of the $\mathcal{M}$ half-edges to a node chosen uniformly at random, i.e., with probability $1/N$. When $(d_v, d_e) = (1, 1)$, we first attach k_i half-edges to each node v_i and s_j half-edges to each hyperedge e_j (Fig. 4.2(e)). Then, we select a free (i.e., yet available) half-edge attached to a node and a free half-edge attached to a hyperedge uniformly at random and connect them to create a hyperedge. We repeat these steps until we exhaust all the free half-edges.

The hyper dK-series with $d_v \in \{0, 1\}$ and $d_e \in \{0, 1\}$ are the same as the existing reference models for bipartite graphs. Specifically, the hyper dK-series with $(d_v, d_e) = (1, 1)$ is a standard configuration model for bipartite graphs [76, 178], which one often uses as a reference model for bipartite graphs [93, 127, 184, 185, 201, 217] and hypergraphs [50]. The hyper dK-series with $(d_v, d_e) = (0, 0)$ is the bipartite version of the Erdős-Rényi random graph [71]. The hyper dK-series with $(d_v, d_e) = (0, 1)$ and $(1, 0)$ has also been used as a reference model for bipartite graphs [202] and hypergraphs [254].

4.3.2 $d_v \in \{2, 2.5\}$

The hyper dK-series with $d_v \leq 1$ and $d_e \leq 1$ exactly preserves up to the degree of each node and the size of each hyperedge. However, it is practically difficult to construct a bipartite graph that exactly preserves the pairwise joint degree distribution of the node by starting from the empty network and adding edges. The intuitive explanation for this difficulty is as follows. Consider an edge, of which one end has already been attached to a node v with degree k. Suppose that we connect the other end of this edge to hyperedge e of size s. If $s \geq 3$, then $m(k, k')$, i.e., the number of hyperedges that a node with degree k and a node with degree k' share simultaneously changes for multiple values of k' in general. This fact makes it difficult to connect edges between nodes and hyperedges one by one while exactly preserving the node's pairwise joint degree distribution, i.e., $P(k, k')$, for all k and k'.

This problem is similar to the one for dyadic networks; it is difficult to construct dyadic networks that exactly preserve higher-order structures than the pairwise joint degree distribution of the node [84, 148, 180]. To mitigate this problem, the algorithm of the dK-series for dyadic networks uses the so-called targeting-rewiring process with the aim of preserving the pairwise joint degree distribution and the triadic relationships, i.e., the degree-dependent clustering coefficient of the node. In the targeting-rewiring process, one repeatedly rewires edges in the generated network such that the final network exactly preserves the pairwise joint degree distribution and approximately preserves the degree-dependent clustering coefficient of the input network.

We extend the targeting-rewiring process for dK-series to the case of bipartite graphs to realize the algorithm of hyper dK-series with $d_v \in \{2, 2.5\}$. We show the composition of the hyper dK-series with $d_v \in \{2, 2.5\}$, which involves the targeting-rewiring process, in Fig. 4.4. Specifically, the hyper dK-series with $d_v = 2$ starts by generating a bipartite graph using the hyper dK-series with $d'_v = 1$ and the given $d_e \in \{0, 1\}$ (see Fig. 4.4(a) and 4.4(b)). The generated network

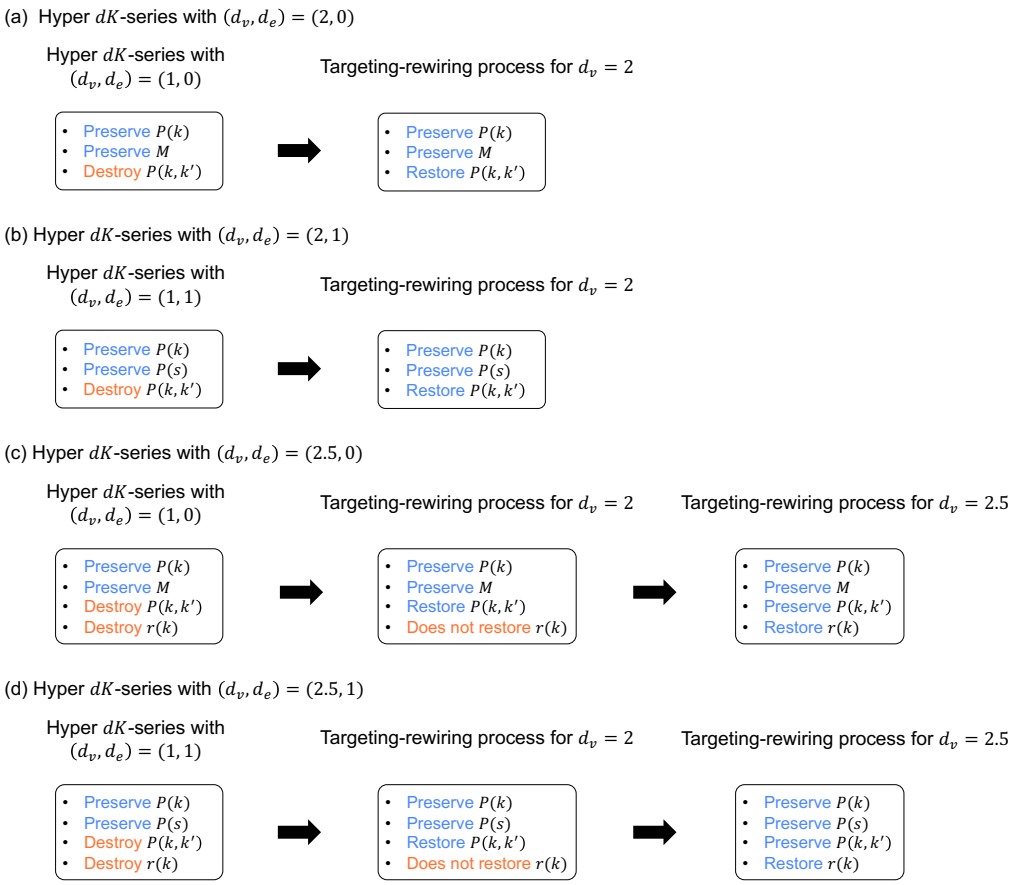


Figure 4.3: Workflow of the hyper dK-series with $d_v \in \{2, 2.5\}$ and $d_e \in \{0, 1\}$. M represents the number of hyperedges; $P(k)$ represents the degree distribution of the node; $P(s)$ represents the size distribution of the hyperedge; $P(k, k')$ represents the joint degree distribution of the node; $r(k)$ represents the degree-dependent redundancy coefficient of the node.

preserves the degree of each node and either the average size of hyperedges or the size of each hyperedge depending on whether $d_e = 0$ or $d_e = 1$, respectively. Then, we run the targeting-rewiring process for $d_v = 2$, which amounts to repeatedly rewiring edges such that the randomized hypergraph approximately restores the joint degree distribution of the original hypergraph while exactly preserving the degree of each node.

The targeting-rewiring process for $d_v = 2$ runs as follows. We first select a pair of edges (v_i, e_j) and $(v_{i'}, e_{j'})$ such that $i \neq i'$ and $j \neq j'$ uniformly at random (see Fig. 4.4(a)). Then, we replace (v_i, e_j) and $(v_{i'}, e_{j'})$ by $(v_i, e_{j'})$ and $(v_{i'}, e_j)$ if and only if a distance between the original and present pairwise joint degree distribution, denoted by D_2, decreases if we rewire the edges. Using the normalized L^1 distance, we define D_2 by

$$\begin{aligned} D_2 &= \frac{\sum_{k=1}^{M} \sum_{k'=k}^{M} |P'(k, k') - P(k, k')|}{\sum_{k=1}^{M} \sum_{k'=k}^{M} P(k, k')} \\ &= \sum_{k=1}^{M} \sum_{k'=k}^{M} \left| \frac{2m'(k, k')}{\sum_{j=1}^{M} s'_j(s'_j - 1)} - \frac{2m(k, k')}{\sum_{j=1}^{M} s_j(s_j - 1)} \right|, \end{aligned} \tag{4.5}$$

where $P'(k, k')$, $m'(k, k')$, and s'_j represent the pairwise joint degree distribution

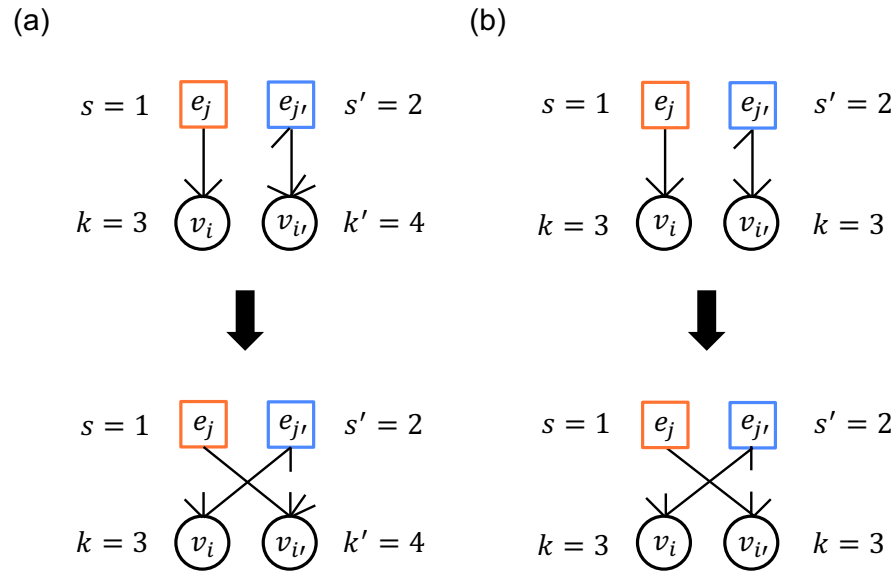


Figure 4.4: Rewiring of two edges in the targeting-rewiring process. (a) $d_v = 2$. (b) $d_v = 2.5$. In (a), we allow $k \neq k'$. In (b), we require $k = k'$. Note that the algorithm for $d_v = 2.5$ undergoes the rewiring process for $d_v = 2$ shown in (a) before one runs the rewiring process shown in (b).

of the node, the number of hyperedges that nodes with degree k and nodes with degree k' share, and the size of hyperedge e_j, respectively, for the rewired hypergraph. To derive the second line in Eq. (4.5), we have used $\sum_{k=1}^{M} \sum_{k'=k}^{M} P(k,k') = 1$. We repeat the rewiring attempts R times until D_2 becomes sufficiently small and hardly decreases by further rewiring. We set $R = 500\mathcal{M}$.

The rewiring preserves the normalization factor, $\sum_{j=1}^{M} s'_j(s'_j - 1)$, because the rewiring does not alter s'_j for any $j = 1, \ldots, M$. This property makes it easy to calculate D_2. In other words, for each edge (v, e) to be added or removed by the rewiring, it is sufficient to calculate how the number of hyperedges, $m'(k,k')$, where k and k' are the degrees of two nodes belonging to hyperedge e, changes (see Eq. (4.5)).

It is also difficult to construct bipartite graphs that exactly preserve the degree-dependent redundancy coefficient of the node, $r(k)$, over the values of k. This is because the redundancy coefficients of multiple nodes simultaneously change if one adds or removes an edge in general. Therefore, for $d_v = 2.5$, we further repeatedly rewire edges of the hypergraph generated by the hyper dK-series with $d_v = 2$ as follows. (We call this procedure targeting-rewriting for $d_v = 2.5$. See also Figs. 4.4(c) and 4.4(d).) We first select a pair of edges (v_i, e_j) and $(v_{i'}, e_{j'})$ such that $i \neq i'$, $j \neq j'$, and $k_i = k_{i'}$ uniformly at random (see Fig. 4.4(b)). Then, we replace (v_i, e_j) and $(v_{i'}, e_{j'})$ by $(v_i, e_{j'})$ and $(v_{i'}, e_j)$ if and only if the distance defined by

$$D_{2.5} = \frac{\sum_{k=1}^{M} |r'(k) - r(k)|}{\sum_{k=1}^{M} r(k)}, \tag{4.6}$$

where $r'(k)$ represents the degree-dependent redundancy coefficient of the node for the rewired hypergraph, decreases after the rewiring. We repeat the rewiring attempts $R = 500\mathcal{M}$ times.

It is easy to calculate $D_{2.5}$ upon a rewiring attempt. To explain this, we

rewrite Eq. (4.6) as

$$D_{2.5} = \frac{\sum_{k=1}^{M} \frac{1}{n(k)} |\sum_{i=1,\ k_i=k}^{N} (r'_i - r_i)|}{\sum_{k=1}^{M} \frac{1}{n(k)} \sum_{i=1,\ k_i=k}^{N} r_i}, \tag{4.7}$$

where r'_i represents the redundancy coefficient of node v_i for the rewired hypergraph. To derive Eq. (4.7), we have used the fact that the rewiring exactly preserves the degree of each node. Equation (4.7) implies that it is sufficient to only calculate the change in r'_i for the nodes that are involved in the rewiring (i.e., v_i and $v_{i'}$) and those that share at least one hyperedge with either v_i or $v_{i'}$.

The first subprocess comprising the hyper dK-series with $d_v \in \{2, 2.5\}$ is to generate a randomized hypergraph using the hyper dK-series with $d_v = 1$ (see Fig. 4.4). This process preserves the node's degree distribution and destroys the degree correlation and redundancy of the node. The second subprocess comprising the hyper dK-series with $d_v \in \{2, 2.5\}$ is the targeting-rewiring process. This process also preserves the node's degree distribution. Therefore, the entire procedure of the hyper dK-series with $d_v \in \{2, 2.5\}$ preserves the node's degree. Furthermore, the targeting-rewriting with $d_v = 2$ and $d_v = 2.5$ makes the degree correlation and redundancy, respectively, approach those of the original hypergraph, which has been lost in the course of the first subprocess. Therefore, the entire hyper dK-series with $d_v = 2$ and $d_v = 2.5$ approximately preserves the degree correlation and the redundancy, respectively.

The targeting-rewiring process for $d_v = 2.5$ also preserves the degree correlation, i.e., $P'(k, k')$ for each k and k', for the following two reasons. First, owing to the constraint that $k_i = k_{i'}$, the rewiring preserves $m'(k, k')$, i.e., the number of hyperedges that nodes with degree k and nodes with degree k' share, for any k and k'. Second, the rewiring preserves the normalization factor $\sum_{j=1}^{M} s'_j(s'_j - 1)$ as in the case of $d_v = 2$.

The targeting-rewiring process for $d_v = 2$ or 2.5 preserves the size of each hyperedge of the randomized hypergraph. However, with $(d_v, d_e) = (2, 0)$ or $(2.5, 0)$, the hyper dK-series does not preserve the size of each hyperedge of the input hypergraph. This is because we first generate a bipartite graph with $(d_v, d_e) = (1, 0)$, which destroys the size distribution of hyperedges, prior to the targeting-rewiring (see Figs. 4.4(a) and 4.4(c)).

4.3.3 An alternative algorithm for $(d_v, d_e) = (2, 1)$: Randomizing rewiring

For $(d_v, d_e) = (2, 1)$, we have an alternative to the targeting-rewiring process, which is an extension of the so-called randomizing-rewiring process in dK-series for dyadic networks [148, 180] to the case of bipartite graphs. The randomizing rewiring produces bipartite graphs that exactly preserve both nodal degree distribution and $P(k, k')$. In randomizing rewiring, the initial bipartite graph is a replica of the original bipartite graph G. Then, we select a pair of edges, (v_i, e_j) and $(v_{i'}, e_{j'})$, such that $i \neq i'$, $j \neq j'$, and $k_i = k_{i'}$ uniformly at random, and then replace (v_i, e_j) and $(v_{i'}, e_{j'})$ by $(v_i, e_{j'})$ and $(v_{i'}, e_j)$. The rewiring preserves the degree of each node, $P(k, k')$, and the size of each hyperedge. We repeat this rewiring procedure R' times, where R' is sufficiently large, and use the final result as $\tilde{G}$. We set $R' = 100\mathcal{M}$ because we have found up to our numerical efforts that the overlap of edges of G and those of the rewired hypergraph converges sufficiently before $R' = 100\mathcal{M}$.

The randomizing rewiring has an advantage over the targeting rewiring in that it exactly preserves both the degree of each node and $P(k, k')$ of the input bipartite graph; the targeting rewiring only approximately preserves $P(k, k')$.

Table 4.2: Properties of the empirical data sets. N: number of nodes, M: number of hyperedges, $\mathcal{M}$: number of edges in the corresponding bipartite graph, $\bar{k}$: average degree of the node, $\bar{s}$: average size of the hyperedge, $\bar{r}$: average redundancy coefficient of the node, $\bar{l}$: average shortest path length between nodes.

Data	N	M	$\mathcal{M}$	$\bar{k}$	$\bar{s}$	$\bar{r}$	$\bar{l}$
drug	628	816	5,688	9.06	6.97	0.70	3.53
Enron	143	1,512	4,550	31.82	3.01	0.35	2.08
primary-school	242	12,704	30,729	126.98	2.42	0.06	1.73
high-school	327	7,818	18,192	55.63	2.33	0.07	2.16

However, in contrast to the case of dyadic networks for which the randomizing rewiring is efficient [148, 180], the randomizing rewiring for the hyper dK-series has two drawbacks. First, it is only practical with $(d_v, d_e) = (2, 1)$. On one hand, although we can easily extend the randomizing rewiring to the case of $d_v \leq 1$ and $d_e \leq 1$, efficient algorithms for generating bipartite graphs exactly preserving the quantities with $d_v \leq 1$ and $d_e \leq 1$ already exist, as we described in Section 4.3.1. On the other hand, it is practically difficult to apply the randomizing rewiring in the case of $(d_v, d_e) = (2, 0), (2.5, 0)$, and $(2.5, 1)$ because a proposed random rewiring that respects the constraints imposed by the given (d_v, d_e) rarely preserves $P(k, k')$. Second, the overlap of the edges in G and those in the rewired hypergraph converges to a nonnegligibly large value with the randomizing rewiring with $(d_v, d_e) = (2, 1)$. In other words, the randomizing rewiring does not sufficiently randomly shuffle the edges of the original bipartite graph even if one carries out the rewiring many times. We show numerical evidence of this phenomenon in Section 4.6. Therefore, we use the targeting rewiring in the following analyses when $(d_v, d_e) = (2, 1)$.

4.4 Results

4.4.1 Data

In this section, we apply the hyper dK-series to four empirical hypergraphs. The NDC-classes hypergraph, which we refer to as the drug hypergraph in the following text, is a drug network constructed from the National Drug Code Directory [30]. Its nodes are class labels, such as serotonin reuptake inhibitor, and a hyperedge is a set of class labels associated with a single drug. The Enron hypergraph is an email communication network [30, 115], in which a node is an email address, and a hyperedge is a set of all recipient addresses of an email. The primary-school hypergraph is a social contact network, where nodes are individuals (i.e., students or teachers), and a hyperedge represents an event in which a set of individuals are in face-to-face contact event with each other [30,212]. The high-school hypergraph is also a social contact network, where nodes are students, and a hyperedge is a face-to-face contact event among a set of students [30,151]. We preprocessed each data set by first removing multiple hyperedges in the original hypergraph, and then by extracting the largest connected component. Table 4.2 shows properties of the largest connected component, which we use in the following analysis, for the four data sets.

4.4.2 Structural properties

For each empirical hypergraph, we compare six structural properties among the given hypergraph and hypergraphs generated by the hyper dK-series with $d_v \in \{0, 1, 2, 2.5\}$ and $d_e \in \{0, 1\}$. We also analyze an existing reference model for bipartite graphs, the B2K [37], as a benchmark. In terms of the terminology of hypergraphs, the B2K model preserves the degree of each node, the size of each hyperedge, and the number of hyperedges with size s to which nodes with degree k belong for each k and s.

Figure 4.5 compares the six structural properties between the drug hypergraph, the hyper dK-series, and the B2K model. The results for the hyper dK-series with $d_e = 0$ and various values of d_v together with those for the original drug hypergraph and the B2K model are shown in Fig. 4.5(a)–4.5(f). We make the following observations. First, Fig. 4.5(a) indicates that the hyper dK-series with $d_v \in \{1, 2, 2.5\}$ but not $d_v = 0$ exactly preserves the degree of each node (and therefore the degree distribution) of the drug hypergraph, which is expected. Second, Fig. 4.5(b) indicates that the hyper dK-series with $d_v \in \{2, 2.5\}$ but not $d_v \in \{0, 1\}$ approximately preserves the average degree of the nearest neighbors of nodes with degree k, denoted by $k_{\text{nn}}(k)$, in the input hypergraph. Because $k_{\text{nn}}(k)$ is a derivative of the pairwise joint distribution of the node's degree, $P(k, k')$, which the hyper dK-series with $d_v \geq 2$ intends to preserve, this result is also expected. The hyper dK-series with $d_v \in \{0, 1\}$ produces networks without noticeable degree correlation of the node (see Fig. 4.5(b)). Third, as expected, the hyper dK-series with $d_v = 2.5$ but not with smaller d_v values approximately preserves the degree-dependent redundancy coefficient of the node, $r(k)$, of the empirical hypergraph (see Fig. 4.5(c)). Fourth, as expected, the hyper dK-series with any $d_v \in \{0, 1, 2, 2.5\}$ and $d_e = 0$ does not preserve the distribution of the size of the hyperedge of the original hypergraph; it only preserves the average size of the hyperedge (see Fig. 4.5(d)). Fifth, the hyper dK-series with a larger value of d_v better approximates the distribution of the shortest path length between node pairs although the hyper dK-series is not designed to preserve this quantity (see Fig. 4.5(e)). Finally, we show in Fig. 4.5(f) the cumulative degree distribution of the one-mode projection, where each pair of nodes in the projected network are adjacent if they belong to at least one common hyperedge, and the multiplicity of the edge is equal to the number of hyperedges that the two nodes share [139, 189]. The hyper dK-series progressively better approximates the cumulative degree distribution of the one-mode projection when d_v is larger, whereas the results are similar between $d_v = 2$ and $d_v = 2.5$. Note that the hyper dK-series is not designed to preserve the degree distribution of the one-mode projection.

We show in Fig. 4.5(g)–4.5(l) the results for the hyper dK-series with $d_e = 1$ and various values of d_v together with those for the B2K model. The results for the empirical hypergraph and the B2K model shown in these figures are the same as those shown in Fig. 4.5(a)–4.5(f). We make the following observations. First, as expected, the results shown in Fig. 4.5(g)–4.5(i) are similar to those shown in Fig. 4.5(a)–4.5(c). In other words, the hyper dK-series with $d_v \geq 1$ preserves the degree distribution of the node, that with $d_v \geq 2$ additionally preserves $k_{\text{nn}}(k)$, and that with $d_v = 2.5$ additionally preserves $r(k)$. Second, Fig. 4.5(j) indicates that the hyper dK-series preserves the distribution of the size of hyperedge, which is because we set $d_e = 1$. Third, similar to the case of $d_e = 0$, the hyper dK-series with a larger d_v value better approximates the distribution of the shortest path length between nodes (see Fig. 4.5(k)). A comparison between Figs. 4.5(e) and 4.5(k) suggests that the approximation accuracy is not notably different between

$d_e = 0$ and $d_e = 1$. Finally, a comparison between Figs. 4.5(f) and 4.5(l) suggests that the hyper dK-series with $d_v \geq 2$ and $d_e = 1$ more accurately approximates the cumulative degree distribution of the one-mode projection than the hyper dK-series with the same d_v value and $d_e = 0$ and than that with $d_v \leq 1$ and $d_e = 1$. This is presumably because the node's degree in the one-mode projection depends not only on the degree of the node in the original hypergraph but also on the size of each hyperedge to which the node belongs.

The B2K model exactly preserves the distributions of node's degree and hyperedge's size, as expected (see Figs. 4.5(a) and 4.5(d)). However, it little preserves the node's degree correlation and the redundancy coefficient of the empirical network (see Figs. 4.5(b) and 4.5(c)). Therefore, roughly speaking, the complexity of the B2K model is somewhere between that of the hyper dK-series with $(d_v, d_e) = (1, 1)$ and that with $(d_v, d_e) = (2, 1)$. We also find that the B2K model accurately preserves the degree distribution of the one-mode projection (see Fig. 4.5(f)) although the B2K model does not intend to preserve it.

To be quantitative, we measure the distance in the distribution of each of the six quantities between the empirical hypergraph and each type of synthetic hypergraph for each data set. For the degree distribution of the node, the size distribution of the hyperedge, and the degree distribution of one-mode projection, we calculate the Kolmogorov-Smirnov distance between the cumulative distribution for the original bipartite graph and that for the generated bipartite graph. The Kolmogorov-Smirnov distance between two cumulative distributions, denoted by $F_1(x)$ and $F_2(x)$, is given by $\sup_x |F_1(x) - F_2(x)|$. For $k_{\rm nn}(k)$, $r(k)$, and the distribution of the shortest path length between nodes (which we denote by $P(\ell)$ for the shortest path length ℓ), we calculate the normalized L^1 distance between the vector corresponding to the original bipartite graph, denoted by $\boldsymbol{x} = (x_1, x_2, \ldots, x_L)$, and that corresponding to the synthetic bipartite graph, denoted by $\tilde{\boldsymbol{x}} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_L)$. Specifically, we set $x_k = k_{\rm nn}(k)$ with $k = 1, \ldots, M$, $x_k = r(k)$ with $k = 1, \ldots, M$, or $x_k = P(\ell)$ with $k = 1, \ldots, N-1$, and similar for $\tilde{\boldsymbol{x}}$. The distance between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ is defined by $\sum_{i=1}^{L} |\tilde{x}_i - x_i| / \sum_{i=1}^{L} |x_i|$. We calculate the distance average of each property over the independent 100 runs for each model. In each model, we generate an independent bipartite graph for each run.

We show the distance measurement results in Table 4.3. The following observations apply to all the data sets unless we state otherwise. First, we verify that the degree distribution of the node is the same between the empirical data and the hyper dK-series with $d_v \geq 1$ and the B2K model. Second, the hyper dK-series with $d_v = 2$ realize a considerably small distance to the empirical data in terms of $k_{\rm nn}(k)$. Third, the hyper dK-series with $d_v = 2.5$ yields a small distance to the empirical data in terms of $r(k)$. Fourth, the distribution of hyperedge's size is the same between the empirical data, any hyper dK-series with $d_e = 1$, and the B2K model. Fifth, for both $d_e = 0$ and $d_e = 1$, the hyper dK-series is more similar to the empirical data in terms of the distribution of shortest path length between nodes (i.e., $P(\ell)$) when d_v is larger. However, with the exception of primary-school hypergraph, the relative error between the hyper dK-series and the empirical hypergraph in terms of $P(\ell)$ is large (i.e., $> 30\%$) even with $(d_v, d_e) = (2.5, 1)$. Finally, the hyper dK-series with $(d_v, d_e) = (2, 1)$, $(2.5, 1)$ and the B2K model are close to the empirical data in terms of the degree distribution of the one-mode projection. All these results are consistent with those shown in Fig. 4.5. We also statistically tested how significantly the hyper dK-series changes each structural property of a given hypergraph (see Section 4.8).

To examine if the targeting rewiring introduces sufficient randomization, we

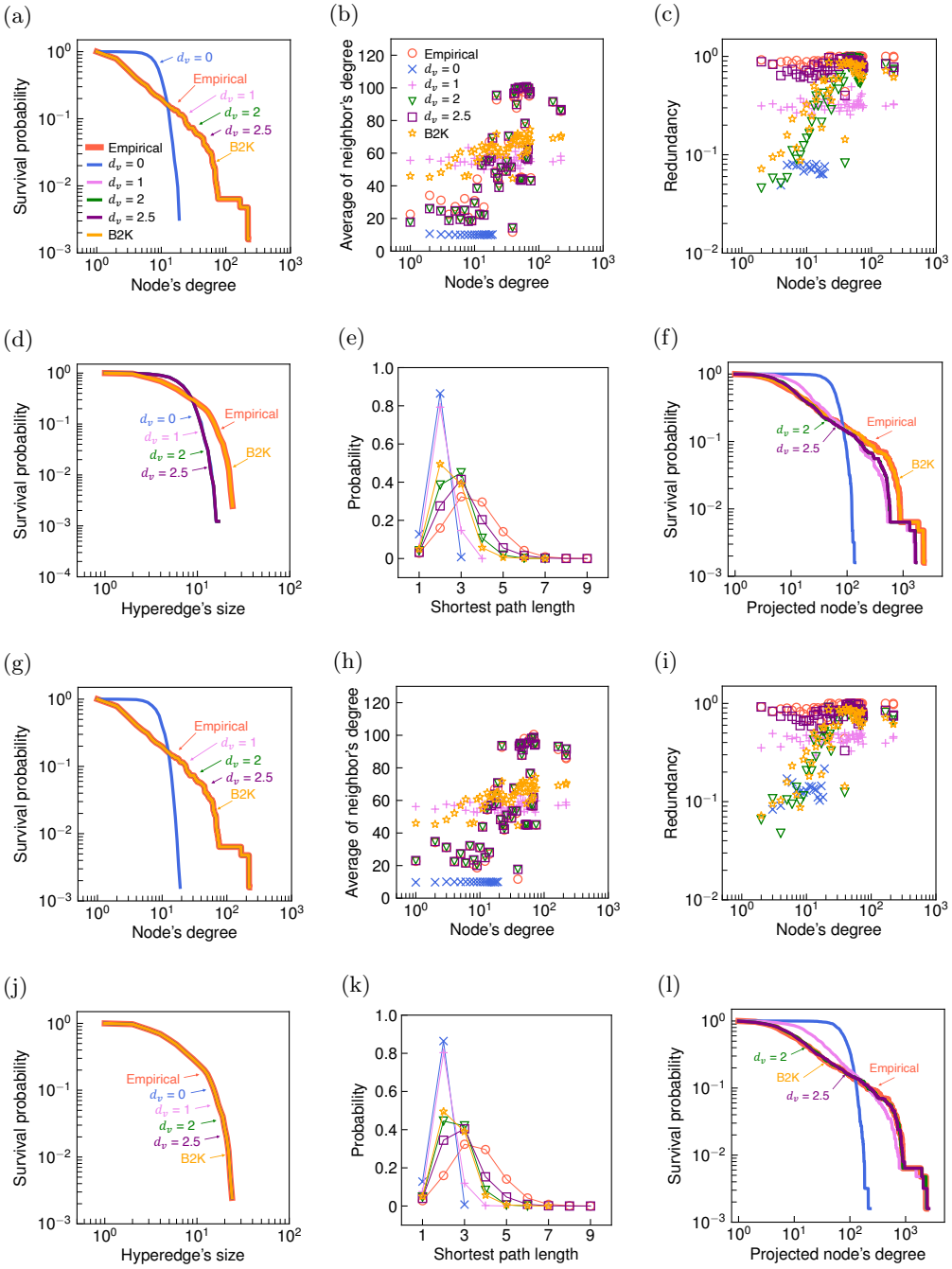


Figure 4.5: Structural properties of the drug hypergraph, the networks generated by the hyper dK-series, and the B2K model. We use the hyper dK-series with $d_e = 0$ in (a)–(f) and $d_e = 1$ in (g)–(l). Panels (a) and (g): cumulative degree distribution of the node, (b) and (h): average degree of nearest neighbors of nodes with degree k, (c) and (i): degree-dependent redundancy coefficient of the node, (d) and (j): cumulative size distribution of the hyperedge, (e) and (k): distribution of shortest path length between nodes, and (f) and (l): cumulative degree distribution of the one-mode projection. We define the shortest path length between two nodes as the smallest number of hyperedges on the path between the two nodes among all the paths. The average shortest path length is the average of the shortest average path between a pair of nodes over all pairs of nodes in the largest connected component. The largest connected component of randomized hypergraphs contains almost all nodes for all the four empirical hypergraphs (see Section 4.7 for details). We indicate the curves by the arrow and label wherever multiple curves completely or heavily overlap each other.

Table 4.3: Distance between the empirical hypergraphs and those generated by the reference models (i.e., hyper dK-series and B2K model). In the table, $P(k)$ represents the cumulative degree distribution of the node; $k_{\mathrm{nn}}(k)$ represents the average degree of the nearest neighbors of nodes with degree k; $r(k)$ represents the degree-dependent redundancy coefficient of the node; $P(s)$ represents the cumulative size distribution of the hyperedge; $P(l)$ represents the distribution of the shortest path length between nodes; $P(\breve{k})$ represents the cumulative degree distribution of the one-mode projection.

Data	Model	$P(k)$	$k_{\mathrm{nn}}(k)$	$r(k)$	$P(s)$	$P(l)$	$P(\breve{k})$
drug	$(d_v, d_e) = (0, 0)$	0.605	0.948	0.977	0.250	1.582	0.669
	$(1, 0)$	0.000	0.396	0.625	0.252	1.335	0.234
	$(2, 0)$	0.000	0.041	0.427	0.252	0.765	0.093
	$(2.5, 0)$	0.000	0.041	0.139	0.252	0.440	0.088
	$(0, 1)$	0.598	0.945	0.957	0.000	1.610	0.701
	$(1, 1)$	0.000	0.397	0.502	0.000	1.409	0.311
	$(2, 1)$	0.000	0.022	0.393	0.000	0.783	0.049
	$(2.5, 1)$	0.000	0.022	0.137	0.000	0.582	0.043
	B2K	0.000	0.326	0.394	0.000	0.850	0.027
Enron	$(d_v, d_e) = (0, 0)$	0.427	0.821	0.955	0.163	0.623	0.385
	$(1, 0)$	0.000	0.195	0.767	0.163	0.487	0.075
	$(2, 0)$	0.000	0.012	0.400	0.163	0.432	0.090
	$(2.5, 0)$	0.000	0.012	0.058	0.163	0.331	0.083
	$(0, 1)$	0.426	0.808	0.948	0.000	0.671	0.393
	$(1, 1)$	0.000	0.195	0.747	0.000	0.483	0.080
	$(2, 1)$	0.000	0.030	0.498	0.000	0.434	0.057
	$(2.5, 1)$	0.000	0.030	0.175	0.000	0.352	0.047
	B2K	0.000	0.192	0.729	0.000	0.474	0.052
primary-school	$(d_v, d_e) = (0, 0)$	0.374	0.832	0.924	0.304	0.860	0.704
	$(1, 0)$	0.000	0.088	0.547	0.305	0.705	0.372
	$(2, 0)$	0.000	0.007	0.370	0.305	0.371	0.358
	$(2.5, 0)$	0.000	0.007	0.206	0.305	0.346	0.357
	$(0, 1)$	0.377	0.834	0.970	0.000	0.537	0.390
	$(1, 1)$	0.000	0.089	0.807	0.000	0.434	0.041
	$(2, 1)$	0.000	0.014	0.563	0.000	0.244	0.031
	$(2.5, 1)$	0.000	0.014	0.112	0.000	0.035	0.032
	B2K	0.000	0.088	0.811	0.000	0.421	0.019
high-school	$(d_v, d_e) = (0, 0)$	0.308	0.698	0.908	0.326	0.534	0.622
	$(1, 0)$	0.000	0.111	0.724	0.326	0.528	0.364
	$(2, 0)$	0.000	0.009	0.434	0.326	0.511	0.321
	$(2.5, 0)$	0.000	0.009	0.030	0.326	0.505	0.322
	$(0, 1)$	0.312	0.692	0.963	0.000	0.534	0.345
	$(1, 1)$	0.000	0.112	0.894	0.000	0.515	0.073
	$(2, 1)$	0.000	0.025	0.792	0.000	0.497	0.050
	$(2.5, 1)$	0.000	0.025	0.092	0.000	0.440	0.051
	B2K	0.000	0.102	0.884	0.000	0.499	0.019

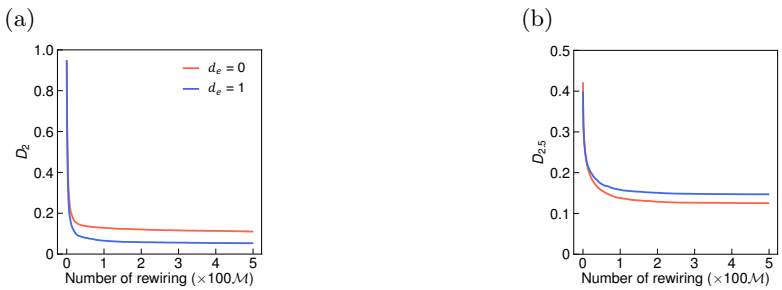


Figure 4.6: Distance between the original and synthetic hypergraphs in the targeting-rewiring process for the drug data set. (a) $d_v = 2$. (b) $d_v = 2.5$.

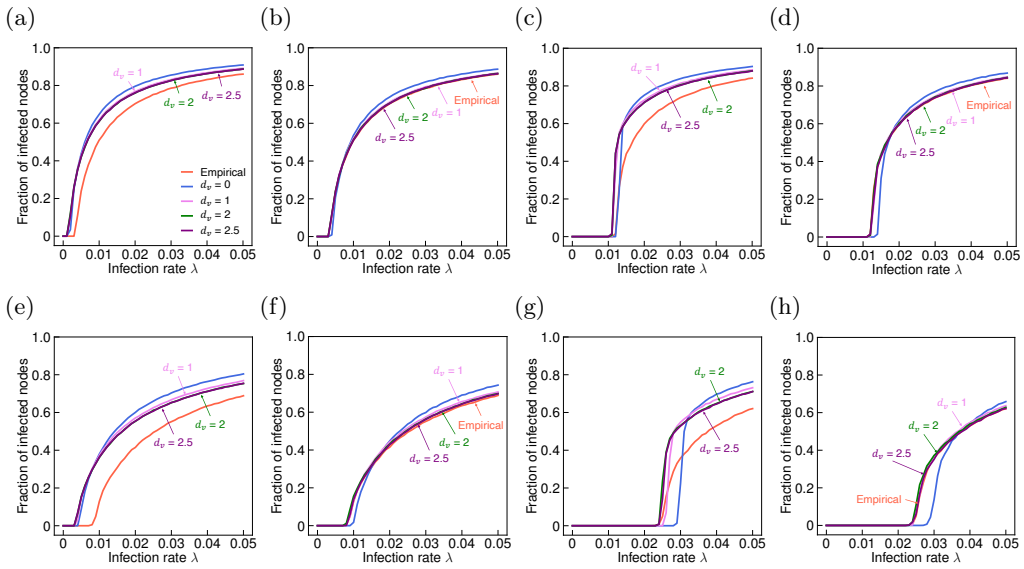


Figure 4.7: Fraction of infected nodes in the SIS model on hypergraphs. The results for primary-school data set are shown in (a)–(d), and those for the high-school data set are shown in (e)–(h). We set $(d_e, \theta) = (0, 0.1)$ in (a) and (e); $(d_e, \theta) = (1, 0.1)$ in (b) and (f); $(d_e, \theta) = (0, 0.5)$ in (c) and (g); $(d_e, \theta) = (1, 0.5)$ in (d) and (h). We indicate the curves by the arrow and label wherever multiple curves heavily overlap each other.

measure the distance measures D_2 and $D_{2.5}$, which are defined in Eqs. (4.5) and (4.6), as a function of the number of rewiring attempts, R, for the hyper dK-series with $d_v \in \{2, 2.5\}$ and $d_e \in \{0, 1\}$. The results for the drug data set are shown in Fig. 4.6. For both $d_e = 0$ and $d_e = 1$, D_2 rapidly decreased to values that are $\approx$ 15% larger than the final value in the first $\approx 100\mathcal{M}$ targeting rewiring attempts. Then, D_2 continued to decrease slowly towards the final value. Similarly, $D_{2.5}$ in the case of both $d_e = 0$ and $d_e = 1$ rapidly decreased to values that are $\approx 10\%$ larger than the final values in the first $100\mathcal{M}$ targeting rewiring attempts and then slowly decayed towards the final values. We confirmed that the trajectories of D_2 and $D_{2.5}$ were similar for the other three data sets.

4.4.3 Epidemic spreading

A primary application of the hyper dK-series is to simulations of dynamical or other processes on hypergraphs. Specifically, comparisons between the results on the original and synthetic hypergraphs generally help us to understand particular

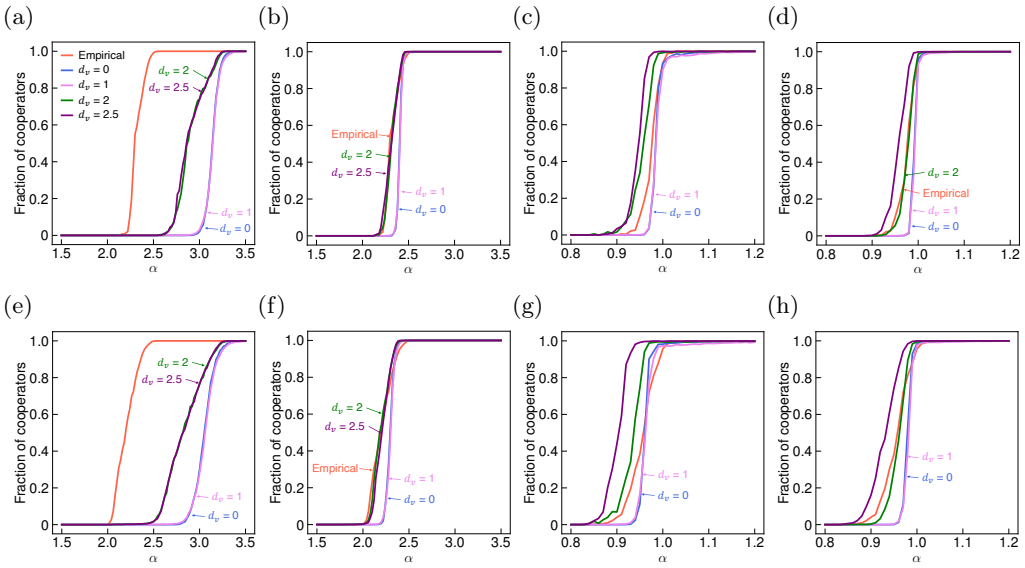


Figure 4.8: Evolution of cooperation in the public goods game on hypergraphs. Panels (a)–(d) show the fraction of cooperators for the primary-school data set, and panels (e)–(h) are for the high-school data set. We set $(d_e, \beta) = (0, 0)$ in (a) and (e); $(d_e, \beta) = (1, 0)$ in (b) and (f); $(d_e, \beta) = (0, 1)$ in (c) and (g); and $(d_e, \beta) = (1, 1)$ in (d) and (h). We indicate the curves by the arrow and label wherever multiple curves heavily overlap each other.

structural properties of the hypergraph that impact the processes on hypergraphs. For example, comparisons between a dynamical process on networks generated by the hyper dK-series with $d_v = 0$ and with $d_v = 1$ will reveal the effect of the node's degree distribution. This is because the hyper dK-series with $d_v = 0$ destroys the degree distribution of the original hypergraph, whereas that with $d_v = 1$ preserves it. Likewise, comparisons between $d_v = 1$ and $d_v = 2$ will reveal the effects of degree correlation; comparisons between $d_v = 2$ and $d_v = 2.5$ will reveal the effects of redundancy; comparisons between $d_e = 0$ and 1 will reveal the effects of the hyperedge's size distribution. We showcase the application of the hyper dK-series with epidemic spreading and evolutionary game dynamics models.

In this section, we examine a susceptible-infected-susceptible (SIS) model on hypergraphs in continuous time [63]. Each node is in either the susceptible state or the infectious state at any time t. Each infectious node recovers and becomes susceptible according to a Poisson process with rate δ. A fundamental assumption underlying the present model, which distinguishes it from other SIS models on hypergraphs [34, 108, 214], is that the contagion process is critical-mass dynamics, which generalizes a previous model [105]. Let ρ_j denote the fraction of infectious nodes in hyperedge $e_j \in E$. For each hyperedge e_j, each susceptible node in e_j becomes infected at rate λ_j if and only if $\rho_j \geq \theta$, where θ is a parameter. We set $\delta = 1$ and $\lambda_j = \lambda \log_2 |e_j|$, where λ is a parameter [63].

We assume that all the nodes are initially infectious and run the SIS model on the primary-school and high-school hypergraphs until $t = 100$. We confirmed that the fraction of infected nodes converges to an approximate stationary value before $t = 100$. For the given θ and λ values, we average the fraction of infected nodes over $95 \leq t \leq 100$ and over 100 runs. In the case of the hyper dK-series, we generate an independent bipartite graph for each run.

In Figs. 4.7(a) and 4.7(b), we set $\theta = 0.1$ and compare the fraction of infected nodes among the primary-school hypergraph and hypergraphs generated by the

corresponding hyper dK-series. We set $d_e = 0$ in Fig. 4.7(a) and $d_e = 1$ in Fig. 4.7(b). The results for the empirical hypergraph shown in Figs. 4.7(a) and 4.7(b) are the same. We make the following observations. First, the hyper dK-series with $d_e = 0$ considerably overestimates the fraction of infected nodes and underestimates the epidemic threshold for the empirical hypergraph for any d_v. Second, the fraction of infected nodes in the hyper dK-series with $d_e = 1$ is closer to that in the empirical hypergraph than with $d_e = 0$. Third, the hyper dK-series with $(d_v, d_e) = (1, 1)$, $(2, 1)$, and $(2.5, 1)$ accurately estimate the fraction of infected nodes and the epidemic threshold in the empirical hypergraph and almost to the same extent. In other words, the hyper dK-series with $(d_v, d_e) = (1, 1)$ is necessary and sufficient for reproducing the fraction of infected nodes as a function of the infection rate. These results indicate that the size of each hyperedge, or equivalently, its distribution, is a main determinant of the epidemic spreading more than are the node's local properties with $d_v > 1$, such as the degree correlation and redundancy coefficient, and mesoscopic or macroscopic structure of the hypergraph. These results qualitatively remain the same for a different threshold value, i.e., $\theta = 0.5$ (see Figs. 4.7(c) and 4.7(d)) and for the high-school hypergraph (see Figs. 4.7(e)–4.7(h)).

4.4.4 Evolutionary dynamics

Next, we compare evolutionary dynamics on the empirical hypergraphs and the hyper dK-series. We use a previously proposed model of public goods game on hypergraphs, which proceeds as follows [18]. Each node selects either to cooperate or defect in each round of evolutionary dynamics. A cooperator transfers an asset c to the public goods of hyperedge e, where $|e| \geq 2$. A defector does not contribute to the public goods. The total investment in e is $n_{\mathrm{C}}c$, where n_{C} is the number of cooperators in e. Then, one multiplies the total investment by the synergy factor R, where $R > 1$, and then equally distributes the multiplied total investment among all the nodes in e. The payoff that a cooperator and defector receives from hyperedge e is equal to $\pi_{\mathrm{C}} = Rn_{\mathrm{C}}c/|e| - c$ and $\pi_{\mathrm{D}} = Rn_{\mathrm{C}}c/|e|$, respectively. As in the previous study [18], we assume $R = \alpha|e_j|^{\beta}$, where $\alpha > 0$ and $\beta \geq 0$.

We numerically simulate the evolutionary public goods game on the given hypergraph as follows. Initially, each node is independently cooperator or defector with a probability of 0.5 each. In each round, we first uniformly randomly select a node v_i, whose strategy (i.e., cooperation or defection) may be updated, with probability $1/N$ and then select a hyperedge e_j to which v_i belongs with probability $1/k_i$ uniformly at random. We continue this selection procedure until we select a hyperedge with $|e_j| \geq 2$. We have confirmed that each node belongs to at least one hyperedge with $|e_j| \geq 2$ in all cases. Then, all the nodes that belong to e_j play the public goods game just once in each of the hyperedges to which they belong. Each node accumulates the payoffs from all the games that the node plays. Then, we divide the accumulated payoff by the number of games that the node has played. We denote by π_i the payoff of node v_i. Node v_i adopts the strategy of the node that has gained the largest payoff in hyperedge e_j, denoted by $v_{i'}$, with probability $(\pi_{i'} - \pi_i)/\Delta$. When $\beta < 1$, we set

$$\Delta = \begin{cases} \alpha\tilde{s}_{\min}^{\beta-1}(\tilde{s}_{\min} - 1) - \alpha s_{\max}^{\beta-1} + 1 & \text{if } \alpha \leq \frac{2}{\tilde{s}_{\min}^{\beta-1} + s_{\max}^{\beta-1}}, \\ \alpha\tilde{s}_{\min}^{\beta} - 1 & \text{otherwise,} \end{cases} \tag{4.8}$$

where $\tilde{s}_{\min} = \max\{s_{\min}, 2\}$, and $s_{\max}$ and $s_{\min}$ are the largest and smallest sizes

of the hyperedge, respectively. When $\beta \geq 1$, we set

$$\Delta = \begin{cases} \alpha s_{\max}^{\beta-1}(s_{\max}-1) - \alpha \tilde{s}_{\min}^{\beta-1} + 1 & \text{if } \alpha \leq \frac{2}{\tilde{s}_{\min}^{\beta-1} + s_{\max}^{\beta-1}}, \\ \alpha s_{\max}^{\beta} - 1 & \text{otherwise.} \end{cases} \tag{4.9}$$

Equations (4.8) and (4.9) guarantees that the probability $(\pi_{i'} - \pi_i)/\Delta$ is normalized (see Ref. [18] for details). If $\pi_{i'} \leq \pi_i$, node v_i does not adopt the strategy of $v_{i'}$. For the given α and β values, we measure the fraction of cooperators as the average over the (10^6+1)st and (10^6+10^3)th rounds in a single run and over 100 runs. In the case of the hyper dK-series, we generate an independent bipartite graph for each run.

In Figs. 4.8(a) and 4.8(b), we set $\beta = 0$ and compare the fraction of cooperators on the primary-school hypergraph and the hyper dK-series. We set $d_e = 0$ in Fig. 4.8(a) and $d_e = 1$ in Fig. 4.8(b). The results for the empirical hypergraph shown in Figs. 4.8(a) and 4.8(b) are the same. We make the following observations. First, at both d_e values, the node's pairwise degree correlation present in the empirical hypergraph promotes the cooperation but the node's degree distribution or the profile of the redundancy coefficient does not. Second, the fraction of cooperators in the hyper dK-series with any d_v and $d_e = 0$ is considerably smaller than that in the empirical hypergraph. In contrast, the fraction of cooperators in the hyper dK-series with $d_e = 1$ is generally close to that in the empirical hypergraph. Therefore, destroying the distribution of the hyperedge's size in the original hypergraph suppresses cooperation. In fact, the size distribution of the hyperedge is a stronger determinant of the amount of cooperation than any of the node's local properties investigated (i.e., the degree distribution, pairwise degree correlation, and redundancy coefficient).

Figures 4.8(c) and 4.8(d) show the results for $\beta = 1$. We make the following observations. First, when $d_e = 0$ (see Fig. 4.8(c)), preserving the node's degree correlation and redundancy of the original hypergraph individually enhances cooperation. However, when one destroys the degree correlation (i.e., $d_v = 0$ or 1), there is less cooperation than in the original hypergraph. Furthermore, intriguingly, the hyper dK-series with $(d_v, d_e) = (2, 0)$ and $(2.5, 0)$ realize more cooperation than on the original hypergraph, suggesting that destroying the network structure that is higher-order than the degree-correlation and redundancy promotes cooperation. Second, there is less cooperation when the distribution of the hyperedge's size is preserved (i.e., $d_e = 1$; Fig. 4.8(d)) than destroyed (i.e., $d_e = 0$; Fig. 4.8(c)). This result is opposite to that for $\beta = 0$ (see Figs. 4.8(a) and 4.8(b)). Third, similarly to the case of $d_e = 0$, the preservation of the node's degree correlation and redundancy (but not higher-order structure) of the original hypergraph individually increases cooperation in the case of $d_e = 1$. In particular, hyper dK-series with $(d_v, d_e) = (2.5, 1)$ realizes more cooperation than on the original hypergraph (see the purple line in Fig. 4.8(d)). A comparison between Figs. 4.8(c) and 4.8(d) suggests that, no matter whether the distribution of the hyperedge's size is destroyed or preserved, destroying the structure that is higher-order than the node's redundancy by randomization yields more cooperation than in the original hypergraph. All these results qualitatively remain the same for the high-school hypergraph (see Figs. 4.8(e)–4.8(h)).

The critical point $\alpha = \alpha_c(\beta)$ separating the defection and cooperation phases is analytically calculated as follows [18]:

$$\alpha_c(\beta) = \frac{1}{\sum_{s=\tilde{s}_{\min}}^{s_{\max}} \tilde{p}(s) s^{\beta-1}}, \tag{4.10}$$

where $\tilde{p}(s) = p(s)/\sum_{s=\tilde{s}_{\min}}^{s_{\max}} p(s)$, and $p(s)$ represents the fraction of hyperedges of size s. Note that it holds that $\sum_{s=\tilde{s}_{\min}}^{s_{\max}} \tilde{p}(s) = 1$. In the infinite well-mixed population, the evolutionary dynamics converge to full defection and full cooperation when $\alpha < \alpha_c(\beta)$ and $\alpha > \alpha_c(\beta)$, respectively.

When $\beta = 0$, the primary-school hypergraph yields $\alpha_c(0) \approx 2.31$. Roughly consistent with this, the fraction of cooperators on the empirical hypergraph reaches ≈ 1.0 at $\alpha \approx 2.5$ in our simulations (see the red lines in Figs. 4.8(a) and 4.8(b)). The corresponding hyper dK-series with any d_v and $d_e = 0$ leads to $\alpha_c(0) \approx 2.77$, which underestimates the threshold obtained from the numerical simulations, i.e., $\alpha \approx 3.3$ (see Fig. 4.8(a)). However, Eq. (4.10) and our numerical results are consistent in the sense that the critical point in terms of α for the hyper dK-series with $d_e = 0$ is larger than that for the empirical hypergraph. The hyper dK-series with any d_v and $d_e = 1$ has the same analytically determined threshold, $\alpha_c(0) \approx 2.31$, as the empirical hypergraph because these hypergraphs have the same distribution of the hyperedge's size. This result is also consistent with our numerical result that the fraction of cooperators reaches ≈ 1.0 at $\alpha \approx 2.5$ in the hyper dK-series with any d_v and $d_e = 1$ (see Fig. 4.8(b)). When $\beta = 1$, Eq. (4.10) predicts that $\alpha_c(1) = 1.0$ regardless of d_v and the size distribution of the hyperedge (therefore, regardless of d_e). This result is consistent with our numerical results shown in Figs. 4.8(c) and 4.8(d).

For the high-school hypergraph, we obtain $\alpha_c(0) \approx 2.23$ for the empirical hypergraph and the hyper dK-series with $d_e = 1$, $\alpha_c(0) \approx 2.75$ for the hyper dK-series with $d_e = 0$, and $\alpha_c(1) = 1.0$ for the empirical and synthetic hypergraphs. In our numerical simulations, we obtain $\alpha_c(0) \approx 2.5$ for for the empirical hypergraph (see the red lines in Figs. 4.8(e) and 4.8(f)) and the hyper dK-series with $d_e = 1$ (see Fig. 4.8(f)), $\alpha_c(0) \approx 3.3$ for the hyper dK-series with $d_e = 0$ (see Fig. 4.8(e)), and $\alpha_c(1) \approx 1.0$ for the empirical and synthetic hypergraphs (see Figs. 4.8(g) and 4.8(h)). These results are qualitatively the same as those for the primary-school hypergraph.

4.5 Conclusion

We proposed a family of reference models for hypergraphs called the hyper dK-series. The hyper dK-series preserves the local properties of nodes and hyperedges in the given hypergraph to different extents. We empirically showed that the hyper dK-series preserves the properties of nodes and hyperedges, as intended, across different hypergraph data sets. We also showcased its use as reference models in investigating epidemic spreading and evolution of cooperation on hypergraphs. Models of dynamical processes on hypergraphs, such as the epidemic spreading [34, 108, 124, 214], evolutionary dynamics [18, 42], opinion dynamics [99, 169, 198], and synchronization [64, 146, 156, 200], have been proposed. Deploying the hyper dK-series to studies of various models of dynamics is expected to better reveal how the dynamics depend on the specific structural properties of the given hypergraphs.

Up to our numerical efforts, we found that the hyper dK-series with a larger d_v value better approximates the distribution of the shortest path length between nodes for the empirical hypergraphs. However, as expected, even the hyper dK-series with the largest d_v value (i.e., $d_v = 2.5$) does not accurately approximate the distribution of the shortest path length. In particular, we found that the average shortest path length for the hypergraphs generated by the hyper dK-series with $d_v = 2.5$ is smaller than that for the empirical hypergraph for all the four data sets (e.g., the drug hypergraph has the average shortest length of

3.53, whereas the hyper dK-series has 3.03 for $(d_v, d_e) = (2.5, 0)$ and 2.77 for $(d_v, d_e) = (2.5, 1)$). The community structure is one of network structures that is higher-order than the redundancy coefficient of the node and likely increases the shortest path length between nodes. Extending the hyper dK-series to reference models that additionally preserve the community structure warrants future work. To this end, it may be useful to employ a family of stochastic block models with the community structure for bipartite graphs [23, 67, 126, 244] or hypergraphs [14, 51, 80, 113].

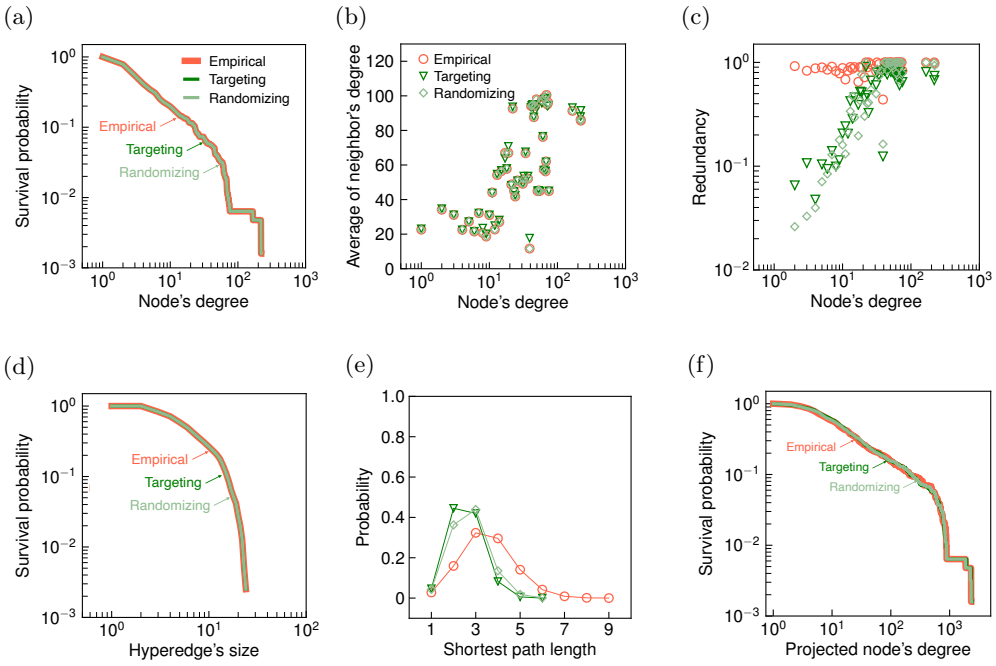


Figure 4.9: Comparison between the targeting-rewiring and randomizing-rewiring processes for the drug hypergraph. We set $(d_v, d_e) = (2, 1)$. (a) Cumulative degree distribution of the node, (b) average degree of the nearest neighbors of nodes with degree k, (c) degree-dependent redundancy coefficient of the node, (d) cumulative size distribution of the hyperedge, (e) distribution of the shortest path length between nodes, and (f) cumulative degree distribution of the one-mode projection. We indicate the curves behind other curves by the arrow and label wherever multiple curves completely or almost overlap each other.

4.6 Comparison of the targeting rewiring and randomizing rewiring for $(d_v, d_e) = (2, 1)$

In this section we compare the targeting-rewiring and randomizing-rewiring processes with $(d_v, d_e) = (2, 1)$. We show the distributions of the six quantities for the two rewiring processes for the drug hypergraph in Fig. 4.9. Both rewiring processes exactly preserve the degree distribution of the node and the size distribution of the hyperedge of the original bipartite graph (see Figs. 4.9(a) and 4.9(d)). The randomizing-rewiring process exactly preserves $k_{\rm nn}(k)$, whereas the targeting-rewiring process only approximately preserves it (see Fig. 4.9(b)). The two rewiring methods produce similar networks in terms of the degree-dependent redundancy coefficient, the distribution of the shortest path length between nodes, and the degree distribution of the one-mode projection, as shown in Figs. 4.9(c), 4.9(e), and 4.9(f), respectively.

We also compare the two rewiring processes in terms of the overlap of the edges of the empirical hypergraph and those of the synthetic hypergraphs. Figure 4.10(a) shows the Jaccard index between sets of edges in the drug hypergraph and the hypergraph generated by the randomizing rewiring as a function of the number of rewiring attempts. The figure indicates that the Jaccard index steadily decreases as the randomizing rewiring proceeds. However, it plateaus at ≈ 0.32, which implies that a set of edges in the synthetic bipartite graph is not sufficiently shuffled due to the constraints that each edge rewiring step has to preserve $P(k, k')$ in addition to the degree of each node. The Jaccard index similarly plateaus at

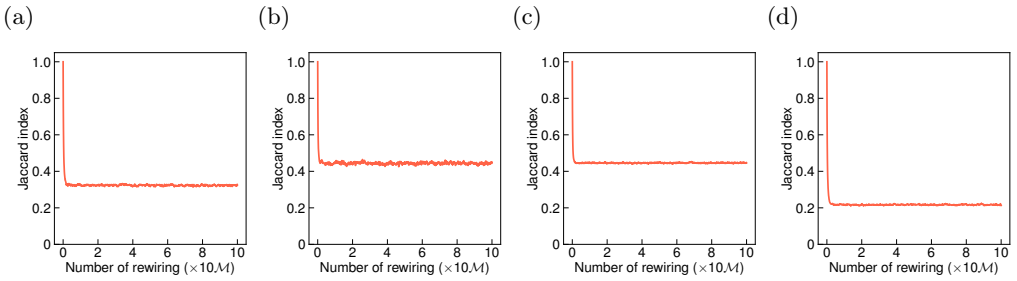


Figure 4.10: The Jaccard index between a set of edges of the empirical hypergraph and that of the hypergraph generated under the randomizing rewiring. We set $(d_v, d_e) = (2, 1)$. (a) Drug, (b) Enron, (c) primary-school, and (d) high-school. The Jaccard index between the sets of edges is given by $|\mathcal{E} \cap \tilde{\mathcal{E}}|/|\mathcal{E} \cup \tilde{\mathcal{E}}|$, where $\mathcal{E}$ and $\tilde{\mathcal{E}}$ are the set of edges in the original and synthetic hypergraphs, respectively. In calculating the Jaccard index, we removed multiplicity of edges in $\tilde{\mathcal{E}}$.

≈ 0.45, ≈ 0.45, and ≈ 0.21 for the Enron, primary-school, and high-school hypergraphs, respectively (see Figs. 4.10(b), 4.10(c), and 4.10(d), respectively). In contrast, the Jaccard index is ≈ 0.036, ≈ 0.016, ≈ 0.006, ≈ 0.005 under the targeting rewiring for the drug, Enron, primary-school, and high-school hypergraphs, respectively. Therefore, we conclude that the randomizing rewiring does not sufficiently shuffle the edges of the input hypergraph.

4.7 Size of the largest connected component of hypergraphs generated by hyper dK-series

We measured how the size (i.e., number of nodes) of the largest connected component of the empirical hypergraphs changes by randomization using the hyper dK-series. We show in Table 4.4 the size of the largest connected component of hypergraphs the hyper dK-series generates, divided by that of the original hypergraph. The table indicates that we barely lose nodes in the largest connected component by the randomization.

4.8 Statistical test for the structural properties of hypergraphs generated by hyper dK-series

In this section, we statistically test whether the hyper dK-series changes each structural property of a given hypergraph. Consider a combination of any of the four empirical hypergraphs, any (d_v, d_e) pair, and any of the six structural properties. To carry out a t-test, we first generate 100 pairs of independent hypergraphs using the hyper dK-series. Second, we measure the distance between the two hypergraphs in each pair in terms of the distance measure for the selected structural property. We denote by μ^{rand} and σ^{rand} the mean and standard deviation, respectively, of the distance calculated on the basis of the 100 pairs of randomized hypergraphs. Third, we generate another 100 hypergraphs using the hyper dK-series with the same (d_v, d_e). Fourth, we measure the distance between the empirical hypergraph and each of the 100 hypergraphs in terms of the selected structural property. We denote by μ^{emp} and σ^{emp} the mean and standard deviation, respectively, of the distance between a randomized hypergraph and the empirical hypergraph calculated on the basis of the 100 pairs. Finally,

we calculate the effect size for the t-test, called the Cohen's d [55], as

$$d = \frac{\mu^{\text{emp}} - \mu^{\text{rand}}}{\sqrt{\frac{(\sigma^{\text{emp}})^2 + (\sigma^{\text{rand}})^2}{2}}}. \tag{S1}$$

We define $d = 0$ if both $\mu^{\text{emp}} - \mu^{\text{rand}}$ and $(\sigma^{\text{emp}})^2 + (\sigma^{\text{rand}})^2$ are equal to zero. We regard the effect size to be very small ($d = \pm 0.01$), small ($d = \pm 0.2$), medium ($d = \pm 0.5$), large ($d = \pm 0.8$), very large ($d = \pm 1.2$), and huge ($d = \pm 2.0$) [55,203].

Table 4.5 shows μ^{rand}, σ^{rand}, μ^{emp}, σ^{emp}, and Cohen's d for the cumulative degree distribution of the node. The effect size is huge when $(d_v, d_e) = (0, 0)$ and $(0, 1)$ for all the four empirical hypergraphs because the hyper dK-series with $(d_v, d_e) = (0, 0)$ and $(0, 1)$ destroys the degree of each node. For the other (d_v, d_e) values, the effect size is zero because the hyper dK-series with $d_v \in \{1, 2, 2.5\}$ exactly preserves the degree of each node.

Table 4.6 shows the results for the average degree of the nearest neighbors of nodes with degree k. The effect size is huge when d_v is 0 or 1 because the hyper dK-series with these d_v values destroys the degree correlation. When d_v is 2 or 2.5, the hyper dK-series intends to preserve the degree correlation of the node. However, Table 4.6 indicates that the effect size ranges from medium to huge values, depending on the empirical network and the d_e value. This is because the σ^{rand} and σ^{emp} are small. Nevertheless, the Cohen's d values in these cases are much smaller than those for $d_v = 0$ and 1.

Table 4.7 shows the results for the degree-dependent redundancy coefficient of the node. The effect size is huge when d_v is 0, 1, or 2 because the hyper dK-series with these d_v values destroys the redundancy of the node of a given hypergraph. When d_v is 2.5, the hyper dK-series intends to preserve the redundancy of the node. However, the effect size is huge for all the four hypergraphs and d_e values. As in the case of the degree correlation, this is because σ^{rand} and σ^{emp} are small. Nevertheless, similar to Table 4.6, d is much smaller with $d_v = 2.5$ than with $d_v \leq 2$.

Table 4.8 shows the results for the cumulative size distribution of the hyperedge. The effect size is huge when $d_e = 0$ for all the four empirical hypergraphs because the hyper dK-series with $d_e = 0$ destroys the size of each hyperedge. When $d_e = 1$, the effect size is equal to zero because the hyper dK-series with $d_e = 1$ exactly preserves the size of each hyperedge.

Tables 4.9 and 4.10 show the results for the distribution of the shortest path length between nodes and those for the cumulative degree distribution of the one-mode projection, respectively. For both properties, the effect size is huge in almost all cases. This result is consistent with the fact that the hyper dK-series does not intend to preserve these two properties. However, the d value is considerably smaller when d_v or d_e is larger in most cases.

Table 4.4: Relative size of the largest connected component of hypergraphs generated by the hyper dK-series. We show the mean ± standard deviation (SD) for each parameter set. We calculated the mean and the standard deviation on the basis of 100 randomized hypergraphs.

Data	(d_v, d_e)	Mean ± SD
drug	$(0,0)$	1.000 ± 0.000
	$(1,0)$	0.999 ± 0.001
	$(2,0)$	0.984 ± 0.014
	$(2.5,0)$	0.980 ± 0.015
	$(0,1)$	0.999 ± 0.001
	$(1,1)$	0.999 ± 0.001
	$(2,1)$	0.974 ± 0.009
	$(2.5,1)$	0.952 ± 0.014
Enron	$(0,0)$	1.000 ± 0.000
	$(1,0)$	0.999 ± 0.001
	$(2,0)$	1.000 ± 0.000
	$(2.5,0)$	0.999 ± 0.001
	$(0,1)$	1.000 ± 0.000
	$(1,1)$	1.000 ± 0.000
	$(2,1)$	0.999 ± 0.001
	$(2.5,1)$	0.999 ± 0.001
primary-school	$(0,0)$	1.000 ± 0.000
	$(1,0)$	1.000 ± 0.000
	$(2,0)$	1.000 ± 0.000
	$(2.5,0)$	1.000 ± 0.000
	$(0,1)$	1.000 ± 0.000
	$(1,1)$	1.000 ± 0.000
	$(2,1)$	1.000 ± 0.000
	$(2.5,1)$	1.000 ± 0.000
high-school	$(0,0)$	1.000 ± 0.000
	$(1,0)$	0.999 ± 0.001
	$(2,0)$	1.000 ± 0.000
	$(2.5,0)$	1.000 ± 0.000
	$(0,1)$	1.000 ± 0.000
	$(1,1)$	1.000 ± 0.000
	$(2,1)$	1.000 ± 0.000
	$(2.5,1)$	1.000 ± 0.000

Table 4.5: Effect size for the cumulative degree distribution of the node.

Data	(d_v, d_e)	μ^{rand}	σ^{rand}	μ^{emp}	σ^{emp}	Cohen's d
drug	$(0,0)$	0.028	0.010	0.603	0.008	62.88
	$(1,0)$	0.000	0.000	0.000	0.000	0.000
	$(2,0)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,0)$	0.000	0.000	0.000	0.000	0.000
	$(0,1)$	0.027	0.008	0.602	0.008	69.04
	$(1,1)$	0.000	0.000	0.000	0.000	0.000
	$(2,1)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,1)$	0.000	0.000	0.000	0.000	0.000
Enron	$(0,0)$	0.062	0.018	0.421	0.019	19.58
	$(1,0)$	0.000	0.000	0.000	0.000	0.000
	$(2,0)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,0)$	0.000	0.000	0.000	0.000	0.000
	$(0,1)$	0.062	0.020	0.421	0.016	19.98
	$(1,1)$	0.000	0.000	0.000	0.000	0.000
	$(2,1)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,1)$	0.000	0.000	0.000	0.000	0.000
primary-school	$(0,0)$	0.053	0.014	0.374	0.009	27.06
	$(1,0)$	0.000	0.000	0.000	0.000	0.000
	$(2,0)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,0)$	0.000	0.000	0.000	0.000	0.000
	$(0,1)$	0.052	0.013	0.373	0.009	27.86
	$(1,1)$	0.000	0.000	0.000	0.000	0.000
	$(2,1)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,1)$	0.000	0.000	0.000	0.000	0.000
high-school	$(0,0)$	0.045	0.011	0.312	0.010	25.51
	$(1,0)$	0.000	0.000	0.000	0.000	0.000
	$(2,0)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,0)$	0.000	0.000	0.000	0.000	0.000
	$(0,1)$	0.042	0.011	0.311	0.010	25.69
	$(1,1)$	0.000	0.000	0.000	0.000	0.000
	$(2,1)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,1)$	0.000	0.000	0.000	0.000	0.000

Table 4.6: Effect size for the average degree of the nearest neighbors of nodes with degree k.

Data	(d_v, d_e)	μ^{rand}	σ^{rand}	μ^{emp}	σ^{emp}	Cohen's d
drug	$(0,0)$	0.146	0.065	0.946	0.004	17.35
	$(1,0)$	0.058	0.007	0.395	0.008	46.99
	$(2,0)$	0.015	0.002	0.042	0.003	10.05
	$(2.5,0)$	0.015	0.002	0.042	0.003	10.05
	$(0,1)$	0.142	0.060	0.946	0.004	18.80
	$(1,1)$	0.049	0.006	0.394	0.006	58.56
	$(2,1)$	0.013	0.002	0.022	0.002	3.644
	$(2.5,1)$	0.013	0.002	0.022	0.002	3.644
Enron	$(0,0)$	0.246	0.075	0.794	0.018	10.07
	$(1,0)$	0.052	0.005	0.194	0.007	22.52
	$(2,0)$	0.015	0.002	0.013	0.002	-1.087
	$(2.5,0)$	0.015	0.002	0.013	0.002	-1.087
	$(0,1)$	0.248	0.081	0.795	0.019	9.300
	$(1,1)$	0.051	0.005	0.194	0.006	25.25
	$(2,1)$	0.028	0.004	0.029	0.003	0.425
	$(2.5,1)$	0.028	0.004	0.029	0.003	0.425
primary-school	$(0,0)$	0.257	0.049	0.838	0.017	15.99
	$(1,0)$	0.021	0.001	0.089	0.002	41.77
	$(2,0)$	0.006	0.001	0.007	0.001	0.844
	$(2.5,0)$	0.006	0.001	0.007	0.001	0.844
	$(0,1)$	0.259	0.058	0.840	0.017	13.56
	$(1,1)$	0.026	0.002	0.089	0.002	32.88
	$(2,1)$	0.010	0.001	0.014	0.001	4.733
	$(2.5,1)$	0.010	0.001	0.014	0.001	4.733
high-school	$(0,0)$	0.209	0.051	0.710	0.014	13.38
	$(1,0)$	0.034	0.004	0.114	0.003	21.34
	$(2,0)$	0.011	0.002	0.010	0.002	-0.565
	$(2.5,0)$	0.011	0.002	0.010	0.002	-0.565
	$(0,1)$	0.211	0.061	0.708	0.015	11.23
	$(1,1)$	0.042	0.005	0.114	0.004	16.33
	$(2,1)$	0.020	0.003	0.025	0.002	2.007
	$(2.5,1)$	0.020	0.003	0.025	0.002	2.007

Table 4.7: Effect size for the degree-dependent redundancy coefficient of the node.

Data	(d_v, d_e)	μ^{rand}	σ^{rand}	μ^{emp}	σ^{emp}	Cohen's d
drug	$(0,0)$	0.362	0.154	0.975	0.004	5.642
	$(1,0)$	0.129	0.013	0.638	0.008	47.62
	$(2,0)$	0.049	0.006	0.430	0.005	68.89
	$(2.5,0)$	0.042	0.005	0.136	0.006	16.30
	$(0,1)$	0.328	0.109	0.956	0.006	8.113
	$(1,1)$	0.124	0.013	0.508	0.008	36.74
	$(2,1)$	0.063	0.008	0.394	0.006	47.09
	$(2.5,1)$	0.050	0.006	0.135	0.006	14.77
Enron	$(0,0)$	0.379	0.082	0.949	0.005	9.835
	$(1,0)$	0.278	0.045	0.765	0.011	14.88
	$(2,0)$	0.104	0.027	0.418	0.014	14.81
	$(2.5,0)$	0.045	0.011	0.058	0.011	1.186
	$(0,1)$	0.420	0.082	0.943	0.006	9.013
	$(1,1)$	0.323	0.049	0.752	0.012	11.99
	$(2,1)$	0.158	0.027	0.499	0.013	16.28
	$(2.5,1)$	0.093	0.017	0.148	0.015	3.530
primary-school	$(0,0)$	0.324	0.047	0.927	0.008	17.83
	$(1,0)$	0.147	0.009	0.547	0.006	54.16
	$(2,0)$	0.058	0.005	0.371	0.008	46.30
	$(2.5,0)$	0.034	0.004	0.206	0.007	29.40
	$(0,1)$	0.326	0.054	0.970	0.003	16.91
	$(1,1)$	0.150	0.012	0.808	0.002	78.27
	$(2,1)$	0.106	0.008	0.564	0.004	68.08
	$(2.5,1)$	0.058	0.005	0.115	0.005	11.72
high-school	$(0,0)$	0.326	0.048	0.910	0.005	17.26
	$(1,0)$	0.304	0.079	0.741	0.031	7.296
	$(2,0)$	0.129	0.019	0.409	0.015	16.32
	$(2.5,0)$	0.013	0.005	0.030	0.005	3.792
	$(0,1)$	0.359	0.059	0.965	0.002	14.57
	$(1,1)$	0.393	0.228	0.895	0.019	3.103
	$(2,1)$	0.225	0.034	0.761	0.008	22.00
	$(2.5,1)$	0.044	0.007	0.064	0.008	2.697

Table 4.8: Effect size for the cumulative size distribution of the hyperedge.

Data	(d_v, d_e)	μ^{rand}	σ^{rand}	μ^{emp}	σ^{emp}	Cohen's d
drug	$(0,0)$	0.023	0.007	0.249	0.011	24.41
	$(1,0)$	0.022	0.007	0.252	0.010	26.86
	$(2,0)$	0.022	0.007	0.252	0.010	26.86
	$(2.5,0)$	0.022	0.007	0.252	0.010	26.86
	$(0,1)$	0.000	0.000	0.000	0.000	0.000
	$(1,1)$	0.000	0.000	0.000	0.000	0.000
	$(2,1)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,1)$	0.000	0.000	0.000	0.000	0.000
Enron	$(0,0)$	0.016	0.006	0.163	0.007	23.91
	$(1,0)$	0.016	0.006	0.164	0.006	24.36
	$(2,0)$	0.016	0.006	0.164	0.006	24.36
	$(2.5,0)$	0.016	0.006	0.164	0.006	24.36
	$(0,1)$	0.000	0.000	0.000	0.000	0.000
	$(1,1)$	0.000	0.000	0.000	0.000	0.000
	$(2,1)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,1)$	0.000	0.000	0.000	0.000	0.000
primary-school	$(0,0)$	0.005	0.002	0.304	0.003	124.64
	$(1,0)$	0.005	0.002	0.304	0.003	125.55
	$(2,0)$	0.005	0.002	0.304	0.003	125.55
	$(2.5,0)$	0.005	0.002	0.304	0.003	125.55
	$(0,1)$	0.000	0.000	0.000	0.000	0.000
	$(1,1)$	0.000	0.000	0.000	0.000	0.000
	$(2,1)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,1)$	0.000	0.000	0.000	0.000	0.000
high-school	$(0,0)$	0.006	0.003	0.325	0.004	96.33
	$(1,0)$	0.007	0.003	0.324	0.003	105.77
	$(2,0)$	0.007	0.003	0.324	0.003	105.77
	$(2.5,0)$	0.007	0.003	0.324	0.003	105.77
	$(0,1)$	0.000	0.000	0.000	0.000	0.000
	$(1,1)$	0.000	0.000	0.000	0.000	0.000
	$(2,1)$	0.000	0.000	0.000	0.000	0.000
	$(2.5,1)$	0.000	0.000	0.000	0.000	0.000

Table 4.9: Effect size for the distribution of the shortest path length between nodes.

Data	(d_v, d_e)	μ^{rand}	σ^{rand}	μ^{emp}	σ^{emp}	Cohen's d
drug	$(0,0)$	0.006	0.005	1.577	0.005	334.86
	$(1,0)$	0.019	0.012	1.331	0.016	92.60
	$(2,0)$	0.043	0.024	0.762	0.032	25.39
	$(2.5,0)$	0.074	0.041	0.437	0.052	7.711
	$(0,1)$	0.005	0.003	1.609	0.005	435.53
	$(1,1)$	0.018	0.013	1.416	0.019	85.59
	$(2,1)$	0.043	0.023	0.788	0.029	28.82
	$(2.5,1)$	0.077	0.036	0.590	0.045	12.61
Enron	$(0,0)$	0.012	0.009	0.607	0.010	61.19
	$(1,0)$	0.013	0.009	0.489	0.004	70.38
	$(2,0)$	0.012	0.007	0.434	0.007	60.01
	$(2.5,0)$	0.016	0.009	0.352	0.014	28.21
	$(0,1)$	0.009	0.006	0.655	0.009	82.50
	$(1,1)$	0.015	0.009	0.486	0.005	66.53
	$(2,1)$	0.013	0.008	0.438	0.008	51.94
	$(2.5,1)$	0.020	0.012	0.370	0.012	28.35
primary-school	$(0,0)$	0.005	0.004	0.860	0.005	191.98
	$(1,0)$	0.005	0.004	0.706	0.005	166.63
	$(2,0)$	0.004	0.003	0.372	0.003	124.09
	$(2.5,0)$	0.004	0.003	0.346	0.004	93.71
	$(0,1)$	0.004	0.003	0.538	0.003	163.88
	$(1,1)$	0.003	0.003	0.435	0.003	154.58
	$(2,1)$	0.003	0.002	0.244	0.003	108.59
	$(2.5,1)$	0.004	0.003	0.035	0.000	16.02
high-school	$(0,0)$	0.004	0.003	0.534	0.000	218.95
	$(1,0)$	0.006	0.003	0.529	0.002	184.54
	$(2,0)$	0.005	0.003	0.510	0.003	185.10
	$(2.5,0)$	0.006	0.003	0.504	0.003	159.05
	$(0,1)$	0.002	0.001	0.534	0.000	677.59
	$(1,1)$	0.004	0.002	0.514	0.003	228.79
	$(2,1)$	0.003	0.002	0.494	0.002	253.58
	$(2.5,1)$	0.004	0.003	0.440	0.003	156.22

Table 4.10: Effect size for the cumulative degree distribution of the one-mode projection.

Data	(d_v, d_e)	μ^{rand}	σ^{rand}	μ^{emp}	σ^{emp}	Cohen's d
drug	$(0,0)$	0.036	0.009	0.663	0.008	73.88
	$(1,0)$	0.026	0.006	0.237	0.009	27.67
	$(2,0)$	0.027	0.008	0.086	0.010	6.477
	$(2.5,0)$	0.029	0.009	0.080	0.011	5.106
	$(0,1)$	0.035	0.010	0.700	0.008	75.60
	$(1,1)$	0.031	0.009	0.313	0.012	27.07
	$(2,1)$	0.030	0.008	0.052	0.009	2.609
	$(2.5,1)$	0.032	0.009	0.045	0.009	1.532
Enron	$(0,0)$	0.087	0.025	0.379	0.011	14.88
	$(1,0)$	0.049	0.011	0.074	0.008	2.553
	$(2,0)$	0.041	0.011	0.085	0.012	3.792
	$(2.5,0)$	0.046	0.011	0.071	0.011	2.274
	$(0,1)$	0.072	0.017	0.390	0.017	18.47
	$(1,1)$	0.058	0.013	0.081	0.012	1.880
	$(2,1)$	0.044	0.010	0.057	0.010	1.297
	$(2.5,1)$	0.050	0.012	0.050	0.010	-0.070
primary-school	$(0,0)$	0.069	0.018	0.697	0.015	38.60
	$(1,0)$	0.038	0.006	0.370	0.008	44.64
	$(2,0)$	0.027	0.006	0.359	0.005	60.54
	$(2.5,0)$	0.028	0.005	0.359	0.006	58.58
	$(0,1)$	0.054	0.015	0.385	0.011	24.59
	$(1,1)$	0.029	0.005	0.044	0.004	3.198
	$(2,1)$	0.024	0.004	0.032	0.003	2.326
	$(2.5,1)$	0.025	0.004	0.032	0.004	1.738
high-school	$(0,0)$	0.056	0.018	0.628	0.014	35.84
	$(1,0)$	0.035	0.008	0.361	0.013	31.38
	$(2,0)$	0.027	0.006	0.321	0.008	41.20
	$(2.5,0)$	0.030	0.006	0.320	0.008	42.14
	$(0,1)$	0.047	0.012	0.347	0.009	28.38
	$(1,1)$	0.027	0.005	0.072	0.005	9.151
	$(2,1)$	0.024	0.004	0.050	0.004	6.595
	$(2.5,1)$	0.024	0.004	0.049	0.004	6.194

Chapter 5

Higher-Order Rich-Club Phenomenon in Collaborative Research Grant Networks

5.1 Introduction

The reliance on teamwork in scientific work has increased over the past decades [75]. Funded research projects are often collaborative among institutions, and institutions with many collaborations tend to be densely connected to each other, which is known as the rich-club phenomenon in networks of research grant collaborations [147].

In this chapter, we represent grant collaboration networks among institutions as bipartite networks to investigate the properties of grant collaborations between two or more institutions. Despite coordination cost that collaborating institutions owe, it is not uncommon that more than two institutions participate in a funded research project [12, 60, 61]. Grants with large monetary amounts often require or at least encourage inter-institutional collaboration and are sometimes a main reason for collaboration among institutions [39]. Large grant teams in terms of the number of investigators tend to be more productive [58], and collaboration with such large and productive teams tends to lead to receiving future grants [69], which may also lead to an increase in the number of collaborating institutions. These observations motivate us to investigate networks of higher-order grant collaborations among institutions.

The relationships between research funding and research productivity have been investigated for individual grants [129], investigators [29, 70, 107], institutions [38, 147, 183, 195], and geographical regions [256]. Understanding such relationships is expected to assist the government and other stakeholders to develop strategies for allocating research funds to different units for enhancing research productivity. Evidence supports positive correlations between the monetary amount of research funding received by an institution and its research productivity [38, 147, 183, 195]. On the other hand, the per-dollar productivity of an institution that receives a large amount of research funding tends to be diminishing [8,231,246,252]. Given this, in the present study we ask the following question: do institutions participating in many collaborative grants gain advantages in their per-dollar productivity when they densely collaborate with each other (i.e., they form a rich club) in research grants? We examine this question using bipartite-network representation of collaborative grants among institutions, which allows us to investigating relationships among rich clubs, research productivity, and the collaboration size.

(a)

Collaborative grant u_1		
Institution	Award number	Award amount
v_1	0000001	$100,000
v_2	0000002	$200,000

Collaborative grant u_2		
Institution	Award number	Award amount
v_1	0000003	$500,000
v_2	0000004	$200,000
v_4	0000005	$300,000

Collaborative grant u_3		
Institution	Award number	Award amount
v_2	0000006	$1,000,000
v_3	0000007	$700,000
v_4	0000008	$400,000

(b)

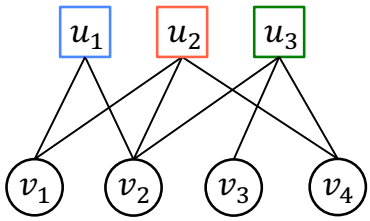


Figure 5.1: An example of three collaborative grants and the corresponding bipartite network of institutions and collaborative grants.

5.2 Methods

5.2.1 Construction of data sets

Collaborative grants

We use publicly available data on the grants administered by the National Science Foundation (NSF) [6]. We focused on the collaborative grants in each of which multiple institutions participate and each institution was responsible for a separate award. Therefore, each collaborative grant is composed of a set of linked awards each of which is separately administered by a single institution. For this type of collaborative grant, research proposals submitted by collaborating institutions must have the same project title beginning with 'Collaborative Research:' (e.g., see Ref. [3] for the latest guide posted by the NSF. We confirmed that this rule was applied at least since 1999 [1]). Therefore, we first collected the data of the awards with the project title beginning with 'Collaborative Research:' and the start date between January 1, 2000 and December 31, 2020. Second, we identified the set of institutions that received at least one such award. Third, we used the Wikipedia APIs [5] to categorize each institution into one of 48 types; see Table 5.1 for the complete list of institution types. Fourth, we obtained the data of the awards received by the institutions whose type name includes 'university', 'college', or 'school' (see Section 5.5 for the list of institution types that we focused on). Among these institutions, there are 14,081 collaborative grants each of which contains at least two awards (i.e., institutions). Fifth, for each collaborative grant, we identified the set of participating institutions, the 7-digit award number (i.e., ID) assigned to each participating institution, and the monetary amount distributed to each participating institution.

To quantify the research outputs produced under the collaborative grants, we use the Web of Science Core Collection database [7]. There are 1,082,349 papers that were published between January 1, 2000 and December 31, 2020 and

include at least one of the words 'National Science Foundation' and 'NSF' in the acknowledgment section. The fraction of papers with acknowledgment data in this data set has increased since 2008 because the Web of Science started recording the funding acknowledgment data in August 2008 [4]. For each of these papers, we extracted the 7-digit award numbers mentioned in the acknowledgement section, the number of times cited by other papers in the database, the research disciplines assigned to the paper, which is available in the data set, the publication year, and the document type. We retained the 1,066,324 papers whose document types are either 'Article', 'Review', 'Letter', 'Editorial Material', 'Meeting Abstract' or 'Proceedings Paper', as suggested in Ref. [232]. Then, for each award comprising a collaborative grant, we identified the papers that mentioned its award number in the acknowledgment section. We removed the collaborative grants with less than five published papers in the database because such collaborative grants often have extreme productivity values due to the small number of the associated papers. Then, we were left with 7,026 collaborative grants, each of which is associated with at least five of the 101,283 published papers. These collaborative grants have been awarded to 570 institutions in total.

Single-institution grants

For comparison, we also analyzed the grants that were composed of just one award given to one institution. To prepare such data, we first identified the awards of which the project title did not begin with 'Collaborative Research:' and the start date was between January 1, 2000 and December 31, 2020. There are 148,795 awards that meet these criteria and have been received by any of the 570 institutions that have participated in at least one collaborative grant. Second, for each of these awards, we identified the institution that received the award, the 7-digit award number (i.e., ID) assigned to the institution, the monetary amount of the award, and the first and last names of a principal investigator (PI) and co-PIs. Third, for each award, we identified the papers that mentioned its award number in the acknowledgment section. We removed the awards associated with less than five published papers in the Web of Science database. Then, we were left with 41,510 awards. According to the NSF's guide [3], these awards belong to one of the following three types of grant: (i) single-institution grant without co-PI, (ii) single-institution grant in which all the co-PIs are from the same institution as the PI's, and (iii) collaborative grant in which at least one co-PI from a different institution from the PI's participates and the PI's institution is responsible for the award.

We focus on the awards of types (i) and (ii) because they are genuine single-institution grants. We found 24,866 awards of type (i) among the 41,510 awards. It is not straightforward to classify the remaining 16,644 awards into types (ii) and (iii) because the affiliations of the co-PIs are not available in our data set. Therefore, we attempted to identify the awards of type (ii) as follows. First, for each co-PI in a given award, we obtain the set of candidate affiliations of the co-PI as the set of the affiliations of the authors who have the same first name initial and the same full last name as the co-PI in any of the papers associated with the award. Second, we regard that an award is of type (ii) if and only if the set of candidate affiliations of every co-PI in the award includes the institution that has received the award. We obtained 7,854 awards of type (ii) among the 16,644 awards with co-PIs. Otherwise, we regard that the award is of type (iii).

In summary, we obtained $24,866 + 7,854 = 32,720$ single-institution grants, each of which is associated with at least five of the 363,116 published papers.

These grants have been awarded to 441 institutions in total.

5.2.2 Bipartite network of institutions and collaborative grants

From the data on the collaborative grants, we construct a bipartite network that consists of a set of institutions $V = \{v_1, \ldots, v_N\}$, where N is the number of institutions, a set of collaborative grants $U = \{u_1, \ldots, u_M\}$, where M is the number of collaborative grants, and a set of edges E. An edge (v_i, u_j) exists between institution v_i and collaborative grant u_j if and only if v_i received an award in the collaborative grant u_j. A unique 7-digit award number and a unique monetary amount are associated with each edge $(v_i, u_j) \in E$. We denote by k_i the degree of v_i, i.e., the number of awards that institution v_i received from collaborative grants. We denote by s_j the degree of u_j, i.e., the number of collaborating institutions in collaborative grant u_j. We show in Fig. 5.1 a hypothetical bipartite network of four institutions and three collaborative grants. In this example, we have $V = \{v_1, v_2, v_3, v_4\}$, $U = \{u_1, u_2, u_3\}$, $E = \{(v_1, u_1), (v_1, u_2), (v_2, u_1), (v_2, u_2), (v_2, u_3), (v_3, u_3), (v_4, u_2), (v_4, u_3)\}$, $k_1 = 2$, $k_2 = 3$, $k_3 = 1$, $k_4 = 2$, $s_1 = 2$, $s_2 = 3$, and $s_3 = 3$.

5.2.3 Detection of rich clubs

A rich club of a dyadic network is defined as a subnetwork in which the nodes with the highest degrees (i.e., the nodes with the largest numbers of connected edges) are densely inter-connected to each other [57,253]. There are a few studies on rich clubs in bipartite networks. Opsahl et al. investigated rich clubs in a bipartite network of academic authors and papers [179]. They constructed a weighted unipartite network in which the weight of each edge between two authors is equal to the number of coauthored papers, which corresponds to the one-mode projection of the bipartite network to a unipartite network, and then applied a method to detect weighted rich clubs for dyadic networks. The same method was applied to detect a rich club in a bipartite bran network [59], a bipartite transportation network [74], and a bipartite technological network [53]. In the present work, we investigate rich clubs in higher-order networks of collaborative grants among institutions, which one-mode projection does not characterize. Specifically, we develop and apply a method to detect rich clubs in bipartite networks without using the one-mode projection.

We define a rich club of a given bipartite network composed of institutions and collaborative grants in which the institutions with the largest degrees densely collaborate with each other. To compute the rich club, we first calculate the rich-club coefficient, denoted by $\phi(k)$, for the original bipartite network for a given degree k. By extending the definition for dyadic networks [57, 253], we define $\phi(k)$ as the number of collaborative grants that are exclusively composed of the institutions with a degree larger than k divided by the maximum possible number of collaborative grants that are exclusively composed of some of these nodes. Formally, we define

$$\phi(k) = \frac{|U_{>k}|}{\sum_{i=2}^{N_{>k}} \binom{N_{>k}}{i}}, \tag{5.1}$$

where $U_{>k}$ is the set of collaborative grants that are exclusively composed of the institutions with a degree larger than k, and $N_{>k}$ is the number of institutions with a degree larger than k. To examine the presence of a rich club, we need

to compare $\phi(k)$ with values for a reference model [57]. Therefore, we define the normalized rich-club coefficient, denoted by $\rho(k)$, as

$$\rho(k) = \frac{\phi(k)}{\phi_{\text{rand}}(k)}, \tag{5.2}$$

where $\phi_{\text{rand}}(k)$ is the rich-club coefficient for the reference model of bipartite network. If $\rho(k)$ is sufficiently larger than 1, we say that the institutions with a degree larger than k form a rich club. For dyadic networks, a standard choice of the reference model is the configuration model, which randomizes the edges of the original network while preserving the degree of each node [57]. Here we use a counterpart of the configuration model for bipartite networks in which we randomize the edges of the original bipartite network while preserving the degree of each institution and each collaborative grant [166, 178]. We compute $\phi_{\text{rand}}(k)$ as the rich-club coefficient averaged over 10,000 randomized bipartite networks.

5.2.4 Measuring research productivity for awards, institutions, and grants

Each award in collaborative grants is associated with a monetary amount and a set of journal and conference papers supported by the award, with which we calculate the per-dollar research productivity [129] as follows. First, to compare the citation count across different publication years and research disciplines, we normalize the number of citations received by each of the 101,283 papers, which are associated with at least one collaborative grant [187, 232]. To this end, we denote by c the number of citations that a given paper z has received. We define c_0 as the number of citations that a paper that was published in the same year as z and belongs to a research discipline assigned to z has received on average. Specifically, we set $c_0 = (\sum_{d \in D(z)} \bar{c}_{d,y(z)})/|D(z)|$, where $D(z)$ is the set of the research disciplines assigned to z, $|D(z)|$ is the number of research disciplines to which z belongs, $y(z)$ is the publication year of z, and $\bar{c}_{d,y(z)}$ is the average number of citations received by the papers published in discipline d and year $y(z)$. Each paper is assigned to at least one of the 42 research disciplines [102] (see Section 5.6 for details). We define the normalized number of citations received by z as c/c_0. Then, we define the per-dollar productivity of the award given to institution v_i in collaborative grant u_j, denoted by x_{ij}, as the sum of c/c_0 over all the papers associated with the award, which we then divide by the monetary amount of the award.

We measure the productivity of collaborative funded research for a given subset of institutions, denoted by V' ($V' \subseteq V$), as follows. We first calculate the average per-dollar productivity of the awards in collaborative grants that the institutions in V' have received, denoted by $\bar{x}_{\text{inst}}(V')$. Then, we define the normalized productivity for the set of institutions V' as $\bar{x}_{\text{inst}}(V')/\bar{x}$, where $\bar{x}$ is the average per-dollar productivity of all the awards in collaborative grants. For example, when we consider the set of institutions $V' = \{v_1, v_3\}$ in a bipartite network shown in Fig. 5.1(b), we obtain $\bar{x}_{\text{inst}}(V') = (x_{11} + x_{12} + x_{33})/3$. Note that $\bar{x} = (x_{11} + x_{12} + x_{21} + x_{22} + x_{23} + x_{33} + x_{42} + x_{43})/8$. If the normalized productivity is larger than 1, the productivity of V' is higher than the average productivity of all the institutions.

We measure the productivity of a given subset of collaborative grants, denoted by U' ($U' \subseteq U$), as follows. We first calculate the average per-dollar productivity of the awards in U', denoted by $\bar{x}_{\text{grant}}(U')$. We are interested in whether institutional collaborations yield higher productivity than the average

productivity of the participating institutions. Therefore, we define the normalized productivity of U' as $\bar{x}_{\text{grant}}(U')/\bar{x}_{\text{inst}}(V'(U'))$, where $V'(U')$ is the set of institutions participating in at least one collaborative grant in U'. Note that $\bar{x}_{\text{inst}}(V'(U'))$ is the average per-dollar productivity of the awards that the institutions in $V'(U')$ have received. As an example, let us consider the set of collaborative grants $U' = \{u_1, u_2\}$ in a bipartite network shown in Fig. 5.1(b). One obtains $\bar{x}_{\text{grant}}(U') = (x_{11} + x_{21} + x_{12} + x_{22} + x_{42})/5$. Because set of institutions $V'(U')$ is $\{v_1, v_2, v_4\}$, one obtains $\bar{x}_{\text{inst}}(V'(U')) = (x_{11} + x_{12} + x_{21} + x_{22} + x_{23} + x_{42} + x_{43})/7$. If the normalized productivity is larger than 1, the productivity of the collaborative grants in U' is higher than the average productivity of the institutions participating in a collaborative grant in U'.

To quantify the productivity of single-institution grants, we adapt the above procedure for collaborative grants to the case of single-institution grants as follows. First, we construct a bipartite network composed of institutions and single-institution grants. Second, we normalize the number of citations received by each of the 363,116 papers that are associated with at least one single-institution grant by the publication year and research discipline. Then, we directly apply the definitions of productivity in the case of bipartite networks of institutions and collaborative grants to the bipartite networks of institutions and single-institution grants.

5.3 Results

5.3.1 Higher-order rich clubs in collaborative grants

We explore possibility of higher-order rich clubs in collaborative grants. We are also interested in how a rich-club phenomenon depends on the number of institutions in a collaborative grant. Therefore, we calculate the normalized rich-club coefficients for the entire bipartite network and the bipartite subnetwork induced by the collaborative grants of degree (i.e., the number of collaborating institutions), s. We consider $s \in \{2, 3, 4, 5\}$ because collaborative grants with $s \geq 6$ are rare; there are less than 100 grants for each $s \geq 6$.

Figure 5.2(a) shows the normalized rich-club coefficients for the different bipartite networks. Figure 5.2(a) indicates that the entire bipartite network shows a rich-club phenomenon (i.e., rich-club coefficient > 1.10, although this criterion is arbitrary) for the threshold of the number of awards from collaborative grants, k, approximately $100 \leq k \leq 200$. (The P-value is less than 0.005 for $1 \leq k \leq 193$ according to the Bonferroni-corrected permutation test; see Section 5.7.) The rich-club coefficient reaches the maximum value of approximately 1.21 at $k = 144$. The figure also indicates that, although the bipartite subnetwork with $s = 2$ has rich clubs that are statistically significant (see Section 5.7), the rich-club coefficient values are modest with the largest value of 1.13. In contrast, the bipartite subnetwork only composed of collaborations among $s = 3$ institutions, the subnetwork restricted to $s = 4$, and that restricted to $s = 5$ show relatively strong and persistent rich clubs across a range of k. Therefore, the institutions that receive the largest numbers of awards from either the triadic, quartic, and quintic collaborative grants tend to more densely collaborate with each other than the institutions with the largest numbers of awards from dyadic collaborative grants. Note that the normalized rich-club coefficient for the entire bipartite network (diamonds in Fig. 5.2(a)) is mostly determined by that for the subnetwork induced by the dyadic collaborative grants (crosses in Fig. 5.2(a)). This is because dyadic collaborative grants are dominant in number; they account

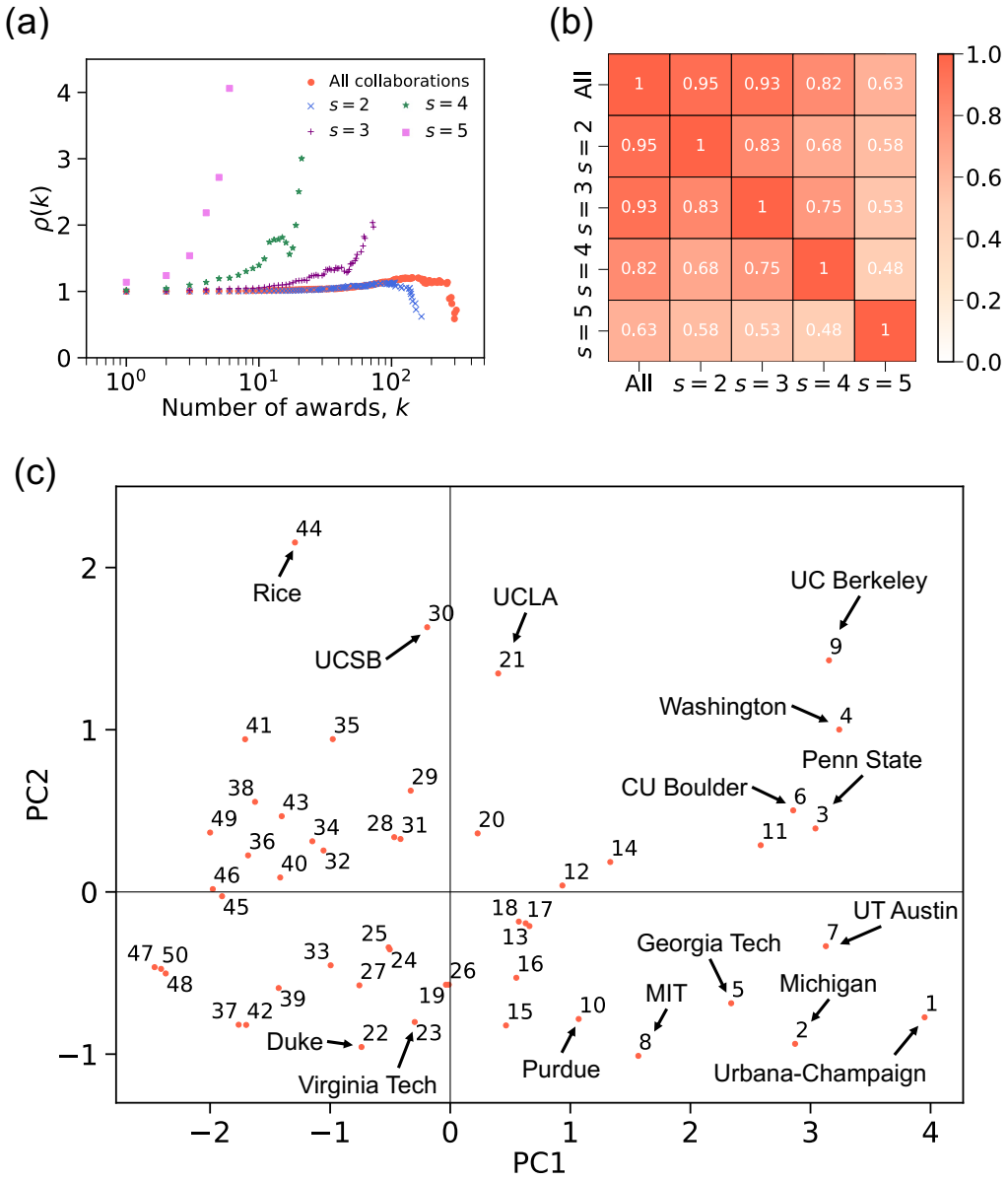


Figure 5.2: Rich-club phenomena in networks of grant collaboration. (a) Normalized rich-club coefficient $\rho(k)$ as a function of the number of awards that the institution received from collaborative grants. We measured $\rho(k)$ for the entire network (labeled "All collaborations"), the subnetwork only composed of collaboration between $s = 2$ institutions, that with $s = 3$, $s = 4$, and $s = 5$. In this figure, Fig. 5.3(b), Fig. 5.4(a)–(e), and Fig. 5.5, we omit data points for a given value of k if there are less than five instances contributing to the data point. (b) Rank correlation matrix between the different networks, where the rank is in terms of the number of awards in collaborative grants that the institution has received. We used the top 50 institutions in the entire network to calculate the rank correlation. (c) PCA result for the 50 institutions with the largest numbers of awards in the entire network. The number indicates the institution's rank in the entire network. See Section 5.8 for the names of the 50 institutions.

for approximately 67% of all the collaborative grants.

We next compare the rich clubs in the different subnetworks. We focus on the 50 institutions with the largest numbers of awards in the entire bipartite network of collaborative grants. For these institutions, we calculate the Spearman's rank correlation coefficient in terms of the number of awards between each pair of the five bipartite networks (i.e., the entire network, $s = 2$ subnetwork, $s = 3$ subnetwork, $s = 4$ subnetwork, and $s = 5$ subnetwork). We show the rank correlation for all pairs of networks in Fig. 5.2(b). We find that the entire network is the most strongly correlated with the $s = 2$ subnetwork. This result is expected because the collaborations between $s = 2$ institutions are by far the largest contributor to the entire network. Figure 5.2(b) also indicates that the correlation is larger when s is closer between two subnetworks.

This result led us to hypothesize that some institutions are good at securing collaborative grants involving fewer institutions, while other institutions are the opposite. To test this hypothesis, we classify the same 50 institutions using a principal component analysis (PCA). To run the PCA, we encode each institution into a four-dimensional vector composed of the normalized number of awards in collaborative grants with $s = 2$, $s = 3$, $s = 4$, and $s = 5$. Specifically, we scale each entry of the vector to have mean 0 and standard deviation 1. Then, we run the PCA on the normalized vectors using the scikit-learn library [186].

We show the PCA result in Fig. 5.2(c). Each data point is labeled with the institution's rank in terms of the number of awards in collaborative grants that the institution has received; see Table 5.2 for the names of the 50 institutions. The first two principal components, denoted by PC1 and PC2, explain 74.7% and 13.1% of the variance of the data, respectively. Therefore, we conclude that the two-dimensional representation of the institutions shown in Fig. 5.2(c), where the two axes correspond to PC1 and PC2, is sufficient. The eigenvector corresponding to PC1 is $(0.53, 0.54, 0.49, 0.44)$, which indicates that the number of awards from collaborative grants of any size of collaboration approximately equally contributes to PC1. As expected, institutions with a higher rank (i.e., data points labeled with a smaller number in Fig. 5.2(c)) tend to have a higher PC1 value. The eigenvector corresponding to PC2 is $(-0.25, -0.28, -0.22, 0.89)$. Therefore, the PC2 classifies the 50 institutions into those frequent in collaborations with smaller numbers of institutions (i.e., $2 \leq s \leq 4$) and those frequent in collaborative grants with $s = 5$. For example, the University of California, Berkeley ranks the 11th, 11th, 3rd, and 1st in the $s = 2$, $s = 3$, $s = 4$, and $s = 5$ subnetworks, respectively; University of Washington ranks the 6th, 2nd, 9th, and 2nd in the same four subnetworks; University of Colorado at Boulder ranks the 8th, 7th, 4th, and 4th; University of California, Los Angeles ranks the 24th, 29th, 22nd, and 7th; University of California, Santa Barbara ranks the 22nd, 38th, 42nd, and 8th; Rice University ranks the 45th, 44th, 82nd, and 6th. The latter three universities have a much higher rank in the subnetwork with $s = 5$ than that in the entire network. The behavior of institutions with a low PC2 value is the opposite. For example, University of Illinois at Urbana-Champaign ranks the 1st, 1st, 8th, and 10th in the $s = 2$, $s = 3$, $s = 4$, and $s = 5$ subnetworks, respectively; University of Michigan, Ann Arbor ranks the 3rd, 3rd, 5th, and 17th in the same four subnetworks; Massachusetts Institute of Technology ranks 5th, 9th, 12th, and 28th; Duke University ranks 18th, 18th, 34th, and 55th; Virginia Polytechnic Institute and State University ranks 32nd, 19th, 14th, and 53rd.

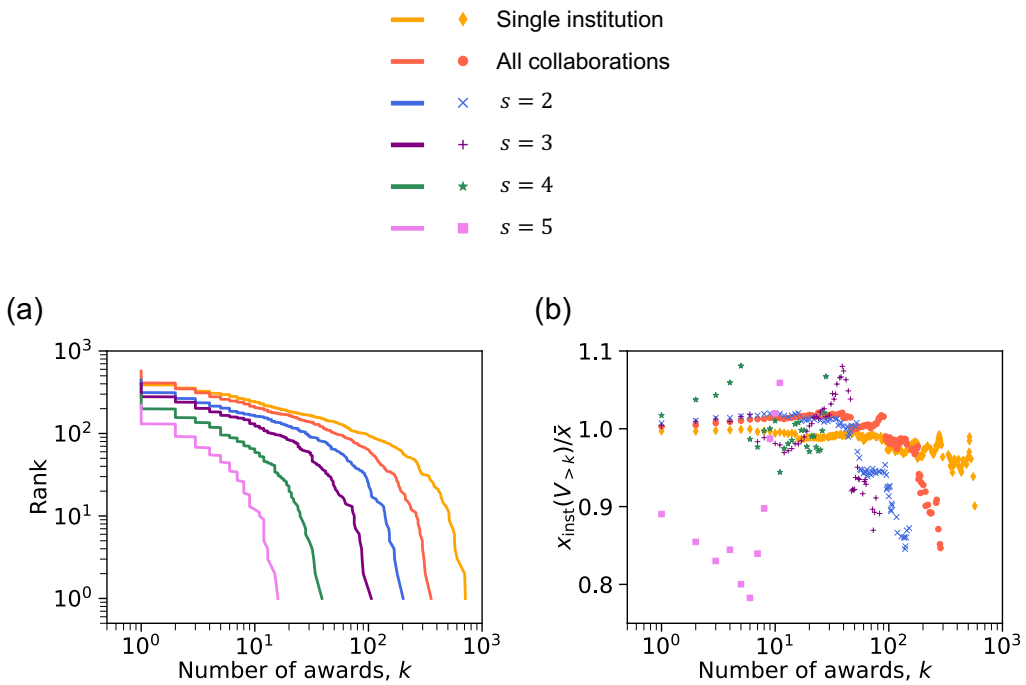


Figure 5.3: Research productivity of award-rich institutions. We analyze the single-institution grants, all the collaborative grants, and the collaborative grants with different values of s. (a) Rank plot of the institutions in terms of the number of awards. (b) Normalized productivity of the institutions with more than k awards from grants. We denote by $V_{>k}$ the set of those institutions.

5.3.2 Research productivity of the institutions with the largest numbers of collaborative grants

We now investigate research productivity of the institutions with the largest numbers of awards from collaborative grants. Note that these institutions form putative rich clubs. For comparison, we also analyze the research productivity of the institutions with the largest numbers of awards from single-institution grants. Here we analyze the data separately for all the collaborative grants, the collaborative grants comprising $s \in \{2, 3, 4, 5\}$ institutions, and single-institution grants.

First, we show the rank plot of the number of awards received by the institution, k, in Fig. 5.3(a). The figure indicates that k is skewed toward the top-ranked institutions. For example, the top 20% of institutions obtained approximately 82% of the awards in collaborative grants and approximately 79% of the awards in single-institution grants. This result is consistent with the concentration of research funding in top-ranked institutions observed in the NSF [242], the National Institutes of Health grants in the US [130, 231], and the Engineering and Physical Sciences Research Council grants in the UK [147]. We also found that the top-ranked institutions less dominate the distribution of awards in the case of collaboration with a larger number of institutions (i.e., larger s). For example, the top 20% of institutions account for approximately 79% of the awards in single-institution grants (i.e., $s = 1$), 76% for $s = 2$, 70% for $s = 3$, 60% for $s = 4$, and 53% for $s = 5$. To be further quantitative, we have calculated the coefficient of variation for the distribution of the number of awards, which is equal to 1.75, 1.67, 1.49, 1.17, and 0.95 for $s = 1$, $s = 2$, $s = 3$, $s = 4$, and $s = 5$, respectively; the Gini coefficient is 0.74, 0.72, 0.66, 0.56, and 0.46 for $s = 1$, $s = 2$, $s = 3$,

$s = 4$, and $s = 5$, respectively.

Second, we show the normalized productivity of the institutions as a function of k in Fig. 5.3(b). We find that the institutions with approximately 100 or more awards from collaborative grants tend to be less productive in the per-dollar sense than those with fewer awards. Similarly, the institutions with approximately 100 or more awards from single-institution grants tend to be less productive than those with fewer awards. This result of the diminishing per-dollar productivity at the institution level is consistent with the previous results [8, 231, 246, 252]. Figure 5.3(b) also indicates that similar diminishing productivity is present for collaborative grants of different collaboration sizes, $s \in \{2, 3, 4, 5\}$.

5.3.3 Research productivity of the collaborative grants within rich clubs

Given the results shown in Fig. 5.3, rich clubs may be detrimental to productivity because a rich club is a set of high-degree nodes, i.e., institutions with many awards. However, Fig. 5.3 does not imply that collaborative grants among rich-club institutions are not productive; we did not look into collaboration among rich-club institutions with Fig. 5.3. Therefore, we now investigate possible associations between the rich clubs in collaborative grant networks and research productivity. We first validate the productivity of the collaborative grants within rich clubs, which are exclusively composed of the institutions with the largest numbers of awards. We denote by $U_{>k,\geq p}$ the set of collaborative grants in which the fraction of the institutions with more than k awards from collaborative grants is at least p. We compare productivity of the collaborative grants, $U_{>k,\geq p}$, for different p values.

We show in Fig. 5.4 the normalized productivity of the collaborative grants in $U_{>k,\geq p}$ for different values of k and p for the entire network and the subnetwork of each collaboration size $s \in \{2, 3, 4, 5\}$. For the entire network, Fig. 5.4(a) indicates that the collaborative grants in $U_{>k,\geq p}$ with $p = 1$ and large k tend to be more productive than the expectation for the participating institutions. The maximum value of the normalized productivity is approximately 1.15 at $k = 159$. The figure also indicates that the collaborative grants in $U_{>k,\geq p}$ with $p = 1$ for given value of k tend to have a higher normalized productivity than those in $U_{>k,\geq p}$ with $0 < p < 1$. For example, at $k = 159$, the normalized productivity is 1.15, 1.10, 1.00, 0.97, and 0.98 for $p = 1$, $p = 0.8$, $p = 0.6$, $p = 0.4$, and $p = 0.2$, respectively. Figures 5.4(b)–(e) indicate that the normalized productivity for $U_{>k,\geq p}$ with $p = 1$ tends to be larger than 1 at large k values in the subnetwork with $s \in \{2, 3, 4, 5\}$. This result is qualitatively the same as that for the entire collaboration network shown in Fig. 5.4(a). Figures 5.4(b)–(e) also indicate that the normalized productivity for $U_{>k,\geq p}$ with $p = 1$ tends to be larger than that for $U_{>k,\geq p}$ with $0 < p < 1$ in each subnetwork with $s \in \{2, 3, 4, 5\}$. By definition, the normalized productivity of the single-institution grants is exactly equal to 1 for any k. Altogether, these results indicate that collaborations among the institutions with the largest numbers of collaborative grants tend to be productive, not because such institutions tend to be strong in research but because they collaborate.

To further investigate the association between rich clubs and productivity, we investigate relationships between the normalized rich-club coefficient, $\rho(k)$, and the normalized productivity of the collaborative grants that are exclusively composed of the institutions in the rich club. We denote by $U_{>k}$ the set of collaborative grants that are exclusively composed of the institutions with more

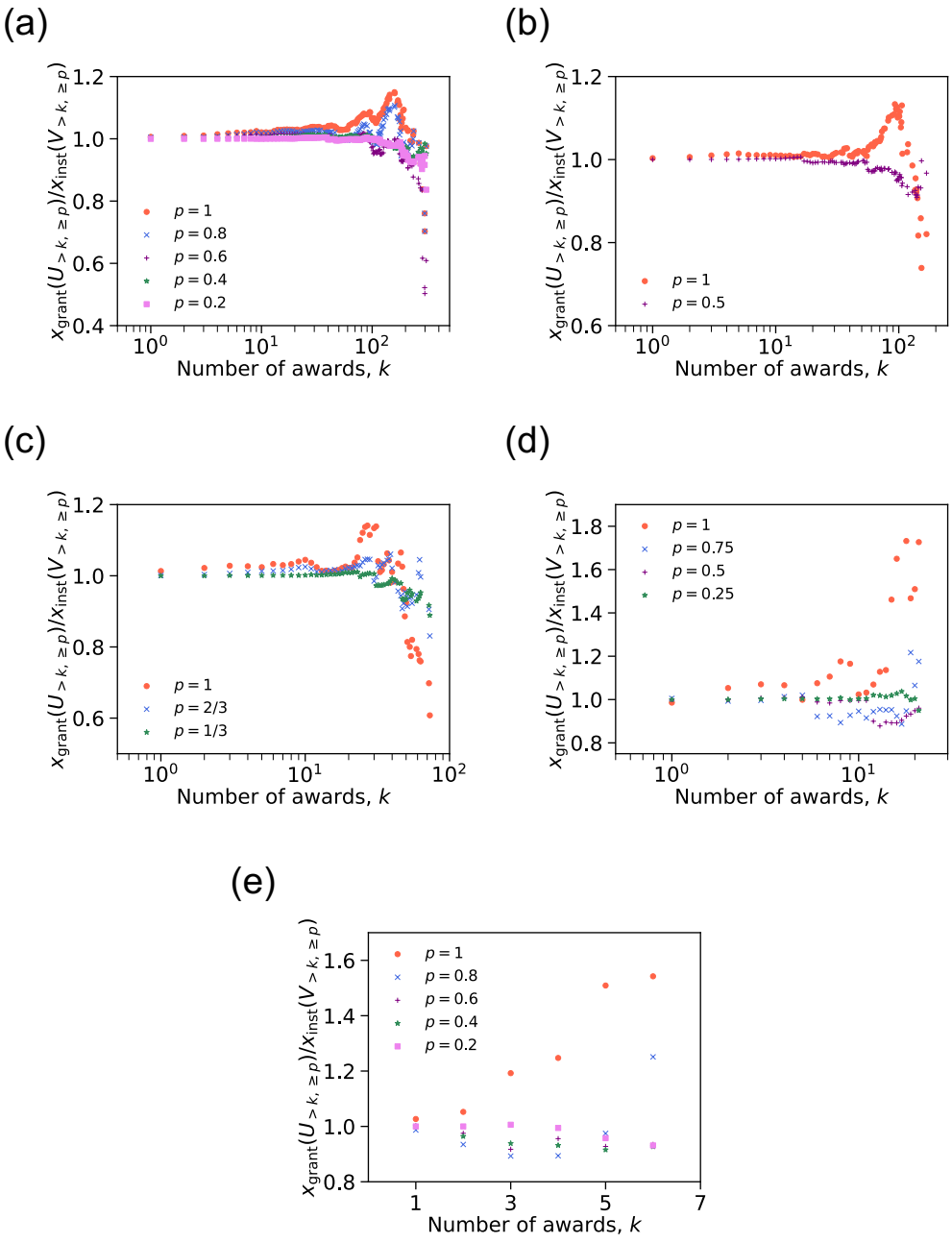


Figure 5.4: Advantage of collaborations between the award-rich institutions. We plot the normalized productivity of the collaborative grants in each of which fraction of the institutions receiving more than k awards from collaborative grants is at least p. We denote by $V_{>k,\geq p}$ the set of the institutions participating in at least one collaborative grant in $U_{>k,\geq p}$. (a) Entire network. (b) Subnetwork with $s = 2$. (c) Subnetwork with $s = 3$. (d) Subnetwork with $s = 4$. (e) Subnetwork with $s = 5$.

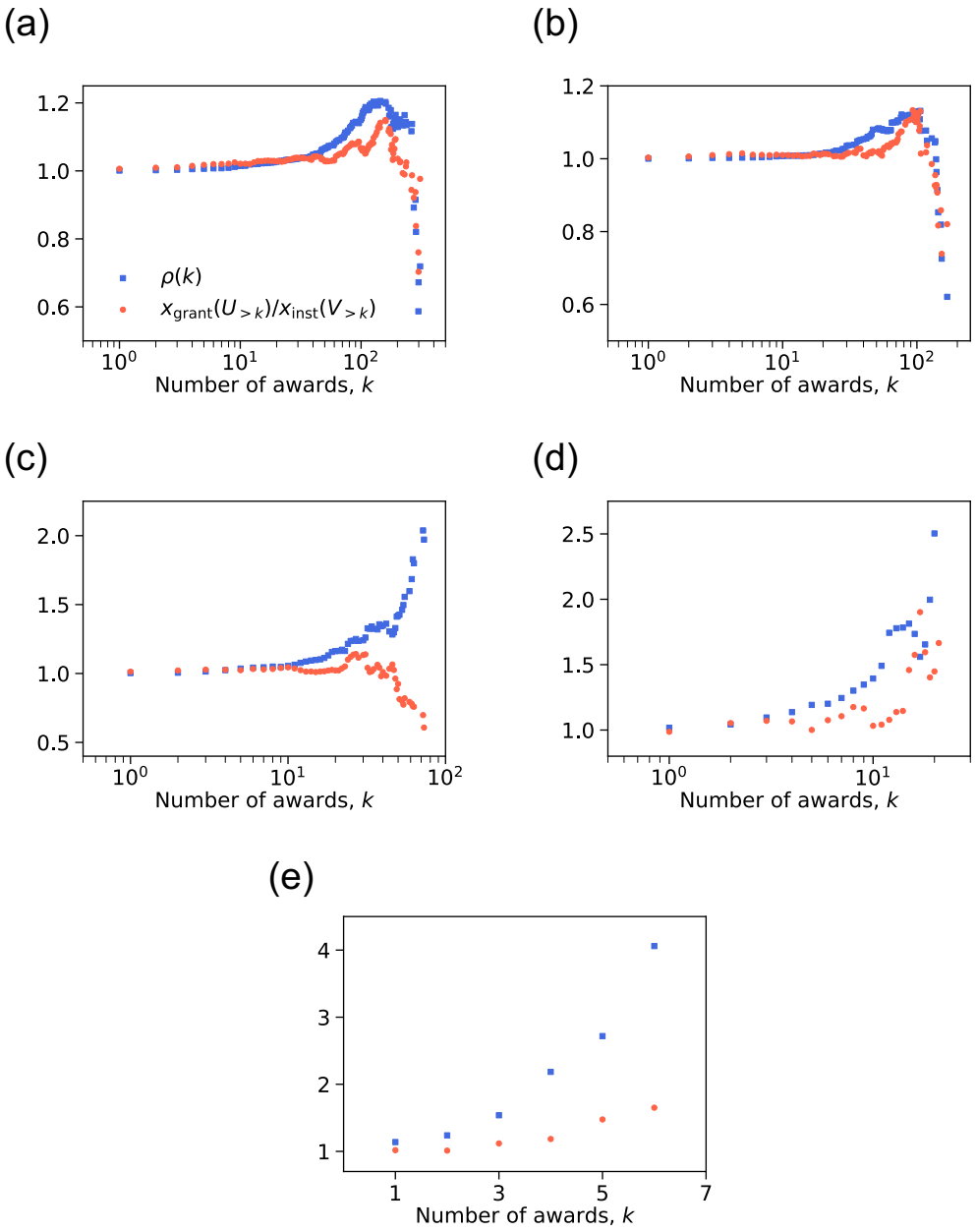


Figure 5.5: Overlay of the rich-club coefficient and productivity of the collaborative grants. Each panel shows the normalized rich-club coefficient and the normalized productivity as a function of the number of awards k that the institution has received from collaborative grants. (a) Entire network. (b) Subnetwork with $s = 2$. (c) Subnetwork with $s = 3$. (d) Subnetwork with $s = 4$. (e) Subnetwork with $s = 5$.

than k awards from collaborative grants. Note that $U_{>k}$ is equivalent to $U_{>k,\geq p}$ with $p = 1$. If $\rho(k)$ is sufficiently larger than 1, then $U_{>k}$ is the set of collaborative grants contained in the rich club. Therefore, if rich clubs are associated with high research productivity, the normalized productivity of $U_{>k}$ should be larger than 1 for the k values at which $\rho(k)$ is sufficiently larger than 1.

We show in Fig. 5.5 the plots of $\rho(k)$ and the normalized productivity of $U_{>k}$ against k, separately for the entire network and the subnetworks with $s \in \{2, 3, 4, 5\}$. The figure indicates that the normalized productivity of $U_{>k}$ tends to be larger than 1 if $\rho(k)$ is larger than 1 in the entire network (Fig. 5.5(a)). For example, $\rho(k)$ is largest at $k = 144$. The institutions with more than 144 awards collaborate with each other approximately 21% more densely than in a randomized network (i.e., $\rho(144) \approx 1.21$). The productivity of the collaborative grants in $U_{>144}$ is approximately 14% higher than expected from the average productivity of the institutions participating in a collaborative grant in $U_{>144}$. However, at $k = 299$, the rich club is absent (i.e., $\rho(299) \approx 0.67$), and the productivity of the collaborative grants in $U_{>299}$ is 30% lower than the expectation for the participating institutions. The Pearson correlation coefficient between $\rho(k)$ and the normalized productivity, where we regarded a pair of these two quantities for a value of k as a data point, is equal to $r = 0.85$ (P-value is less than 0.001). We also found a significant positive correlation between these two quantities for the subnetwork with $s = 2$ ($r = 0.89,\ P < 0.001$; see Fig. 5.5(b)), $s = 4$ ($r = 0.61,\ P < 0.005$; see Fig. 5.5(d)), and $s = 5$ ($r = 0.98,\ P < 0.001$; see Fig. 5.5(e)). For the subnetwork with $s = 3$, while we found a negative correlation ($r = -0.81,\ P < 0.001$; see Fig. 5.5(c)), the normalized productivity tends to be larger than 1 if $\rho(k)$ is larger than 1 for approximately $1 \leq k \leq 45$.

5.4 Discussion

We investigated higher-order rich-club phenomena in networks of collaborative research grants. To this end, we developed a method to detect rich clubs in bipartite networks. We observed rich clubs in both the entire bipartite network and the subnetworks induced by the collaborative grants with a given number of collaborating institutions, s, where $s \in \{2, 3, 4, 5\}$. The subnetworks with $s = 3$, 4, and 5 had stronger rich clubs than that with $s = 2$. Regarding performances of rich clubs, we found that the collaborative grants within rich clubs tend to have higher per-dollar productivity than the average productivity expected for the institutions participating in the collaboration. We emphasize that the higher productivity of rich clubs is a genuine effect of collaboration because the productivity of the single-institution grants is normalized to 1. These results support our hypothesis that collaborations among institutions in rich clubs are productive.

Our results extend the findings on the rich clubs in grant collaboration networks shown in a previous study [147] in the following two aspects. First, we found that some collaboration-rich institutions tend to densely collaborate with each other in research grants involving fewer institutions, whereas other collaboration-rich institutions tend to do so in research grants involving more institutions. One factor underlying this phenomenon may be strategies of individual institutions regarding interdisciplinary research projects. Evidence suggests that interdisciplinary research projects are less likely to attract funding in a short term [40], whereas they positively contribute to long-term funding performance [213]. This tendency may affect funding strategy of individual researchers and institutions, which may affect the distribution of the size of collaboration in terms of the number of institutions for the institution to which the researchers belong. Note that

Ma et al. employed the one-mode projection and therefore the impact of the size of collaboration is not a question that they focused on in their study. Second, the benefits of rich clubs to the per-dollar productivity seem to come from collaborations among the institutions that belong to the rich clubs. Ma et al. indicated that the rich clubs attract a large number or monetary amount of awards and tend to produce a large number of papers with high quality [147]. In contrast, our results indicate that collaborations among the institutions in rich clubs are productive in terms of the per-dollar productivity, whereas the institutions themselves with many collaborations are not particularly productive.

The generality of rich clubs in grant collaboration networks deserves further investigation. For example, the presence of rich-club phenomena and their association with productivity may be stronger in some research disciplines than in others. Our results do not guarantee the benefits of rich clubs in productivity across different disciplines. In fact, the strength of the correlation between productivity and institutional collaborations in writing papers substantially depends on research disciplines [11]. Rich clubs and their relevance to research productivity may also depend on funding agencies. The National Institute of Health financially encourages that multiple investigators with expertise in different health profession fields work together in research projects [143], which may lead to rich-club phenomena in networks in which the node is a department or institution. Moreover, higher-order rich-club phenomena in grant collaboration networks may depend on the definition of the node. In fact, Ma et al. reported that a British collaboration network among investigators in which an edge represents two investigators' co-funded research projects does not have rich clubs [147].

We did not address causality between rich clubs and research productivity. Furthermore, the higher productivity of the collaborative grants within the rich clubs may be associated with various properties of the member institutions other than the density of their collaborations, including the internationality of the faculty [149], departmental and institutional size [68], grant type [107], and funding support from industries [94], which may affect productivity. Additionally, there are other forms of dense mesoscopic structure of grant collaboration networks, most famous one of which is probably the community structure. Such other forms of dense mesoscopic structure may also affect research productivity. Examples of collaborations that may form such mesoscopic or community structures include teams composed of private universities that may be subsidized by their financial resources [12], collaborations among investigators from different departmental affiliations [158], and collaborations between universities and industries [19]. Moreover, many co-authorship networks among authors also show structures including the community structure and rich clubs [82, 179, 251]. The present method is also applicable to the investigation of higher-order rich-club phenomena in co-authorship networks. Further exploring the associations and causality between mesoscopic structure of networks involving higher-order interaction and research productivity for various types of scientific collaborations warrants future work.

5.5 Institution types

Table 5.1 shows the list of institution types that we have used.

5.6 Research disciplines

Each paper in our data set is originally assigned to at least one of the 153 research disciplines defined in the Web of Science Core Collection database. However, some disciplines contain, for example, only one paper published in a given year. Therefore, we used a previously proposed set of 42 disciplines that is a coarse graining of the original categorization [102]. See Supplementary Table S1 in Ref. [102] for the mapping from the 153 disciplines to the 42 disciplines.

5.7 Statistical test for normalized rich-club coefficients

To assess the significance of the normalized rich-club coefficient, we ran the permutation test employed in previous studies [225, 226, 235]. We denote by $\mathcal{D}$ the set of degrees, k, such that there are at least five collaborative grants in which only the institutions with more than k awards participate. For a given degree $k \in \mathcal{D}$, we calculate the rich-club coefficient of the original network, i.e., $\phi(k)$, and 10,000 values of $\phi_{\text{rand}}(k)$ using the random bipartite network model. Then, for each $k \in \mathcal{D}$, we define the P-value as the fraction of the $\phi_{\text{rand}}(k)$ values that are larger than $\phi(k)$. Our null hypothesis is that $\phi(k)$ is equal to the average of the 10,000 values of $\phi_{\text{rand}}(k)$. The alternative hypothesis is that $\phi(k)$ is larger than the average of the 10,000 values of $\phi_{\text{rand}}(k)$. We test the null hypothesis with Bonferroni-adjusted α-level of $0.005/|\mathcal{D}|$ for each degree $k \in \mathcal{D}$. We show in Fig. 5.6 the significant and nonsignificant rich-club coefficients for the entire network and the different subnetworks.

5.8 Top 50 institutions in terms of the number of collaborative grants

Table 5.2 shows the top 50 institutions with the largest number of awards from collaborative grants.

Table 5.1: Types of institutions. The institutions not found on the Wikipedia database are assigned 'N/A' type. Those with * are the types of institution that we have used in the present study.

Aquarium	*Private university
Arboretum	Public academic health science center
Garden	Public agency
Government	*Public college
Health center	*Public community college
Hospital	*Public community college district
Medical center	*Public community college system
Military academy	*Public graduate school
Museum	*Public law school
Naval academy	*Public liberal arts college
*Private art and design college	*Public liberal arts university
*Private art and design school	*Public medical school
*Private college	*Public research university
*Private community college	*Public school of optometry
*Private engineering and technology school	*Public two-year college
*Private graduate college	*Public university
*Private graduate medical school	Research agency
*Private graduate school	Research facility
*Private liberal arts college	Research institute
*Private liberal arts university	Science center
*Private medical and professional school	Space agency
*Private medical school	Think tank
*Private research university	Zoo
*Private undergraduate and graduate school	N/A

Table 5.2: The top 50 institutions in terms of the number of collaborative grants. "Public" and "Private" in the last column refer to public and private research university, respectively.

Rank	Institution	Number of awards	Institution type
1	University of Illinois at Urbana-Champaign	356	Public
2	University of Michigan, Ann Arbor	316	Public
3	Pennsylvania State University	308	Public
4	University of Washington	299	Public
5	Georgia Institute of Technology	298	Public
6	University of Colorado at Boulder	285	Public
7	University of Texas at Austin	282	Public
8	Massachusetts Institute of Technology	273	Private
9	University of California, Berkeley	273	Public
10	Purdue University	264	Public
11	Columbia University	261	Private
12	University of Wisconsin, Madison	237	Public
13	Arizona State University	235	Public
14	University of Minnesota, Twin Cities	229	Public
15	Ohio State University	221	Public
16	Cornell University	211	Private
17	Stanford University	211	Private
18	University of Arizona	210	Public
19	Carnegie Mellon University	201	Private
20	Oregon State University	193	Public
21	University of California, Los Angeles	190	Public
22	Duke University	189	Private
23	Virginia Polytechnic Institute and State University	185	Public
24	Rutgers University, New Brunswick	184	Public
25	Princeton University	183	Private
26	University of Florida	180	Public
27	Northwestern University	177	Private
28	University of Southern California	177	Private
29	University of California, Davis	176	Public
30	University of California, Santa Barbara	172	Public
31	University of California, San Diego	171	Public
32	University of Maryland, College Park	169	Public
33	Harvard University	160	Private
34	University of California, Irvine	159	Public
35	University of California, Santa Cruz	144	Public
36	Michigan State University	143	Public
37	North Carolina State University	141	Public
38	University of Massachusetts at Amherst	138	Public
39	University of North Carolina at Chapel Hill	138	Public
40	Yale University	135	Private
41	University of Pennsylvania	132	Private
42	Iowa State University	127	Public
43	Stony Brook University	127	Public
44	Rice University	126	Private

45	Boston University	123	Private
46	Johns Hopkins University	121	Private
47	University of Pittsburgh	121	Public
48	University of Virginia	120	Public
49	University of Alaska Fairbanks	118	Public
50	University of Delaware	117	Public

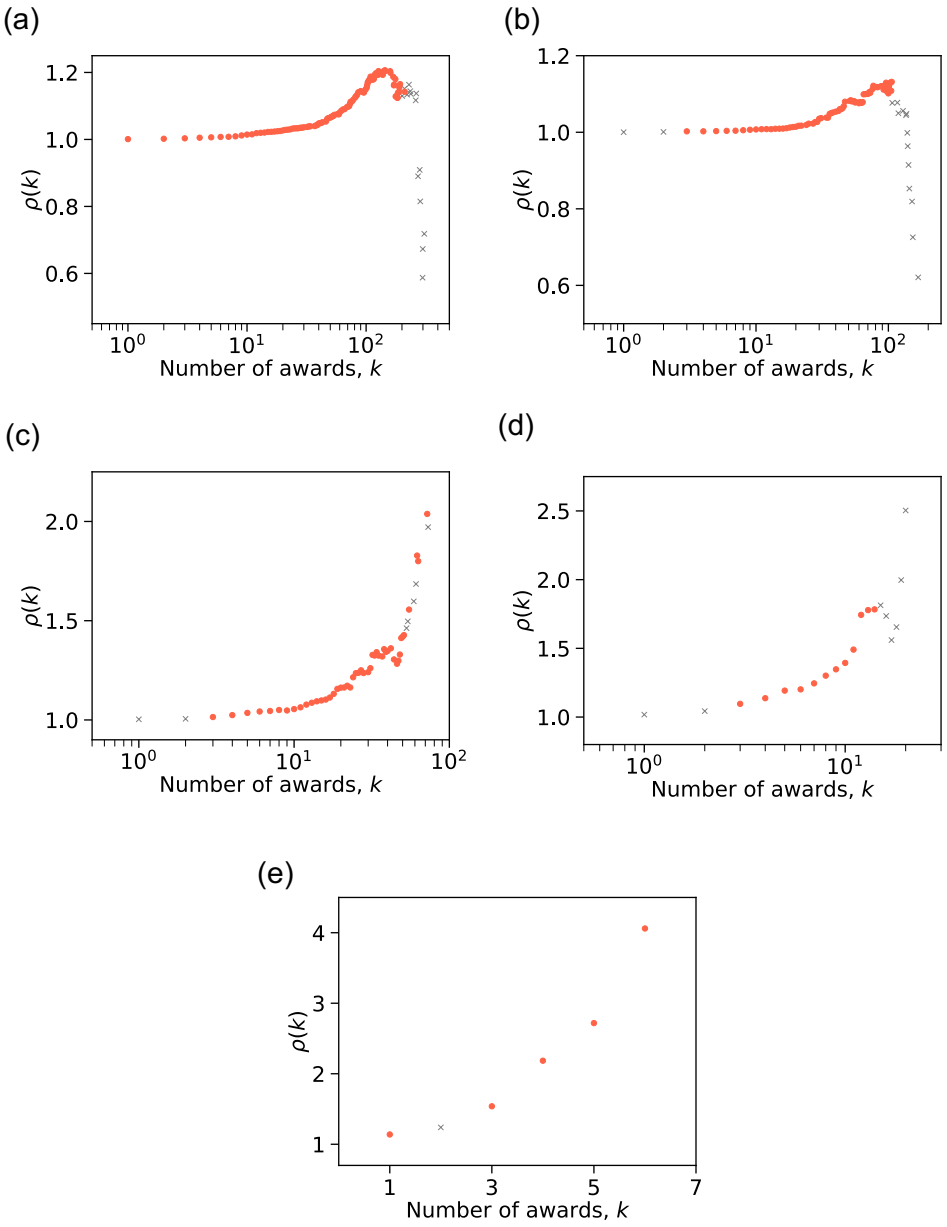


Figure 5.6: Normalized rich-club coefficient as a function of the number of awards received by the institution. A circle indicates a significant normalized rich-club coefficient ($P < 0.005$, Bonferroni-corrected permutation test). A cross indicates a non-significant normalized rich-club coefficient. (a) Entire network. (b) Subnetwork with $s = 2$. (c) Subnetwork with $s = 3$. (d) Subnetwork with $s = 4$. (e) Subnetwork with $s = 5$.

Chapter 6

Conclusions

In this thesis, I presented four works of my contributions to the field of social network analysis, with the hope to highlight the opportunities and promises to better understand social structures and dynamics. In the first two works presented in Chapters 2 and 3, we studied how to accurately estimate properties of OSNs by querying a small fraction of nodes via a random walk. In the last two works presented in Chapters 4 and 5, we studied how to analyze the structure and dynamics of real-world social networks involving higher-order interactions without using conventional one-mode projection. My research spirit, which is consistent throughout this thesis, is to realize analysis methods that are faithful to the empirical data and practical scenarios of real-world social networks.

To realize exhaustive analyses of online social networks with limited data access, I suggest investing in solutions for the social graph restoration problem, proposed in Chapter 3. Specifically, there are two future directions. The first is to pursue algorithms to estimate local structural properties more accurately based on the re-weighted random walk. Estimators of local structural properties seem to be an essential resource for restoring the original social network from its small sample. Developing more accurate random-walk-based estimators of local structural properties directly contributes to restoring the original social network. The second is to explore generative models that more accurately reproduce various structural properties of an empirical social network at hand using its local structural properties. Note that we require to estimate the input properties of such generative models in the social graph restoration problem. In these two respects, the dK-series provided one powerful solution for the social graph restoration problem.

Empirical networks involving higher-order interactions are increasingly available, and various measurements, dynamical process models, theories, and analytical methods have been developed for hypergraphs and bipartite graphs, especially in recent years. Whether networks involving higher-order interactions are represented by hypergraphs or bipartite graphs depends on the empirical data. As in Chapter 4, when we are mainly interested in the properties of higher-order interactions between nodes, a standard choice is to model the original network as a hypergraph. On the other hand, when a node has a specific role or meaning in a higher-order interaction, we should model the original network as a bipartite graph and we should not use the term 'hyperedges' for their interactions. For example, in the network of NSF's research grants in Chapter 5, each institution was responsible for a separate award in a collaborative grant, and hence, we should use bipartite-network representation.

The so-called 'big data' does not solve all the problems in social network analysis. For example, big data pitfalls learned from Google Flu Trends are

known [131]. Google Flu Trends, which was an active project between 2008 and 2015, tried to predict flu activity using a huge number of Google Search queries. Despite a huge amount of query data, Google Flu Trends was predicting over twice as many doctor visits as the Centers for Disease Control and Prevention (CDC) reported [43]. Two issues that contributed to the mistakes of Google Flu Trends were explored: 'big data hubris' and 'algorithm dynamics' [131]. In light of this lesson, let us take a bird's eye view of my works. In the work of Chapter 2 in this thesis, while the empirical data on social networks involving private nodes were already available in 2011, the issues related to private nodes were left until we addressed them in 2020. We addressed those issues by modeling a social network involving private nodes and developing algorithms considering them. In the work of Chapter 5, methods to detect the rich clubs and measure the research productivity help us to find associations between rich clubs and the research productivity. Note that it is practically difficult to strictly link each collaboration among institutions and its research outputs using dyadic network representation. Toward a comprehensive and deep understanding of social interactions, we require not only valuable empirical data but also analysis methods that are faithful to the empirical data and practical scenarios of real-world social networks, the latter being the common research spirit in all of my works in this thesis.

References

[1] National Science Foundation. The Grant Proposal Guide (GPG). https://www.nsf.gov/pubs/1999/nsf992/cont.htm, 1999. Accessed February 2022.

[2] National Science Foundation. Research collaboration among multiple institutions is growing trend. https://www.nsf.gov/news/news_summ.jsp?cntn_id=125070, 2012. Accessed Jun 2022.

[3] National Science Foundation. Proposal and Award Policies and Procedures Guide. https://www.nsf.gov/pubs/policydocs/pappg22_1/index.jsp, 2021. Accessed February 2022.

[4] Web of Science. Funding Acknowledgements. http://wokinfo.com/products_tools/multidisciplinary/webofscience/fundingsearch/, 2021. Accessed February 2022.

[5] Wikipedia Python library. https://github.com/goldsmith/Wikipedia, 2021. Accessed February 2022.

[6] National Science Foundation. Download awards by year. https://www.nsf.gov/awardsearch/download.jsp, 2022. Accessed February 2022.

[7] Web of Science. https://www.webofknowledge.com/, 2022. Accessed January 2022.

[8] K. Aagaard, A. Kladakis, and M. W. Nielsen. Concentration or dispersal of research funding? *Quantitative Science Studies*, 1:117–149, 2020.

[9] A. Abbasi, J. Altmann, and L. Hossain. Identifying the effects of co-authorship networks on the performance of scholars: A correlation and regression analysis of performance measures and social network analysis measures. *Journal of Informetrics*, 5:594–607, 2011.

[10] A. Abbasi, K. S. K. Chung, and L. Hossain. Egocentric analysis of co-authorship network structure, position and performance. *Information Processing and Management*, 48:671–679, 2012.

[11] G. Abramo, C. A. D'Angelo, and F. Di Costa. Research collaboration and productivity: is there correlation? *Higher Education*, 57(2):155–171, 2009.

[12] J. D. Adams, G. C. Black, J. R. Clemmons, and P. E. Stephan. Scientific teams and institutional collaborations: Evidence from u.s. universities, 1981–1999. *Research Policy*, 34:259–285, 2005.

[13] N. K. Ahmed, J. Neville, and R. Kompella. Network sampling: From static to streaming graphs. *ACM Transactions on Knowledge Discovery from Data*, 8, 2013. article No. 7.

[14] K. Ahn, K. Lee, and C. Suh. Hypergraph spectral clustering in the weighted stochastic block model. *IEEE Journal of Selected Topics in Signal Processing*, 12:959–974, 2018.

[15] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of Topological Characteristics of Huge Online Social Networking Services. In *Proceedings of the 16th International Conference on World Wide Web*, pages 835–844, 2007.

[16] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

[17] B. K. AlShebli, T. Rahwan, and W. L. Woon. The preeminence of ethnic diversity in scientific collaboration. *Nature Communications*, 9, 2018. Article No. 5163.

[18] U. Alvarez-Rodriguez, F. Battiston, G. F. de Arruda, Y. Moreno, M. Perc, and V. Latora. Evolutionary dynamics of higher-order interactions in social networks. *Nature Human Behaviour*, 5:586–595, 2021.

[19] S. Ankrah and O. AL-Tabbaa. Universities–industry collaboration: A systematic review. *Scandinavian Journal of Management*, 31:387–408, 2015.

[20] D. Antonakaki, P. Fragopoulou, and S. Ioannidis. A survey of twitter research: Data model, graph structure, sentiment analysis and attacks. *Expert Systems with Applications*, 164, 2021. Article No. 114006.

[21] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna. Four Degrees of Separation. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 33–42, 2012.

[22] D. A. Bader and K. Madduri. Parallel algorithms for evaluating centrality indices in real-world networks. In *2006 International Conference on Parallel Processing (ICPP)*, pages 539–550, 2006.

[23] B. Ball, B. Karrer, and M. E. J. Newman. Efficient and principled method for detecting communities in networks. *Physical Review E*, 84, 2011. Art. no. 036103.

[24] A. L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311:590–614, 2002.

[25] A. Barrat, M. Barthélemy, and A. Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, Cambridge, UK, 2008.

[26] K. E. Bassler, C. I. Del Genio, P. L. Erdős, I. Miklós, and Z. Toroczkai. Exact sampling of graphs with prescribed degree correlations. *New Journal of Physics*, 17, 2015. Art. no. 083052.

[27] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, pages 361–362, 2009.

[28] 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*, 874:1–92, 2020.

[29] C. Beaudry and S. Allaoui. Impact of public and private research funding on scientific production: The case of nanotechnology. *Research Policy*, 41:1589–1606, 2012.

[30] A. R. Benson, R. Abebe, M. T. Schaub, A. Jadbabaie, and J. Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences USA*, 115:E11221–E11230, 2018.

[31] A. R. Benson, D. F. Gleich, and J. Leskovec. Higher-order organization of complex networks. *Science*, 353:163–166, 2016.

[32] S. K. Bera and C. Seshadhri. How to count triangles, without seeing the whole graph. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 306–316, 2020.

[33] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424:175–308, 2006.

[34] Á. Bodó, G. Y. Katona, and P. L. Simon. SIS epidemic propagation on hypergraphs. *Bulletin of Mathematical Biology*, 78:713–735, 2016.

[35] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann. NetGAN: Generating graphs via random walks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 610–619, 2018.

[36] S. P. Borgatti, M. G. Everett, and J. C. Johnson. *Analyzing social networks*. SAGE Publications Ltd, 2018.

[37] A. A. Boroojeni, J. Dewar, T. Wu, and J. M. Hyman. Generating bipartite networks with a prescribed joint degree distribution. *Journal of Complex Networks*, 5:839–857, 2017.

[38] K. W. Boyack and K. Börner. Indicator-assisted evaluation and funding of research: Visualizing the influence of grants on the number and citation counts of research papers. *Journal of the Association for Information Science and Technology*, 54:447–461, 2003.

[39] B. Bozeman and E. Corley. Scientists' collaboration strategies: implications for scientific and technical human capital. *Research Policy*, 33:599–616, 2004.

[40] L. Bromham, R. Dinnage, and X. Hua. Interdisciplinary research has consistently lower funding success. *Nature*, 534:684–687, 2016. Article No. 7609.

[41] F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera. Comparing twitter and facebook user behavior: Privacy and other aspects. *Computers in Human Behavior*, 52:87–95, 2015.

[42] G. Burgio, J. T. Matamalas, S. Gómez, and A. Arenas. Evolution of cooperation in the presence of higher-order interactions: From networks to hypergraphs. *Entropy*, 22, 2020. Art. no. 744.

[43] D. Butler. When google got flu wrong: Us outbreak foxes a leading web-based method for tracking seasonal flu. *Nature*, 494:155–157, 2013.

[44] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9, 2009. article No. 717.

[45] S. A. Catanese, P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti. Crawling Facebook for Social Network Analysis Purposes. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, 2011. Article No. 52.

[46] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A recursive model for graph mining. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)*, pages 442–446, 2004.

[47] K. Chen, Y. Zhang, G. Zhu, and R. Mu. Do research institutes benefit from their network positions in research collaboration networks with industries or/and universities? *Technovation*, 94–95, 2020. Article No. 102002.

[48] X. Chen, Y. Li, P. Wang, and J. Lui. A General Framework for Estimating Graphlet Statistics via Random Walk. *Proceedings of the VLDB Endowment*, 10:253–264, 2016.

[49] F. Chiericetti, A. Dasgupta, R. Kumar, S. Lattanzi, and T. Sarlós. On sampling nodes in a network. In *Proceedings of the 25th International Conference on World Wide Web*, pages 471–481, 2016.

[50] P. S. Chodrow. Configuration models of random hypergraphs. *Journal of Complex Networks*, 8, 2020. Art. no. cnaa018.

[51] P. S. Chodrow, N. Veldt, and A. R. Benson. Hypergraph clustering: From blockmodels to modularity. *arXiv preprint arXiv:2101.09611*, 2021.

[52] G. Cimini, T. Squartini, F. Saracco, D. Garlaschelli, A. Gabrielli, and G. Caldarelli. The statistical physics of real-world networks. *Nature Reviews Physics*, 1:58–71, 2019.

[53] M. Cinelli. Generalized rich-club ordering in networks. *Journal of Complex Networks*, 7:702–719, 2019.

[54] M. Coccia and L. Wang. Evolution and convergence of the patterns of international scientific collaboration. *Proceedings of the National Academy of Sciences USA*, 113:2057–2061, 2016.

[55] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Routledge Academic, New York, NY, 1988.

[56] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin. Resilience of the internet to random breakdowns. *Physical Review Letters*, 85, 2000. Article No. 4626.

[57] V. Colizza, A. Flammini, M. A. Serrano, and A. Vespignani. Detecting rich-club ordering in complex networks. *Nature Physics*, 2:110–115, 2006.

[58] I. Cook, S. Grange, and A. Eyre-Walker. Research groups: How big should they be? *PeerJ*, 3, 2015. Article No. e989.

[59] N. A. Crossley, A. Mechelli, P. E. Vértes, T. T. Winton-Brown, A. X. Patel, C. E. Ginestet, P. McGuire, and E. T. Bullmore. Cognitive relevance of the community structure of the human brain functional coactivation network. *Proceedings of the National Academy of Sciences USA*, 110:11583–11588, 2013.

[60] J. N. Cummings and S. Kiesler. Collaborative research across disciplinary and organizational boundaries. *Social Studies of Science*, 35:703–722, 2005.

[61] J. N. Cummings and S. Kiesler. Coordination costs and project outcomes in multi-university collaborations. *Research Policy*, 36:1620–1634, 2007.

[62] A. Dasgupta, R. Kumar, and T. Sarlos. On Estimating the Average Degree. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 795–806, 2014.

[63] G. F. de Arruda, G. Petri, and Y. Moreno. Social contagion models on hypergraphs. *Physical Review Research*, 2, 2020. Art. no. 023032.

[64] G. F. de Arruda, M. Tizzani, and Y. Moreno. Phase transitions and stability of dynamical processes on hypergraphs. *Communications Physics*, 4, 2021. Art. no. 24.

[65] M. De Choudhury, Y.-R. Lin, H. Sundaram, K. S. Candan, L. Xie, and A. Kelliher. How does the data sampling strategy impact the discovery of information diffusion in social media? In *Fourth international AAAI conference on weblogs and social media (ICWSM)*, pages 34–41, 2010.

[66] R. Dey, Z. Jelveh, and K. Ross. Facebook users have become much more private: A large-scale study. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 346–352, 2012.

[67] P. Doreian, V. Batagelj, and A. Ferligoj. Generalized blockmodeling of two-mode network data. *Social Networks*, 26:29–53, 2004.

[68] H. Dundar and D. R. Lewis. Determinants of research productivity in higher education. *Research in Higher Education*, 39:607–631, 1998.

[69] A. Ebadi and A. Schiffauerova. How to receive more funding for your research? get connected to the right people! *PLOS ONE*, 10, 2015. Article No. e0133061.

[70] A. Ebadi and A. Schiffauerova. How to boost scientific production? a statistical analysis of research funding and other influencing factors. *Scientometrics*, 106:1093–1116, 2016.

[71] P. Erdős and A. Rényi. On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[72] Facebook. Facebook 500 Million Stories. `https://www.facebook.com/notes/facebook/500-million-stories/409753352130/`, 2010.

[73] Facebook. Q1 2022 Earnings. `https://investor.fb.com/investor-events/event-details/2022/Q1-2022-Earnings/default.aspx`, 2022.

[74] S. Feng, B. Hu, C. Nie, and X. Shen. Empirical study on a directed and weighted bus transport network in China. *Physica A: Statistical Mechanics and its Applications*, 441:85–92, 2016.

[75] S. Fortunato, C. T. Bergstrom, K. Börner, J. A. Evans, D. Helbing, S. Milojević, A. M. Petersen, F. Radicchi, R. Sinatra, B. Uzzi, A. Vespignani, L. Waltman, D. Wang, and A.-L. Barabási. Science of science. *Science*, 359, 2018. Article No. eaao0185.

[76] B. K. Fosdick, D. B. Larremore, J. Nishimura, and J. Ugander. Configuring random graph models with fixed degree sequences. *SIAM Review*, 60:315–355, 2018.

[77] L. C. Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press, Vancouver, BC, 2004.

[78] M. Fukuda, K. Nakajima, and K. Shudo. Estimating the bot population on twitter via random walk based sampling. *IEEE Access*, 10:17201–17211, 2022.

[79] J. Gómez-Gardeñes, M. Romance, R. Criado, D. Vilone, and A. Sánchez. Evolutionary games defined at the network mesoscale: The public goods game. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21, 2011. Art. no. 016113.

[80] D. Ghoshdastidar and A. Dukkipati. Consistency of spectral hypergraph partitioning under planted partition model. *Annals of Statistics*, 45:289–315, 2017.

[81] K. J. Gile, L. G. Johnston, and M. J. Salganik. Diagnostics for respondent-driven sampling. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 178, 2015. Article No. 241.

[82] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences USA*, 99:7821–7826, 2002.

[83] C. Giusti, R. Ghrist, and D. S. Bassett. Two's company, three (or more) is a simplex. *Journal of Computational Neuroscience*, 41:1–14, 2016.

[84] M. Gjoka, M. Kurant, and A. Markopoulou. 2.5K-graphs: From sampling to generation. In *2013 Proceedings of IEEE INFOCOM*, pages 1968–1976, 2013.

[85] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. Walking in Facebook: A Case Study of Unbiased Sampling of OSNs. In *2010 Proceedings IEEE INFOCOM*, pages 1–9, 2010.

[86] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. Practical Recommendations on Crawling Online Social Networks. *IEEE Journal on Selected Areas in Communications*, 29:1872–1892, 2011.

[87] S. Goel and M. J. Salganik. Assessing respondent-driven sampling. *Proceedings of the National Academy of Sciences*, 107:6743–6747, 2010.

[88] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2:129–233, 2010.

[89] L. A. Goodman. Snowball sampling. *The Annals of Mathematical Statistics*, pages 148–170, 1961.

[90] M. Grandjean. A social network analysis of Twitter: Mapping the digital humanities community. *Cogent Arts and Humanities*, 3, 2016. Article No. 1171458.

[91] J. Grilli, G. Barabás, M. J. Michalska-Smith, and S. Allesina. Higher-order interactions stabilize dynamics in competitive network models. *Nature*, 548:210–213, 2017.

[92] N. Grinberg, K. Joseph, L. Friedland, B. Swire-Thompson, and D. Lazer. Fake news on twitter during the 2016 us presidential election. *Science*, 363:374–378, 2019.

[93] J. L. Guillaume and M. Latapy. Bipartite graphs as models of complex networks. *Physica A: Statistical Mechanics and its Applications*, 371:795–813, 2006.

[94] M. Gulbrandsen and J.-C. Smeby. Industry funding and university professors ' research performance. *Research Policy*, 34:932–950, 2005.

[95] G. Han and H. Sethu. Waddling random walk: Fast and accurate mining of motif statistics in large graphs. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 181–190, 2016.

[96] M. S. Handcock and K. J. Gile. Modeling social networks from sampled data. *The Annals of Applied Statistics*, 4:5–25, 2010.

[97] S. J. Hardiman and L. Katzir. Estimating clustering coefficients and size of social networks via random walk. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 539–550, 2013.

[98] D. D. Heckathorn. Respondent-driven sampling: A new approach to the study of hidden populations. *Social Problems*, 44:174–199, 1997.

[99] A. Hickok, Y. Kureh, H. Z. Brooks, M. Feng, and M. A. Porter. A bounded-confidence model of opinion dynamics on hypergraphs. *arXiv preprint arXiv:2102.06825*, 2021.

[100] H. Hou, H. Kretschmer, and Z. Liu. The structure of scientific collaboration networks in scientometrics. *Scientometrics*, 75:189–202, 2008.

[101] Y. Hu. Efficient and high-quality force-directed graph drawing. *Mathematica Journal*, 10:37–71, 2005.

[102] J. Huang, A. J. Gates, R. Sinatra, and A.-L. Barabási. Historical comparison of gender inequality in scientific careers across countries and disciplines. *Proceedings of the National Academy of Sciences USA*, 117(9):4609–4616, 2020.

[103] M. Huisman. Imputation of missing network data: Some simple procedures. *Journal of Social Structure*, 10:1–29, 2009.

[104] D. R. Hunter. Curved exponential family models for social networks. *Social Networks*, 29:216–230, 2007.

[105] I. Iacopini, G. Petri, A. Barrat, and V. Latora. Simplicial models of social contagion. *Nature Communications*, 10, 2019. Art. no. 2485.

[106] J. Illenberger and G. Flötteröd. Estimating network properties from snowball sampled data. *Social Networks*, 34:701–711, 2012.

[107] B. A. Jacob and L. Lefgren. The impact of research grant funding on scientific productivity. *Journal of Public Economics*, 95:1168–1177, 2011.

[108] B. Jhun, M. Jo, and B. Kahng. Simplicial SIS model in scale-free uniform hypergraph. *Journal of Statistical Mechanics*, 2019, 2019. Art. no. 123207.

[109] B. F. Jones, S. Wuchty, and B. Uzzi. Multi-university research teams: Shifting impact, geography, and stratification in science. *Science*, 322:1259–1262, 2008.

[110] G. L. Jones. On the Markov chain central limit theorem. *Probability Surveys*, 1:299–320, 2004.

[111] B. Karrer and M. E. J. Newman. Random graphs containing arbitrary distributions of subgraphs. *Physical Review E*, 82, 2010. Art. no. 066118.

[112] L. Katzir, E. Liberty, and O. Somekh. Estimating sizes of social networks via biased sampling. In *Proceedings of the 20th International Conference on World Wide Web*, pages 597–606, 2011.

[113] Z. T. Ke, F. Shi, and D. Xia. Community detection for hypergraph networks via regularized tensor power iteration. *arXiv preprint arXiv:1909.06503*, 2019.

[114] M. Kim and J. Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *Proceedings of the 2011 SIAM International Conference on Data Mining (SDM)*, pages 47–58, 2011.

[115] B. Klimt and Y. Yang. The Enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pages 217–226, 2004.

[116] D. Knoke and S. Yang. *Social Network Analysis. Third Edition.* SAGE Publications Ltd, 2019.

[117] G. Kossinets. Effects of missing data in social networks. *Social Networks*, 28:247–268, 2006.

[118] B. Krishnamurthy, P. Gill, and M. Arlitt. A few chirps about twitter. In *Proceedings of the First Workshop on Online Social Networks*, page 19–24, 2008.

[119] J. Kunegis. KONECT–The Koblenz Network Collection. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1343–1350, 2013.

[120] M. Kurant, M. Gjoka, C. T. Butts, and A. Markopoulou. Walking on a Graph with a Magnifying Glass: Stratified Sampling via Weighted Random Walks. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, pages 281–292, 2011.

[121] M. Kurant, A. Markopoulou, and P. Thiran. Towards unbiased BFS sampling. *IEEE Journal on Selected Areas in Communications*, 29:1799–1809, 2011.

[122] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th International Conference on World Wide Web*, pages 591–600, 2010.

[123] R. Lambiotte, M. Rosvall, and I. Scholtes. From networks to optimal higher-order models of complex systems. *Nature Physics*, 15:313–320, 2019.

[124] N. W. Landry and J. G. Restrepo. The effect of heterogeneity on hypergraph contagion models. *Chaos*, 30, 2020. Art. no. 103117.

[125] V. Larivière, S. Haustein, and K. Börner. Long-distance interdisciplinarity leads to higher scientific impact. *PLOS ONE*, 10, 2015. Article No. e0122565.

[126] D. B. Larremore, A. Clauset, and A. Z. Jacobs. Efficiently inferring community structure in bipartite networks. *Physical Review E*, 90, 2014. Art. no. 012805.

[127] M. Latapy, C. Magnien, and N. D. Vecchio. Basic notions for the analysis of large two-mode networks. *Social Networks*, 30:31–48, 2008.

[128] V. Latora, V. Nicosia, and G. Russo. *Complex Networks: Principles, Methods and Applications*. Cambridge University Press, Cambridge, UK, 2017.

[129] M. S. Lauer. Citations per dollar as a measure of productivity. `https://nexus.od.nih.gov/all/2016/04/28/citations-per-dollar/`, 2016. Accessed February 2022.

[130] M. S. Lauer and D. Roychowdhury. Inequalities in the distribution of national institutes of health research project grant funding. *eLife*, 10, 2021. Article No. e71712.

[131] D. Lazer, R. Kennedy, G. King, and A. Vespignani. The parable of google flu: Traps in big data analysis. *Science*, 343:1203–1205, 2014.

[132] C.-H. Lee, X. Xu, and D. Y. Eun. Beyond random walk and metropolis-hastings samplers: Why you should not backtrack for unbiased graph sampling. *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, 40:319–330, 2012.

[133] G. Lee, M. Choe, and K. Shin. How do hyperedges overlap in real-world hypergraphs? - Patterns, measures, and generators. In *Proceedings of the Web Conference 2021*, pages 3396–3407, 2021.

[134] S. H. Lee, P.-J. Kim, and H. Jeong. Statistical properties of sampled networks. *Physical Review E*, 73, 2006. article No. 016102.

[135] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11:985–1042, 2010.

[136] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 631–636, 2006.

[137] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[138] D. A. Levin and Y. Peres. *Markov Chains and Mixing Times*, volume 107. American Mathematical Soc., 2017.

[139] M. Li, J. Wu, D. Wang, T. Zhou, Z. Di, and Y. Fan. Evolving model of weighted networks inspired by scientific collaboration networks. *Physica A: Statistical Mechanics and its Applications*, 375:355–364, 2007.

[140] R.-H. Li, J. X. Yu, L. Qin, R. Mao, and T. Jin. On random walk based graph sampling. In *2015 IEEE 31st International Conference on Data Engineering*, pages 927–938, 2015.

[141] Y. Li, Z. Wu, S. Lin, H. Xie, M. Lv, Y. Xu, and J. C. Lui. Walking with perception: Efficient random walk sampling via common neighbor awareness. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 962–973, 2019.

[142] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, 58:1019–1031, 2007.

[143] M. M. Little, C. A. St Hill, K. B. Ware, M. T. Swanoski, S. A. Chapman, M. N. Lutfiyya, and F. B. Cerra. Team science as interprofessional collaborative research practice: a systematic review of the science of team science literature. *Journal of Investigative Medicine*, 65:15–22, 2017.

[144] L. Lovász. Random walks on graphs: A survey. In *Combinatorics, Paul Erdős is Eighty*, volume 2, pages 353–398. 1996.

[145] X. Lu, L. Bengtsson, T. Britton, M. Camitz, B. J. Kim, A. Thorson, and F. Liljeros. The sensitivity of respondent-driven sampling. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 175:191–216, 2012.

[146] M. Lucas, G. Cencetti, and F. Battiston. Multiorder laplacian for synchronization in higher-order networks. *Physical Review Research*, 2, 2020. Art. no. 033410.

[147] A. Ma, R. J. Mondragón, and V. Latora. Anatomy of funded research in science. *Proceedings of the National Academy of Sciences USA*, 112:14760–14765, 2015.

[148] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat. Systematic topology analysis and generation using degree correlations. *SIGCOMM Computer Communication Review*, 36:135–146, 2006.

[149] K. Mamiseishvili and V. J. Rosser. International and citizen faculty in the united states: An examination of their productivity at research universities. *Research in Higher Education*, 51, 2009. Article No. 88.

[150] S. Maslov, K. Sneppen, and A. Zaliznyak. Detection of topological patterns in complex networks: Correlation profile of the Internet. *Physica A: Statistical Mechanics and its Applications*, 333:529–540, 2004.

[151] R. Mastrandrea, J. Fournet, and A. Barrat. Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLOS ONE*, 10, 2015. Art. no. e0136497.

[152] G. Melin and O. Persson. Studying research collaboration using co-authorships. *Scientometrics*, 36:363–377, 1996.

[153] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298:824–827, 2002.

[154] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pages 29–42, 2007.

[155] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6:161–180, 1995.

[156] R. Mulas, C. Kuehn, and J. Jost. Coupled dynamics on hypergraphs: Master stability of steady states and synchronization. *Physical Review E*, 101, 2020. Art. no. 062313.

[157] S. A. Myers, A. Sharma, P. Gupta, and J. Lin. Information Network or Social Network? The Structure of the Twitter Follow Graph. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 493–498, 2014.

[158] R. Nagarajan, A. T. Kalinka, and W. R. Hogan. Evidence of community structure in biomedical research grant collaborations. *Journal of Biomedical Informatics*, 46:40–46, 2013.

[159] K. Nakajima. Code and datasets. `https://www.dropbox.com/sh/qrtxb1p7ifhd58f/AADBseKsUzVqPge2ZEvtvDKNa?dl=0`.

[160] K. Nakajima, K. Iwasaki, T. Matsumura, and K. Shudo. Estimating top-k betweenness centrality nodes in online social networks. In *2018 IEEE Intl Conf on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pages 1128–1135, 2018.

[161] K. Nakajima and K. Shudo. Estimating high betweenness centrality nodes via random walk in social networks. *Journal of Information Processing*, 28:436–444, 2020.

[162] K. Nakajima and K. Shudo. Estimating properties of social networks via random walk considering private nodes. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 720–730, 2020. DOI: 10.1145/3394486.3403116.

[163] K. Nakajima and K. Shudo. Measurement error of network clustering coefficients under randomly missing nodes. *Scientific Reports*, 11, 2021. Article No. 2815.

[164] K. Nakajima and K. Shudo. Random walk sampling in social networks involving private nodes. *ACM Transactions on Knowledge Discovery from Data*, 17, 2022. Article No. 51. DOI: 10.1145/3561388.

[165] K. Nakajima and K. Shudo. Social Graph Restoration via Random Walk Sampling. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 806–819, 2022. DOI: 10.1109/ICDE53745.2022.00065.

[166] K. Nakajima, K. Shudo, and N. Masuda. Randomizing hypergraphs preserving degree correlation and local clustering. *IEEE Transactions on Network Science and Engineering*, 9:1139–1153, 2022. DOI: 10.1109/TNSE.2021.3133380.

[167] K. Nakajima, K. Shudo, and N. Masuda. Higher-order rich-club phenomenon in collaborative research grant networks. *Scientometrics*, 2023. DOI: 10.1007/s11192-022-04621-1. In press.

[168] A. Nazi, Z. Zhou, S. Thirumuruganathan, N. Zhang, and G. Das. Walk, not wait: Faster sampling over online social networks. *Proceedings of the VLDB Endowment*, 8:678–689, 2015.

[169] L. Neuhäuser, A. Mellor, and R. Lambiotte. Multibody interactions and nonlinear consensus dynamics on networked systems. *Physical Review E*, 101, 2020. Art. no. 032310.

[170] M. E. J. Newman. Scientific collaboration networks. I. Network construction and fundamental results. *Physical Review E*, 64, 2001. Art. no. 016131.

[171] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences USA*, 98:404–409, 2001.

[172] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences USA*, 98:404–409, 2001.

[173] M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89, 2002. Art. no. 208701.

[174] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences USA*, 103:8577–8582, 2006.

[175] M. E. J. Newman. Random graphs with clustering. *Physical Review Letters*, 103, 2009. Art. no. 058701.

[176] M. E. J. Newman. Network structure from rich but noisy data. *Nature Physics*, 14:542–545, 2018.

[177] M. E. J. Newman. *Networks. Second Edition.* Oxford University Press, Oxford, UK, 2018.

[178] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64, 2001. Art. no. 026118.

[179] T. Opsahl, V. Colizza, P. Panzarasa, and J. J. Ramasco. Prominence and control: The weighted rich-club effect. *Physical Review Letters*, 101, 2008. Article No. 168702.

[180] C. Orsini, M. M. Dankulov, P. Colomer-de-Simón, A. Jamakovic, P. Mahadevan, A. Vahdat, K. E. Bassler, Z. Toroczkai, M. Boguñá, G. Caldarelli, S. Fortunato, and D. Krioukov. Quantifying randomness in real networks. *Nature Communications*, 6, 2015. Art. no. 8627.

[181] R. Pastor-Satorras, A. Vázquez, and A. Vespignani. Dynamical and correlation properties of the Internet. *Physical Review Letters*, 87, 2001. Art. no. 258701.

[182] A. Patania, G. Petri, and F. Vaccarino. The shape of collaborations. *EPJ Data Science*, 6, 2017. Art. no. 18.

[183] A. Payne and A. Siow. Does federal research funding increase university research output? *The B.E. Journal of Economic Analysis and Policy*, 3:1–24, 2003.

[184] C. Payrató-Borras, L. Hernández, and Y. Moreno. Breaking the spell of nestedness: The entropic origin of nestedness in mutualistic systems. *Physical Review X*, 9, 2019. Art. no. 031024.

[185] J. Peña and Y. Rochat. Bipartite graphs as models of population structures in evolutionary multiplayer games. *PLOS ONE*, 7, 2012. Art. no. e44514.

[186] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[187] F. Radicchi, S. Fortunato, and C. Castellano. Universality of citation distributions: Toward an objective measure of scientific impact. *Proceedings of the National Academy of Sciences USA*, 105:17268–17272, 2008.

[188] J. J. Ramasco, S. N. Dorogovtsev, and R. Pastor-Satorras. Self-organization of collaboration networks. *Physical Review E*, 70, 2004. Art. no. 036106.

[189] J. J. Ramasco and S. A. Morris. Social inertia in collaboration networks. *Physical Review E*, 73, 2006. Art. no. 016122.

[190] B. Ribeiro and D. Towsley. Estimating and Sampling Graphs with Multidimensional Random Walks. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, pages 390–403, 2010.

[191] G. O. Roberts and J. S. Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71, 2004.

[192] G. Robins, P. Pattison, and J. Woolcock. Missing data in networks: exponential random graph (p*) models for networks with non-respondents. *Social Networks*, 26:257–283, 2004.

[193] G. Robins, P. Pattison, Y. Kalish, and D. Lusher. An introduction to exponential random graph (p*) models for social networks. *Social Networks*, 29:173–191, 2007.

[194] L. E. C. Rocha, A. E. Thorson, R. Lambiotte, and F. Liljeros. Respondent-driven sampling bias induced by community structure and response rates in social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180:99–118, 2017.

[195] J. L. Rosenbloom, D. K. Ginther, T. Juhl, and J. A. Heppert. The effects of research & development funding on scientific productivity: Academic chemistry, 1990-2009. *PLOS ONE*, 10, 2015. Article No. e0138176.

[196] R. A. Rossi and N. K. Ahmed. The Network Data Repository with Interactive Graph Analytics and Visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 4292–4293, 2015.

[197] B. Rozemberczki, O. Kiss, and R. Sarkar. Little ball of fur: A python library for graph sampling. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pages 3133–3140, 2020.

[198] R. Sahasrabuddhe, L. Neuhäuser, and R. Lambiotte. Modelling non-linear consensus dynamics on hypergraphs. *Journal of Physics: Complexity*, 2, 2021. Art. no. 025006.

[199] M. J. Salganik and D. D. Heckathorn. Sampling and estimation in hidden populations using respondent-driven sampling. *Sociological Methodology*, 34:193–240, 2004.

[200] A. Salova and R. M. D'Souza. Cluster synchronization on hypergraphs. *arXiv preprint arXiv:2101.05464*, 2021.

[201] F. Saracco, R. Di Clemente, A. Gabrielli, and T. Squartini. Randomizing bipartite networks: The case of the world trade web. *Scientific Reports*, 5, 2015. Art. no. 10595.

[202] F. Saracco, M. J. Straka, R. Di Clemente, A. Gabrielli, G. Caldarelli, and T. Squartini. Inferring monopartite projections of bipartite networks: An entropy-based approach. *New Journal of Physics*, 19, 2017. Art. no. 053022.

[203] S. S. Sawilowsky. New effect size rules of thumb. *Journal of Modern Applied Statistical Methods*, 8:26, 2009.

[204] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie. Random walks on simplicial complexes and the normalized Hodge 1-Laplacian. *SIAM Review*, 62:353–391, 2020.

[205] T. A. Schieber, L. Carpi, A. Díaz-Guilera, P. M. Pardalos, C. Masoller, and M. G. Ravetti. Quantification of network structural dissimilarities. *Nature Communications*, 8, 2017. article No. 13928.

[206] J. Scott. Social network analysis. *Sociology*, 22:109–127, 1988.

[207] M. Á. Serrano and M. Boguñá. Tuning clustering in random networks with arbitrary degree distributions. *Physical Review E*, 72, 2005. Art. no. 036133.

[208] A. Shine and D. Kempe. Generative graph models based on laplacian spectra. In *The World Wide Web Conference*, pages 1691–1701, 2019.

[209] J. A. Smith and J. Moody. Structural effects of network sampling coverage I: Nodes missing at random. *Social Networks*, 35:652–668, 2013.

[210] SocioPatterns. `http://www.sociopatterns.org`.

[211] I. Stanton and A. Pinar. Constructing and sampling graphs with a prescribed joint degree distribution. *Journal of Experimental Algorithmics*, 17, 2012. Art. no. 3.5.

[212] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J.-F. Pinton, M. Quaggiotto, W. Van den Broeck, C. Régis, B. Lina, and P. Vanhems. High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6, 2011. Art. no. e23176.

[213] Y. Sun, G. Livan, A. Ma, and V. Latora. Interdisciplinary researchers attain better long-term funding performance. *Communications Physics*, 4, 2021. Article No. 263.

[214] Q. Suo, J.-L. Guo, and A.-Z. Shen. Information spreading dynamics in hypernetworks. *Physica A: Statistical Mechanics and its Applications*, 495:475–487, 2018.

[215] M. Szell and R. Sinatra. Research funding goes to rich clubs. *Proceedings of the National Academy of Sciences USA*, 112:14749–14750, 2015.

[216] L. Takac and M. Zabovsky. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, 2012.

[217] F. Tarissan, B. Quoitin, P. Mérindol, B. Donnet, J. J. Pansiot, and M. Latapy. Towards a bipartite graph modeling of the Internet topology. *Computer Networks*, 57:2331–2347, 2013.

[218] B. Tillman, A. Markopoulou, M. Gjoka, and C. T. Buttsc. 2K+ graph construction framework: Targeting joint degree matrix and beyond. *IEEE/ACM Transactions on Networking*, 27:591–606, 2019.

[219] A. Tomas and K. J. Gile. The effect of differential recruitment, non-response and non-recruitment on estimators for respondent-driven sampling. *Electronic Journal of Statistics*, 5:899–934, 2011.

[220] A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications: Statistical Mechanics and its Applications*, 391:4165–4180, 2012.

[221] Twitter. Twitter API GET followers/ids. `https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-followers-ids.html`, 2022. Accessed on April 2022.

[222] Twitter. Twitter api get friends/ids. `https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friends-ids.html`, 2022. Accessed on April 2022.

[223] S. Uddin, L. Hossain, and K. Rasmussen. Network effects on scientific collaborations. *PLOS ONE*, 8, 2013. Article No. e57546.

[224] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.

[225] M. P. van den Heuvel, R. S. Kahn, J. Goñi, and O. Sporns. High-cost, high-capacity backbone for global brain communication. *Proceedings of the National Academy of Sciences USA*, 109:11372–11377, 2012.

[226] M. P. van den Heuvel and O. Sporns. Rich-club organization of the human connectome. *Journal of Neuroscience*, 31:15775–15786, 2011.

[227] K. Van Koevering, A. Benson, and J. Kleinberg. Random graphs with prescribed k-core sequences: A new null model for network analysis. In *Proceedings of the Web Conference 2021*, page 367–378, 2021.

[228] O. Varol, E. Ferrara, C. Davis, F. Menczer, and A. Flammini. Online human-bot interactions: Detection, estimation, and characterization. *Proceedings of the International AAAI Conference on Web and Social Media*, 11:280–289, 2017.

[229] E. Vasilyeva, A. Kozlov, K. Alfaro-Bittner, D. Musatov, A. M. Raigorodskii, M. Perc, and S. Boccaletti. Multilayer representation of collaboration networks with higher-order interactions. *Scientific Reports*, 11, 2021. article No. 5666.

[230] E. Volz and D. D. Heckathorn. Probability based estimation theory for respondent driven sampling. *Journal of Official Statistics*, 24:79–97, 2008.

[231] W. P. Wahls. The national institutes of health needs to better balance funding distributions among us institutions. *Proceedings of the National Academy of Sciences USA*, 116:13150–13154, 2019.

[232] L. Waltman. A review of the literature on citation impact indicators. *Journal of Informetrics*, 10:365–391, 2016.

[233] J. Wang. Knowledge creation in collaboration networks: Effects of tie configuration. *Research Policy*, 45:68–80, 2016.

[234] P. Wang, J. Lui, B. Ribeiro, D. Towsley, J. Zhao, and X. Guan. Efficiently Estimating Motif Statistics of Large Networks. *ACM Transactions on Knowledge Discovery from Data*, 9, 2014. Article No. 8.

[235] Y. Wang, F. Deng, Y. Jia, J. Wang, S. Zhong, H. Huang, L. Chen, G. Chen, H. Hu, L. Huang, and R. Huang. Disrupted rich club organization and structural brain connectome in unmedicated bipolar disorder. *Psychological Medicine*, 49:510–518, 2019.

[236] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK, 1994.

[237] D. J. Watts and S. H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393:440–442, 1998.

[238] B. Western, A. Braga, D. Hureau, and C. Sirois. Study retention as bias reduction in a hard-to-reach population. *Proceedings of the National Academy of Sciences*, 113:5477–5485, 2016.

[239] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European Conference on Computer Systems*, pages 205–218, 2009.

[240] L. Wu, D. Wang, and J. A. Evans. Large teams develop and small teams disrupt science and technology. *Nature*, 566:378–382, 2019.

[241] S. Wuchty, B. F. Jones, and B. Uzzi. The increasing dominance of teams in production of knowledge. *Science*, 316:1036–1039, 2007.

[242] Y. Xie. “Undemocracy”: inequalities in science. *Science*, 344:809–810, 2014.

[243] Q. Ye, H. Song, and T. Li. Cross-institutional collaboration networks in tourism and hospitality research. *Tourism Management Perspectives*, 2-3:55–64, 2012.

[244] T.-C. Yen and D. B. Larremore. Community detection in bipartite networks with stochastic block models. *Physical Review E*, 102, 2020. Art. no. 032309.

[245] P. Yi, H. Xie, Y. Li, and J. C. Lui. A bootstrapping approach to optimize random walk based statistical estimation over graphs. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 900–911, 2021.

[246] Z. Yin, Z. Liang, and Q. Zhi. Does the concentration of scientific research funding in institutions promote knowledge output? *Journal of Informetrics*, 12:1146–1159, 2018.

[247] S. Yoon, H. Song, K. Shin, and Y. Yi. How much and when do we need higher-order information in hypergraphs? A case study on hyperedge prediction. In *Proceedings of The Web Conference 2020*, pages 2627–2633, 2020.

[248] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 5708–5717, 2018.

[249] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[250] A. Zeng, Y. Fan, Z. Di, Y. Wang, and S. Havlin. Fresh teams are associated with original and multidisciplinary research. *Nature Human Behaviour*, 5:1314–1322, 2021.

[251] A. Zeng, Z. Shen, J. Zhou, J. Wu, Y. Fan, Y. Wang, and H. E. Stanley. The science of science: From the perspective of complex systems. *Physics Reports*, 714–715:1–73, 2017.

[252] Q. Zhi and T. Meng. Funding allocation, inequality, and scientific research output: an empirical study based on the life science sector of natural science foundation of china. *Scientometrics*, 106:603–628, 2016.

[253] S. Zhou and R. Mondragon. The rich-club phenomenon in the internet topology. *IEEE Communications Letters*, 8:180–182, 2004.

[254] W. Zhou and L. Nakhleh. Properties of metabolic graphs: Biological organization or representation artifacts? *BMC Bioinformatics*, 12, 2011. Art. no. 132.

[255] Z. Zhou, N. Zhang, Z. Gong, and G. Das. Faster random walks by rewiring online social networks on-the-fly. *ACM Transactions on Database Systems*, 40, 2016. Article No. 26.

[256] L. G. Zucker, M. R. Darby, J. Furner, R. C. Liu, and H. Ma. Minerva unbound: Knowledge stocks, knowledge flows and new knowledge production. *Research Policy*, 36:850–863, 2007.