Sampling Graphlets of Multiplex Networks: A Restricted Random Walk Approach

SIMIAO JIAO, School of Information Science and Engineering, Fudan University
ZIHUI XUE, School of Information Science and Engineering, Fudan University
XIAOWEI CHEN, Bytedance Inc.
YUEDONG XU, School of Information Science and Engineering, Fudan University

Graphlets are induced subgraph patterns that are crucial to the understanding of the structure and function of a large network. A lot of effort has been devoted to calculating graphlet statistics where random walk-based approaches are commonly used to access restricted graphs through the available application programming interfaces (APIs). However, most of them merely consider individual networks while overlooking the strong coupling between different networks. In this article, we estimate the graphlet concentration in multiplex networks with real-world applications. An inter-layer edge connects two nodes in different layers if they actually belong to the same node. The access to a multiplex network is restrictive in the sense that the upper layer allows random walk sampling, whereas the nodes of lower layers can be accessed only through the inter-layer edges and only support random node or edge sampling. To cope with this new challenge, we define a suit of two-layer graphlets and propose novel random walk sampling algorithms to estimate the proportion of all the three-node graphlets. An analytical bound on the sampling steps is proved to guarantee the convergence of our unbiased estimator. We further generalize our algorithm to explore the tradeoff between the estimated accuracy of different graphlets when the sample budget is split into different layers. Experimental evaluation on real-world and synthetic multiplex networks demonstrates the accuracy and high efficiency of our unbiased estimators.

CCS Concepts: • Information systems → Social networks;

Additional Key Words and Phrases: Graphlets, multiplex network, graph sampling, random walk, unbiased estimation

ACM Reference format:
Simiao Jiao, Zihui Xue, Xiaowei Chen, and Yuedong Xu. 2021. Sampling Graphlets of Multiplex Networks: A Restricted Random Walk Approach. ACM Trans. Web 15, 4, Article 18 (June 2021), 31 pages. https://doi.org/10.1145/3456291
1 INTRODUCTION

Complex networks have attracted great attention due to their ample examples in the real world such as road networks [16, 20], social networks [12, 43], and biological networks [41]. With an enormous amount of data in these fields being available, significant advances in understanding the structure and function of networks, and mathematical models of networks have been achieved in the past decade. Extensive efforts have been devoted to characterizing network properties, including measures of degree distribution, node clustering, network modularity, local graph structures and so on [25, 41, 52]. However, the literature deals almost exclusively with single-layer networks whose nodes and edges of a network exist in an isolated system. In many state-of-the-art systems, an individual network is actually one component within a more complicated multiplex network, or shows a strong coupling with other networks. Consider the scenario where we have two Online Social Networks (OSNs) Facebook and Twitter. Because of the diversity in their services, a fraction of users may possess the identities of both sites, thus linking them together. We define there exists a link between two accounts in Facebook and Twitter respectively if they belong to the same person. A single-layer model is not capable of capturing the links between these networks. Multiplex (two-layer by default) networks that consist of layers of several networks overcome the disadvantages of single-layer models and are able to measure interactions across different networks. Still take Facebook and Twitter as examples. These two OSNs can form a two-layer network with each OSN on a layer. The “intra-layer” links (links within one layer) are friend relationships between accounts in the corresponding OSNs, and the “inter-layer” links are connections between the same person in Facebook and Twitter. More examples on multiplex networks include the cyber-physical systems where one layer can be a physical acquaintance network of users.

Graphlets, which are referred to as induced subgraph patterns or motifs, are the building block of complex networks. One famous example in the graphlet family is the triangle. Computing graphlet counts in a network is an important task, because the frequencies of graphlets offer important statistics to characterize the local topology structures. For instance, Heider developed the balance theory [23] that uses three-node graphlets to explain social proverbs: “A friend of my friend is my friend” and “The enemy of my enemy is my friend.” Kunegis et al. took the concentration of graphlets as a metric to gauge the stability of signed friend or foe subgraphs [33]. Juszczyszyn et al. used the triad transition pattern to predict whether a link would be constructed between pairwise users at a future time. Rahman and Hasan proposed to extract feature representation of graphlet transition events for link prediction in dynamic networks [29].

Despite the comprehensive research on graphlets in single-layer networks, it remains largely open in multiplex networks. The definition of multiplex graphlets itself is the first obstacle in which we generalize the single-layer counterparts to this new scenario. Taking the two-layer network in Figure 1 as a simple example where the upper-layer edges are colored in blue, and the lower-layer ones are colored in red. The gray dashed lines indicate that the corresponding nodes appear in both layers. The subgraphs in Figure 1(a) and (b) represents the two-layer three-node graphlets that characterize their local structures in the blue and red graphs. One can squish a two-layer graphlet into a single-layer subgraph, e.g., Figure 1(a) to (c) and Figure 1(b) to (d), and use different colors to differentiate the edges in different layers. If not mentioned explicitly, then we choose the squished subgraphs (e.g., Figure 1(c) and (d)) to visualize the two-layer graphlets for simplicity throughout this work. The multiplex graphlets not only inherit their significance in each individual networks, but also reveal the interactions between networks. The graphlet in Figure 1(a) manifests a strong tie among three users due to their dense connectivity in both layers. In Figure 1(b), user A can expand the coverage of service recommendation: diffusing information to his neighbor B.
at the blue layer and reaching user C at the red layer. Furthermore, the triad-based friendship recommendation [53] and spam detection [8] can benefit from the multiplex structure.

The purpose of our study is to efficiently compute the frequencies of each two-layer graphlet that appeared in a given graph. The percentage of a particular graphlet type is called the “graphlet concentration” or “graphlet statistics.”

Challenges. The first challenge is the prohibitive complexity of exhaustive counting. As is well known, there exist a huge amount of graphlets even for a moderately sized single-layer graph. For two-layer networks, the time complexity of exhaustive counting is even higher, because there are more distinct graphlets. For example, for the real-world dataset ff-tw (it has moderate size) we used in the experiment part, using the tool provided in Reference [51], it takes more than 75 minutes to exactly count the concentration of graphlets in a machine with 2.5 GHz CPU and 8 G RAM. However, based on the sampling algorithm, it may only take several seconds to obtain a result with acceptable accuracy.

Therefore, we resort to an alternative approach that “samples” a small fraction of nodes and edges of the graph in order to significantly reduce running time with an acceptable error level. A string of sampling algorithms have been proposed to estimate graphlet concentration so far [2, 3, 9, 14, 27, 28, 30, 42, 44, 47, 48, 49].

The second challenge is the restricted access to the complete graph data. The restriction is even more stringent in multiplex networks than in single-layer ones, which gives rise to a fundamentally different rationale in designing sampling approaches. The OSN service providers are unwilling to share the complete graph but allow the calling of some application programming interfaces (APIs). With these APIs, we can query a node and retrieve a list of the neighboring edges. Random walk [40] is the de facto approach to tackle the restricted graph access problem in single-layer networks. However, additional restrictions have been imposed on the sampling in two-layer networks. Consider a "social-hub" network [21] consisting of a Twitter-like open social network (upper layer) and a Facebook-like privacy-sensitive social network (lower layer). The upper layer allows random walks through the provided APIs, while the lower layer only lists the direct friends of a sampled user. That means we have unequal access restrictions on both layers. In a coupled cyber-physical network with a Facebook layer and a physical acquaintance layer. The crawler can randomly walk on the Facebook graph, but it is costly to dive into the physical layer to “query”

---

1The default privacy setting of existing OSNs usually goes to two extremes. On Twitter, a stranger is allowed to visit the friend list of a user and explores the friends of his friends while in LinkedIn, a stranger is forbidden to see the friend list. We consider a semi-private setting that will hopefully be considered in the future: A stranger is allowed to see the friend list of a user but cannot explore the friends of his friends.
all the real-world friends of a person or his friends-of-friends. Usually speaking, taking a random walk on different network layers may have different costs, if the costs have huge differences like the cyber-physical case, then we may want to try our best to obtain the estimation by just taking a random walk on the layer with lower cost. The specific strategy could be determined by the ratio of costs. Here we consider a primary situation. In a word, the sampling of two-layer graphlets allows random walk on one layer and only node or edge sampling on the other layer. Such a restriction makes the existing single-layer random walk approaches no longer applicable.

1.1 Related work

**Exact Graphlet Counting.** Counting subgraphs is a computationally intensive task. Ahmed et al. proposed a fast parallel algorithm that leverages a number of proven combinatorial arguments for different graphlets with three or four nodes [5]. Hocevar and Demsar [24] proposed a combinatorial method that builds a system of equations to allow the computing of the number of occurrence of graphlets with up to five nodes. Suri et al. [45] presented a triangle counting algorithm tailored for MapReduce computation in parallel.

**Single-layer Network Sampling.** Existing sampling methods can be classified into three categories according to the graph access modes. The first one is memory-based, in which the entire graph can be directly accessed so that either the node sampling or the edge sampling can be easily implemented [28, 42, 44, 49]. The second one is the graph stream. A streaming graph is a rapid, continuous, and possibly unbounded time-varying stream of edges [4] that is usually hard to be put in a small memory or is not fully observable at any time. Some related works used a traversal on the edge streaming, e.g., Reference [27], and some others adopted sampling, e.g., Reference [3]. Single-pass reservoir sampling schemes are proposed in Reference [47] in which the sample is maintained incrementally over the stream, and remains useful at any time for stream property estimations. Ahmed et al. [2] proposed the graph priority sampling method, an improvement over order-based reservoir sampling, through weighting the sampling of edges. The third category is restricted access. For a large network, we can only access it by using some APIs, for example, large OSNs such as Facebook or Twitter. To sample this graph, we need to crawl the identity of a person (i.e., a node) as well as his neighbors, and then randomly select one of his neighbors to crawl repeatedly. This method is named random walk and is commonly used to estimate the degree distribution [34, 37], the concentration of graphlets [9, 14, 48] or the clustering coefficient [30].

**Multiplex Network Sampling.** The sampling of multiplex heterogeneous networks is very rarely studied. The plausible reason is that the classical sampling algorithms can be generalized to the situations where the nodes or the edges are of multiple properties. Gjoka et al. observed multitype relationships (edges) among nodes of OSNs, combine all individual graphs into a single union graph before sampling [19]. Li and Yeh decomposed an OSN into a multiple-layer one in which the nodes at the same layer possess the same identity [36]. Three known methods (i.e., random node, random walk and respondent-driven sampling) are applied to estimate the node or edge type distributions. In Reference [7] and Reference [22], the authors proposed algorithms of motif detection in multilayer networks. D. Boekhout et al. extended efficient temporal motifs counting algorithms to evolving multilayer network [10]. To the best of our knowledge, there is no earlier study on the random walk sampling of multiplex networks where the graph is not fully accessible and the visit of one layer is feasible upon which the associated nodes of the other layer have been visited before. Especially, it remains untouched when the sampling methods on multiple layers are different and coupled.
1.2 Our Contributions

**Novel Sampling Problem.** The novelty of this work lies in three aspects. To the best of our knowledge, we are the first to investigate the restricted sampling problem in multiplex networks that has ample real-world examples. On the upper layer, the access to the complete raw data is prohibited, but indirect access through APIs is allowed. On the lower layer, the access to a node’s neighboring nodes or edges is possible on when the corresponding node in the upper layer has been visited. This coupled restriction has not been considered previously, and makes the graphlet sampling very tricky meanwhile. Second, we design a novel sampling algorithm that uses the visited Markov states to infer the three-node graphlet concentration with isomorphic state precomputation. Third, for a given budget of sampling steps, we explore the tradeoff between the accuracies of different graphlets when more nodes or edges are sampled at the lower layer network. Owing to the heterogeneous time or economic cost of sampling an edge in different layers, appropriately assigning the sampling budget may benefit the accuracy of specific types of graphlets. We believe that the raised problem will elicit a good many works in new scenarios and new sampling algorithms in multiplex networks.

**Provable Guarantee.** We prove the unbiasedness of our sampling algorithms, and derive an analytic Chernoff-Hoeffding bound on the needed sample size to achieve a certain error. Especially, the theoretical analysis shows how the performance of sampling is influenced by network parameters.

**Extensive Experimental Analysis.** Extensive experiments on real-world and synthetic multiplex networks are conducted to evaluate the accuracy of graphlet concentration. Experimental results confirmed the unbiasedness, accuracy and convergence of the proposed estimators. Our algorithms demonstrate comparable accuracy with the random walk sampling on both layers with no restriction.

The remainder of this article is organized as follows. Section 2 presents the problem formulation. We design a novel sampling algorithm with provable performance in Section 3. Section 4 explores the tradeoff of assigning sampling steps in different layers. Section 5 evaluates the proposed algorithm on synthetic and real-world multiplex networks and Section 6 concludes this work.

2 PROBLEM FORMULATION

In this section, we present the multiplex network model and the theoretic foundations of graph sampling.

2.1 Network Model

A connected complex network is denoted by \( G = (V, E) \) where \( V \) is the set of nodes and \( E \) is the set of undirected edges. We need multiple colors to characterize the relationship of pairwise nodes on multiple layers. Denote by \( V_B \subseteq V \) the set of first layer nodes colored in BLUE and \( V_R \subseteq V \) the set of second layer nodes colored in RED. Similarly, let \( E := \{E_B, E_R, E_C\} \) where \( E_B \) and \( E_R \) are the set of edges in the blue and red graphs, respectively, and \( E_C \) consists of the edges connecting two graphs. An edge in \( E_C \) means that the two nodes can be seen as the same node (e.g., in a two-layer social network, those two nodes correspond to one person). Let \( G_B = (V_B, E_B) \) and \( G_R = (V_R, E_R) \) be the corresponding blue and red graphs. There have been a variety of real-world counterparts regarding this two-layer network model in which we hereby name a few.

- **Cyber-physical social networks.** If edges are interpreted as the friendships between persons, then the corresponding nodes bond with each other in online social networks, physical networks, or both, thus forming a multiplex social network.
A user employs the same account for two different social networks so that many such users link them into one giant two-layer social network.

In the following, for the convenience of narration, we regard two nodes connected by an edge in $E_C$ as identical. That means if $(u_1, u_2)$ is in $E_C$, then we take $u_1$ and $u_2$ as one node $u$. For each edge $(u, v)$, we define the neighbors of an edge (or neighboring edges interchangeably) as the extra edges connecting either node $u$ or $v$ but not both in the whole network. Let $b_u$ be the blue degree of node $u$, which means the number of blue nodes adjacent to $u$, and let $r_u$ be the corresponding red degree of $u$. For the blue edge $(u, v)$, we denote by $b(u, v)$ the number of its blue neighbors and by $r(u, v)$ the number of its red neighbors. There are

$$b(u, v) := b_u + b_v - 2\mathbb{I}_E(u, v),$$
$$r(u, v) := r_u + r_v - 2\mathbb{I}_R(u, v),$$

where $\mathbb{I}_E(u, v)$ is 1 if the red edge $E_R(u, v)$ exists and 0 otherwise. Note that the parameters $b(u, v)$ and $r(u, v)$ will be used to calculate state transition probabilities in the random walk. For example, in Figure 1(a), $b(A, B) = 2 + 2 - 2 = 2$, $r(A, B) = 2 + 1 - 2 = 1$, so that means the blue edge between $A$ and $B$ has 2 blue neighboring edges, while the red edge between $A$ and $B$ has 1 red neighboring edge.

### 2.2 Two-layer Graphlets

Graphlets are defined as small induced subgraphs of a large network, where an induced subgraph means that once some nodes are selected, all the edges between them are selected too. For example, there are two well-known connected three-node graphlets in a single layer network, which are triangle and wedge. Knowing graphlet statistics is of great importance in network science and engineering. Extensive efforts have been devoted to the understanding on how social relationship patterns are formed and evolve in online social networks [17, 26, 31]. However, an important yet largely overlooked problem is to measure graphlet concentration in multiplex networks. The graphlets in the above two-layer networks simultaneously capture the interaction between users in both the online social network and the physical world or another social network.

An induced graph of $G$, $G' = (V', E')$, is a connected subgraph whose vertices and edges are all in $G$, i.e., $E' = \{(u, v) : u, v \in V', (u, v) \in E\}$. Let us define $C^{(k)}$ as the set of all connected and induced subgraphs (CISes) containing $k$ nodes. Consider two graphs $G'_1$ and $G'_2$. If there exists a bijection $\varphi : V'_1 \rightarrow V'_2$ with $(u, v) \in E'_1 \Leftrightarrow (\varphi(u), \varphi(v)) \in E'_2$, then we claim that $G'_1$ and $G'_2$ are isomorphic. The two isomorphic graphlets are deemed as the same type of graphlet. After grouping isomorphic CISes, $C^{(k)}$, is partitioned into $N_k$ classes in which $C^{(k)}_i$ refers to the $i$th type of isomorphic CISes. Different $k$ yields different $N_k$ that grows exponentially with regard to $k$. The number of non-isomorphic classes is 2 when $k$ is 3, and it grows to 6 when $k$ is 4. The non-isomorphic graphlets in our two-layer graph $G$ are more complicated, embracing richer representations among local nodes. Figure 2 illustrates all 16 graphlets with different edge colors.
graphlet consists of three nodes in two layers where two nodes with an inter-layer link are deemed as the same node. An edge colored in **BLUE**+**RED** means that the two nodes are connected at both layers. Note that \( C_1^{(3)} \) and \( C_6^{(3)} \) contain only blue edges, and \( C_{15}^{(3)} \) and \( C_{16}^{(3)} \) contain only red edges. In this article, we focus on the case \( k = 3 \) so that the superscript \( k \) is ignored. Meanwhile, because it is assumed that we only conduct random walk on blue layer, the 15th and 16th graphlets cannot be visited. Thus, we only take the first 14 graphlets into account. The concentration of graphlets is defined as

\[
d_i = \frac{|C_i|}{|C|},
\]

where \( |C_i| \) is the number of type \( i \) CISes and \( |C| \) is the total number of CISes. In the experiment part, we will see that the concentration of graphlets in a network with moderate size (about 1 million edges) may vary in a wide range such as from \( 10^{-1} \) to \( 10^{-6} \).

### 2.3 Two-layer Graphlet Sampling

New challenges arise in the sampling of a multiplex network besides the unavailability of complete network topology. *Not all layers can be sampled in the same way.* For instance, an OSN layer can be queried through a random walk approach. After querying a node or an edge, this approach is able to jump to one of its neighbors. Such a random walk on physical person-to-person networks or privacy-sensitive social networks is either infeasible or costly. For instance, a crawler can visit a person’s direct connections, but not those beyond one-hop. Furthermore, querying the friendship relationships in the physical world has to pay some coupons or needs multiple rounds of interactions. We consider a novel sampling problem in restricted and multiplex networks, in which the first layer allows random walk, while the second layer only supports node sampling from the nodes already visited at the first layer.

**Random Walk Sampling:** A random walk sampling over graph \( G \) is a process that enables the moving from a node or an edge to one of its neighbors chosen uniformly at random and that starts from an initial node and terminates until certain stopping criteria.

**Node/Edge Sampling:** A target fraction of nodes/edges are chosen independently and uniformly at random for inclusion in a graph \( G \), and the attached edges/nodes to these nodes/edges are included to construct the induced subgraph.

In general, sampling algorithms (including the existed ones and the proposed one) sample graphlets from the set of all graphlets appeared in the graph. Hence, the probabilities of being sampled for different graphlets are not the same. We need to “remedy” the unequal probabilities and finally obtain an unbiased estimator. Our graph sampling problem differs from the literature in two aspects. First, the random walk is feasible only at one layer, and the node sampling depends on the result of a random walk. Second, our random walk procedure is mixed up with sampling to obtain some information from the restricted layer. The major notations are summarized in Table 1.

### 3 DESIGN OF SAMPLING ALGORITHMS

In this section, we present a random walk framework for graphlet estimation in restricted two-layer networks (a two-layer network as a typical instance of multiplex network). Both the node-based and the edge-based random walks are investigated. As we mentioned, we assume an unequal access restriction on two layers. To be specific, in this article, including this node by node algorithm and the following algorithms, we assume that the blue layer supports random walk, while the red layer only allows us to do node/edge sampling.
Table 1. Notations

| Symbol | Description |
|--------|-------------|
| $G$    | a two-layer complex network |
| $\mathcal{V}$ | a set of all nodes, including nodes in blue and red levels |
| $\mathcal{E}$ | a set of all edges, including blue and red edges |
| $\mathcal{V}_B$ | a set of nodes in blue level |
| $\mathcal{E}_B$ | a set of blue edges |
| $\mathcal{E}_C$ | a set of edges connecting two levels |
| $|C_i|$ | the exact number of $i_{th}$ graphlet |
| $d_i$ | the exact concentration of $i_{th}$ graphlet |
| $\hat{|C_i|}$ | the estimator of $|C_i|$ |
| $\hat{d}_i$ | the estimator of $d_i$ |
| $X_m$ | one blue edge or one blue node |
| $Y_{m+1}$ | one red edge or one red node, specifically, it is adjacent to $X_m$ |
| $S_I$ | a state in our algorithms consisted of three nodes or two edges |
| $b_{X_m}$ | the number of blue edges/nodes that are adjacent to $X_m$ |
| $r_{X_m}$ | the number of red edges/nodes that are adjacent to $X_m$ |
| $W$ | the state space of our Markov chain |
| $\pi$ | the stationary distribution of our Markov chain |
| $P$ | the state transition matrix of our Markov chain |
| $\tau(\epsilon)$ | the mixing time of our Markov chain |
| $\alpha_i$ | the number of states that are correspond to the $i_{th}$ graphlet |

### 3.1 Node-by-node Random Walk

Now, we describe the first algorithm we proposed, which is the node-by-node random walk (RWNbN). We organize this section in the following order: the thinking behind the algorithm followed by an illustrated toy example showing how this algorithm works, then the formal implementation of the method and the detailed analysis of the method including why it is unbiased and how many steps does it need to attain a given accuracy.

We first give the intuitive thinking behind the proposed algorithm. Recall that a graphlet contains three nodes in one or two layers. It is the same as the case when we want to estimate graphlet concentration in a single layer network. In that situation, we only need to let the random walker keep visiting nodes among the neighbors of the previous node, and providing we get a three-node graphlet, we record the type of it (i.e., triangle or wedge). After many steps, we can obtain a set of sampled graphlets. Some of them are triangles, and the others are wedges. It is just like we take a sampling among all the three-node graphlets, but the sampling probability is not uniform. Then we can take advantage of the concentrations of our sampled graphlets to estimate the true concentrations of the whole graph. However, in our setting, there is a two-layer network with unequal access restrictions, so we choose to keep the main body of the algorithm mentioned above. That is we still mainly do random walk on the available layer (i.e., blue layer) but node/edge sampling on another (i.e., red layer) should also be integrated into the algorithm.

Here is a toy example demonstrating how the algorithm works. The initial 3-tuple is $(A, B, C)$ at Figure 3(a) with all the nodes in blue. In the next step, the random walker picks a red neighbor of $C$, that is $A$, with probability $\frac{1}{4}$ (because $b_C = 3$ and $r_C = 1$). This makes the random walker enter the red layer as Figure 3(b). We denote $A_y$ as the node $A$ at the red layer. The 3-tuple now turns into $(B, C, A_y)$. At the third step, owing to the restriction of accessing $A_y$’s neighbors, the random walker must return to the blue layer, yielding a new tuple $(B, C, D)$ shown in Figure 3(c) with probability $\frac{1}{3}$. 

ACM Transactions on the Web, Vol. 15, No. 4, Article 18. Publication date: June 2021.
The random walk on graph \( \mathcal{G} \) with probability \( S \) yields state at the red layer if it exists. Define a 3-tuple \( b \) in our sampling. If the third node is at the red layer via node sampling, a state cannot include two red nodes because only one-hop neighbor at the red layer is allowed to visit. Define the state space as \( \mathcal{S} \). Our restricted random walk process in a two-layer network is an irreducible Markov chain. For any two states \( (X_m, X_{m+1}, X_{m+2}) \) and \( (X_n, X_{n+1}, X_{n+2}) \), they can reach.

Now, we provide the detailed implementation of the algorithm. Formally, a node \( u \) is visited at the current step, the node-by-node random walk (RWNbN) drives the sampler to visit the node \( v \) at the next step where \( v \) is a neighbor of \( u \). Let \( X_m \) be the \( m \)th node at the blue layer and let \( Y_{m+1} \) be the randomly sampled neighbor of \( X_m \) at the red layer if it exists. Define a 3-tuple \( S_t \) as a state of random walk that consists of the three most recently traversed nodes at the \( t \)th step. Note that three visited nodes can induce a three-node graphlet. For clarity, there is an unbiased estimator and the error bound of it.

Table 2. State Transition Matrix of Node-by-node RW

| Current | Next | \( (X_m, X_{m+1}, X_{m+2}) \) | \( (X_m, X_{m+1}, Y_{m+2}) \) |
|---------|------|-------------------------------|-------------------------------|
| \( (X_m, X_{m+1}, Y_{m+2}) \) | \( \frac{1}{b_{X_{m+1}}} \) | \( \frac{1}{r_{X_{m+1}}+b_{X_{m+1}}} \) | \( \frac{1}{r_{X_{m+1}}+b_{X_{m+1}}} \) |
| \( (X_{m-1}, X_m, X_{m+1}) \) | \( \frac{1}{r_{X_{m+1}}+b_{X_{m+1}}} \) | \( \frac{1}{r_{X_{m+1}}+b_{X_{m+1}}} \) | \( \frac{1}{r_{X_{m+1}}+b_{X_{m+1}}} \) |

The random walk on a graph constitutes an ergodic Markov process. We next derive the state transition probabilities of the restricted random walk. At a given time step \( S_t = (X_{m-1}, X_m, X_{m+1}) \), we randomly pick a node among \( b_{X_{m+1}} \) blue and \( r_{X_{m+1}} \) red ones to visit. With probability \( \frac{1}{r_{X_{m+1}}+b_{X_{m+1}}} \), the random walk moves to a blue node \( X_{m+2} \) and yields state \( S_{t+1} = (X_m, X_{m+1}, X_{m+2}) \) at the next time step, and with probability \( \frac{1}{r_{X_{m+1}}+b_{X_{m+1}}} \) it moves to an adjacent red node and reaches the state \( S_{t+1} = (X_m, X_{m+1}, Y_{m+2}) \). At a given state \( S_t = (X_m, X_{m+1}, Y_{m+2}) \), because the red graph does not allow random walk, a blue node adjacent to \( X_{m+1} \), namely \( X_{m+2} \), is chosen uniformly so that the state at the \( (t+1) \)th step is expressed as \( S_{t+1} = (X_m, X_{m+1}, X_{m+2}) \) with probability \( \frac{1}{b_{X_{m+1}}} \). We should notice that \( Y_{m+2} \) is not necessarily the corresponding node of \( X_{m+2} \) in the red layer. We summarize the state transition probabilities in Table 2.

In the following parts, we discuss some property of the algorithm including how we can get an unbiased estimator and the error bound of it.

3.1.1 Unbiased Estimator. The random walk on graph \( \mathcal{G} \) does not automatically reveal the graphlet concentration. We need to derive the stationary distribution of our Markov chain and map Markovian states into graphlets.

Stationary Distribution. Our restricted random walk process in a two-layer network is an irreducible Markov chain. For any two states \( (X_m, X_{m+1}, X_{m+2}) \) and \( (X_n, X_{n+1}, X_{n+2}) \), they can reach.
an indicator function of an arbitrary Markovian state $\alpha_u$.

$E$ is the edge set of blue layer, and $\mathcal{V}_B$ is the node set of blue layer. The detailed analysis can be found in Appendix A. One can observe that $\pi(X_m, X_{m+1}, X_{m+2})$ is larger than $\pi(X_m, X_{m+1}, Y_{m+2})$ because the sampling of a red edge is possible only when the corresponding blue edge has been sampled by random walk.

**Isomorphic State Precomputation.** We hereby provide a mapping method from traversed states to three-node graphlets. A key observation is that a graphlet corresponds to several Markovian states in the random walk. Hence, before estimating the graphlet concentration, we need to figure out the relationship between states and graphlets.

Note that any two states forming the same graphlet are called isomorphic states. The isomorphic state coefficient refers to the number of isomorphic states in correspondence with the same graphlet. The isomorphic state coefficient can be computed in advance. Its precomputation isomorphic state coefficient refers to the number of isomorphic states in correspondence with the same graphlet. The isomorphic state coefficient can be computed in advance. Its precomputation

\[
\begin{align*}
\pi(X_m, X_{m+1}, X_{m+2}) &= \frac{1}{M_{X_{m+1}}} \\
\pi(X_m, X_{m+1}, Y_{m+2}) &= \frac{1}{M(r_{X_{m+1}} + b_{X_{m+1}})}
\end{align*}
\]

where $M = 2|E_B| + \sum_{v \in \mathcal{V}_B} b_v r_v$. $E_B$ is the edge set of blue layer, and $\mathcal{V}_B$ is the node set of blue layer.

**Example 1.** If we perform a restricted random walk on the sixth graphlet consisting of three blue edges and denote three nodes by $u, v,$ and $w$. Different traverse trajectories result in six states respectively: $(u, v, w), (u, w, v), (v, u, w), (v, w, u), (w, u, v)$, and $(w, v, u)$. The isomorphic state coefficient is 6 accordingly.

**Example 2.** If a restricted random walk happens on the 10th graphlet, then there are only two isomorphic states: $(u, v)$ and $(v, u, w)$, where two end nodes of blue edge are denoted by $u, v$ and the third node is $w$. Since only the node sampling on the red graph is allowed, two blue nodes should be visited first in our restricted random walk on the blue graph, leading to the isomorphic state coefficient of two.

We use $\alpha_i$ to denote the isomorphic state coefficient of the $i$th graphlet, and summarize all the coefficients of three-node graphlets in Table 3. Intuitively, a larger coefficient means more blue edges in the blue graph that supports random walk.

**Estimator.** Denote by $g_i$ an indicator function of an arbitrary Markovian state $S \in \mathcal{W}$ that has

\[
g_i(S) = \begin{cases} 
1 & S \text{ induces the } i\text{th graphlet} \\
0 & \text{otherwise}
\end{cases}
\]
Summing all the possible states and mapping them into different graphlets, there exists \( \sum_{S \in W} g_i(S) = \alpha_l |C_i| \). The parameter \( \alpha_l \) means that a graphlet has multiple isomorphic copies in the state space. The random walk operations harvest \( n \) states, \( \{S_i\}_{i=1}^n \subset W \).

The unbiased estimation is put on the foundation of **Strong Law of Large Numbers (SLLN)** [18]. If an irreducible Markov chain has finite state space \( W \) with a stationary distribution \( \pi \), and there exists a function of the state \( S, h(S) : W \rightarrow \mathbb{R} \), then the expectation of \( h(S) \) at all the states can be defined as

\[
\mu = \mathbb{E}_\pi[h(S)] = \sum_{S \in W} h(S)\pi(S).
\]  

Given \( n \) states \( \{S_j\}_{j=1}^n \), SLLN provides the following theorem.

**Theorem 3.1** [18]. \( \frac{1}{n} \sum_{j=1}^{n} h(S_j) \xrightarrow{a.s.} \mathbb{E}_\pi[h] \) as \( n \rightarrow \infty \).

We can obtain

\[
\frac{1}{n} \sum_{j=1}^{n} \frac{g_i(S_j)}{\pi(S_j)} \xrightarrow{a.s.} \mathbb{E}_\pi \frac{g_i(S)}{\pi(S)} = \sum_{S \in W} \frac{g_i(S)}{\pi(S)} \times \pi(S),
\]

and the last term is actually \( \sum_{S \in W} g_i(S) = \alpha_l |C_i| \). Thus, \( |C_i| = \frac{1}{n} \sum_{j=1}^{n} \frac{g_i(S_j)}{\alpha_l \pi(S_j)} \) is an unbiased estimator of the number of the \( i \)th graphlet. Our goal is to compute the concentration of the \( i \)th graphlet, denoted by \( d_i \), \( d_i = \frac{|C_i|}{\sum_{j=1}^{n} |C_j|} \). Because \( |C_i| \) estimates \( |C_i| \) without bias, the unbiased estimator of the concentration of the \( i \)th graphlet is given by

\[
\hat{d}_i = \frac{|\hat{C}_i|}{\sum_{j=1}^{n} |\hat{C}_j|} = \frac{\sum_{j=1}^{n} \frac{g_i(S_j)}{\alpha_l \pi(S_j)}}{\sum_{j=1}^{n} \sum_{i=1}^{14} \frac{g_i(S_j)}{\alpha_l \pi(S_j)}}.
\]

Note that the stationary distribution \( \pi(S_j) \) contains an unknown variable \( M \). This variable is pertinent to the global information of the two-layer graph, and cannot be known as **a priori**. Fortunately, \( M \) appears at both the numerator and denominator so that it is canceled out at the computation of \( \hat{d}_i \). Algorithm 1 specifies the procedure of obtaining an unbiased estimator. When we calculate \( \pi(S_j) \), \( M \) is set aside actually.

### 3.1.2 Error Bound

We next analyze the error bound of our random walk algorithm, i.e., how many steps of random walk are required to guarantee a given estimation accuracy.

**Theorem 3.2.** [15] Let \( MC \) be an ergodic Markov chain with state space \( W \) and stationary distribution \( \pi \). Let \( \tau(\zeta) \) be its \( \zeta \) – mixing time for \( \zeta \leq \frac{1}{8} \). Let \( \{S_j\}_{j=1}^n \) denote an \( n \)-step random walk on \( MC \) starting from an initial distribution \( \phi \) on \( W \), i.e., \( S_1 \leftarrow \phi \). For every \( i \in \{1, 2, \ldots, n\} \), let \( f_i : W \rightarrow [0, 1] \) be a weight function at step \( i \) such that the expected weight \( \mathbb{E}_\pi[f_i] = \mu \) for all \( i \). Define the total weight of the walk \( \{S_j\}_{j=1}^n \) by \( Z = \sum_{i=1}^{n} f_i(S_i) \). There exists a constant \( c \) (which is independent of \( \mu, \epsilon \) and \( \zeta \) ) such that

\[
Pr\{|\frac{Z}{n} - \mu| > \epsilon \} \leq c||\phi||_\pi \exp(-\epsilon^2 \mu n/(72\tau))
\]

for \( 0 < \epsilon < 1 \).

Define \( H = \max_{S \in W} \frac{1}{\pi(S)} \) and \( \alpha_{min} = \min_{1 \leq i \leq 14} \alpha_i \). Denote by \( \tau(\zeta) \) the mixing time of our Markov chain and denote by \( \phi \) the initial distribution of the visited states with

\[
||\phi||_\pi = \sum_{S \in W} \frac{\phi^2(S)}{\pi(S)}.
\]
ALGORITHM 1: Node-by-node Random Walk: RWNbN

Input: two-layer network: \( G \), number of samples: \( n \)
Output: estimated graphlet concentration \( \hat{d}_i \)

\(|\hat{C}_i| = 0, \ \forall i; \ \text{Random walk counter } t = 1\)
Randomly pick a valid initial state \( S_1 = (X_1, X_2, X_3) // X_1 \) and \( X_2 \) and \( X_3 \) are nodes in blue level that induce a three-node graphlet

while \( t \leq n \) do

\( i = \) type of the subgraph induced by \( S_t \)
\( |\hat{C}_i| = |\hat{C}_i| + \frac{1}{a_i \pi(3t)} \)
\( t = t + 1 \)

if \( S_t \) is in the form \( (X_m, X_{m+1}, X_{m+2}) \) then

Uniformly pick a node in the blue layer adjacent to \( X_{m+2} \) with probability \( \frac{1}{r_{X_{m+2}} + b_{X_{m+2}}} \), or

a node in the red layer adjacent to node \( X_{m+2} \) with probability \( \frac{1}{r_{X_{m+2}} + b_{X_{m+2}}} \)

else

Uniformly pick a node in the blue layer that is a neighbor of \( X_{m+1} \)

end if

\( \hat{d}_i = \frac{|\hat{C}_i|}{\sum_{j=1}^{14} |\hat{C}_j|}, i \in \{1, 2, \ldots, 14\} \)

Let \( \Lambda_i = \min\{a_i |C_i|, a_{min} |C|\}, \) where \( |C| = \sum_{i=1}^{14} |C_i| \). The subscript \( i \) is dropped when we do not specify the type of the graphlet. The following theorem guarantees that the relative error is below a sufficiently small \( \epsilon \) with a high probability as the number of random walk steps is greater than a certain threshold.

**Theorem 3.3.** \( \forall 0 < \delta < 1, \ \exists \ \text{constant } \xi, \ \text{such that, when } n \geq \xi \frac{H}{\Lambda_i} \frac{e}{\epsilon^2} \ln \left( \frac{||\phi||_b}{\delta} \right), \) we have

\[
Pr((1 - \epsilon)d_i \leq \hat{d}_i \leq (1 + \epsilon)d_i) > 1 - \delta.
\]  

(7)

The detailed proofs can be found in Appendix B. The parameters \( \tau, H \) and \( \Lambda_i \) in the inequality are with regard to the threshold \( n \) that influence the convergence rate of a sampling algorithm. For the mixing time \( \tau \), in Reference [39], the authors measured the mixing time of several social networks. They provided a theoretical bound of the mixing time. Specifically, if \( k \) is the dimension of transition matrix of random walk (i.e., the number of possible states), and \( \mu \) is the second largest eigenvalue of transition matrix, then we have \( \tau(\epsilon) \leq \frac{\log(k) + \log(\frac{1}{\epsilon})}{1 - \mu} \). Besides that, in the experiment part, they show that the average mixing time is better than the worst-case mixing time. For example, when we take \( \epsilon = 0.1 \), the mixing time of several datasets in their article (those datasets have various sizes that can cover the datasets we use) are around \( 10^2 \) or \( 10^3 \). For constant \( H \), it is determined by the specific sampling method. Take the RWNbN as an example. \( H = M \max(r_{X_{m+1}} + b_{X_{m+1}}) \), where \( M \) is a constant mentioned in part 3.1.1. So, if we define the degree of a node as the sum of its blue degree and red degree, then \( H \) is the largest degree in the graph times \( M \). The \( \Lambda \) depends on the graphlet count of a dataset. Thus, the sampling of a dense graphlet needs fewer random walk steps to converge, while that of a rare one demands many more steps.

### 3.2 Edge-by-edge Random Walk

Similarly, we can induce three-node graphlets by packing two sampled edges as a two-tuple. After understanding how does the node-by-node random walk operates, the thinking of edge-by-edge random walk is quite straightforward. Suppose that an edge \((u, v)\) is visited at the current step,
the edge-by-edge random walk drives the sampler to visit the edge \((u', v')\) at the next step where either \(u'\) or \(v'\) remains unaltered, i.e., \(u = u'\) or \(v = v'\) but not both.

Now, we demonstrate the implementation details. For convenience, we use the same notation as the previous algorithm. We let \(X_m\) be the \(m\)th edge of the blue graph and let \(Y_{m+1}\) be the neighbor of the corresponding edge of the red graph if it exists. With certain abuse of notations, we define a two-tuple \(S_t\) as a state of random walk that consists of the two most recently traversed edges. There is \(S_t := (X_m, X_{m+1})\) if both edges are in blue, and \(S_t := (X_m, Y_{m+1})\) if a red edge is just visited via edge sampling. Our design of state resounds to two important properties. One is that the two edges are adjacent to each other, the other is that an edge at the red graph can only be reached through its neighbor at the blue graph, and must return to the blue graph afterward. Let the state space be \(W = \{(X_m, X_{m+1}) | X_m \cap X_{m+1} \neq \emptyset, \forall m\} \cup \{(X_m, Y_{m+1}) | X_m \cap Y_{m+1} \neq \emptyset, \forall m\}\).

To better understand the operations of edge-by-edge random walk, we illustrate the procedure in Figure 4. A hollow line denotes an edge at the current state, e.g., \((AB, BC)\) in Figure 4(a). If the random walker enters the red layer via node \(C\) and samples the red edge \(CyAu\), then the new state turns into \((BC, CyAu)\) in Figure 4(b). Due to the restriction at the red graph, the random walker returns to the blue layer (i.e., to node \(C\)) and visits a new blue edge \(CD\) that yields a new state \((BC, CD)\) in Figure 4(c).

We next derive the state transition probabilities of the restricted random walk. At a given state \(S_t := (X_m, X_{m+1})\), we randomly pick an edge among \(b_{X_{m+1}}\) blue and \(r_{X_{m+1}}\) red ones to visit. With probability \(\frac{1}{r_{X_{m+1}} + b_{X_{m+1}}}\) the random walker moves to a blue edge \(X_{m+2}\) and yields state \(S_{t+1} := (X_{m+1}, X_{m+2})\), and with probability \(\frac{1}{r_{X_{m+1}} + b_{X_{m+1}}}\) it moves to an adjacent red edge and yields state \(S_{t+1} := (X_m, Y_{m+1})\). If the current state is \(S_t = (X_m, Y_{m+1})\), because the red graph does not allow random walk, then a blue edge adjacent to \(X_m\), namely \(X_{m+1}\), is chosen uniformly so that the new state is expressed as \(S_{t+1} := (X_m, X_{m+1})\) with probability \(\frac{1}{b_{X_m}}\). For clarity, we summarize the state transition probabilities in Table 4.

**Stationary Distribution.** We directly show the stationary distribution \(\pi\) as the following:

\[
\begin{align*}
\pi(X_m, X_{m+1}) & = \frac{1}{M} b_{X_m} \\
\pi(X_m, Y_{m+1}) & = \frac{1}{M} \frac{b_{X_m}}{r_{X_{m+1}} + b_{X_m}}.
\end{align*}
\]
Table 5. State Transition Matrix of RWOMRN

| Current state               | Next state                  | \( (X_m, X_{m+1}, X_{m+2}) \) | \( (X_m, X_{m+1}, Y_{m+2}) \) | \( (X_{m+1}, Y_{m+2}, Y_{m+3}) \) |
|----------------------------|-----------------------------|-------------------------------|------------------------------|----------------------------------|
| \( (X_{m-1}, X_m, X_{m+1}) \) | \( \frac{1}{r_{X_{m+1}}+b_{X_{m+1}}} \) | \( \frac{1}{r_{X_{m+1}}+b_{X_{m+1}}} \) | 0                             |                                  |
| \( (X_m, X_{m+1}, Y_{m+2}) \) | 0                           | 0                             | \( \frac{1}{r_{Y_{m+2}}} \)  |                                  |
| \( (X_{m+1}, Y_{m+2}, Y_{m+3}) \) | \( \frac{1}{b_{X_{m+1}}} \) | 0                             | 0                             | 0                                |

where \( M = 2|E_B| + \sum_{v \in V_B} \frac{b_v r_v}{b_v + r_v} \). The detailed analysis is very similar to the proof of node by node random walk in Appendix A. One can observe that \( \pi(X_m, X_{m+1}) \) is larger than \( \pi(X_m, Y_{m+1}) \) because the sampling of a red edge is possible only when the corresponding blue edge has been sampled by a random walk. Because the mapping from two edges to a three-node graphlet is the same as that from three nodes to a three-node graphlet, the precomputations of isomorphic are the same. We directly show the procedure of edge-by-edge random walk (RWEbE) in Algorithm 2.

The connection between RWEbE and RWNbN is the following. For a given graph, we can create a new relationship graph in which each new node refers to an edge in the original graph, and two nodes are connected if their corresponding edges are adjacent in the original graph. Therefore, RWEbE is actually the node-by-node random walk on this relationship graph. The relationship graph is of larger size, since there are more edges than nodes in the original graph, and the average degree is expected to be larger. In a general single layer graph, it is hard to tell whether RWEbE or RWNbN is more accurate on a specific graphlet. When constructing a relationship graph on the two-layer restricted network, we magnify the blue layer much more than the red layer, since the access to red nodes is limited to be within one-hop. Hence, the chance of visiting the graphlets with more blue nodes is higher, and the chance of meeting those with more red nodes is lower. The accuracy of graphlet concentration is influenced accordingly. The convergence property of RWEbE is guaranteed by Theorem 3.3. The convergence rate largely depends on how rare a graphlet is in a network.

Compared to RWNbN, RWEbE has better expandability. When we want to consider the concentration of four-node graphlet, RWNbN may not work. For example, for 3-star, i.e., one node connects the other three nodes with degree 1, it can be easily found that a four-tuple of nodes obtained by node-by-node random walk cannot induce that a graphlet. However, we can use three edges to induce it. So, if someone want to explore graphlets with more than three nodes, RWEbE can be seen as a building block.

In the above section, we proposed the main algorithms in our article that are the node-by-node random walk and the edge-by-edge random walk. Also, based on these algorithms, we provide detailed introduction, illustration and analysis. In the following parts, we will introduce two extensions. Besides the implementation details, the intuition behind those methods and the theoretical analysis are similar, so we will describe them in short.

4 SAMPLING BUDGET ADJUSTMENT

An interesting property of sampling graphlets in two-layer networks is that “all graphlets in different layers are not sampled equally.” For instance, the random walker enters the red layer through the blue layer so that the latter may be visited more frequently. Even though our proposed sampling algorithms are unbiased with provable error bounds, one can observe that the estimation
Algorithm 2: Edge-by-edge Random Walk: RWEbE

**Input:** two-layer network: $G$, random walk steps: $n$

**Output:** estimated graphlet concentration: $\hat{d}_i$

$|\hat{C}_i| = 0$, $\forall i$; Random walk counter $t = 1$

Randomly pick an initial state $S_1 = (X_1, X_2)$

// $X_1$ and $X_2$ are blue edges with one common node

while $t \leq n$

$i = \text{type of the subgraph induced by } S_t$

$|\hat{C}_i| = |\hat{C}_i| + \frac{1}{a_i \pi(S_t)}$

$t = t + 1$

if $S_t$ is in the form $(X_m, X_{m+1})$

Uniformly pick a blue edge adjacent to $X_{m+1}$ with probability $\frac{1}{r_{X_{m+1}} + b_{X_{m+1}}}$ or a red edge adjacent to $X_{m+1}$ with probability $\frac{1}{r_{X_{m+1}} + b_{X_{m+1}}}$

else

Uniformly pick a blue edge adjacent to $X_m$

end if

Update $S_t$

end while

$\hat{d}_i = \frac{|\hat{C}_i|}{\sum_{j=1}^{14} |\hat{C}_j|}$, $i \in \{1, 2, \ldots, 14\}$

Accuracy of the graphlets with more blue edges is better off than that with more red edges. Given the same total random walk steps, one can choose to distribute more steps to a particular layer so as to improve the sampling accuracy of some graphlets, whereas at the cost of degraded accuracy of the others. This is a new phenomenon that has not appeared in traditional single-layer networks. In this section, we consider the distribution of the sampling budget in two layers and explore the tradeoff on the accuracy of different graphlets.

4.1 Sampling More Graphlets with More Red Edges

Here, we show the intuition of the algorithm proposed in this section. Previously, when the sampling is at a state $(X_m, X_{m+1}, Y_{m+2})$, it will go back to the blue layer and transit to a new state $(X_m, X_{m+1}, X_{m+2})$. In the new setting, the sampler can visit one more red node, i.e., transiting to the state $(X_{m+1}, Y_{m+2}, Y_{m+3})$, before returning to the blue layer. In other words, when there is an opportunity to enter the red network, the red nodes are sampled twice. Our sampling framework applies to this new situation. That means the stationary distribution, state transition matrix, isomorphism coefficients and the corresponding sampling algorithm namely RWOMRN (OMRN means one more red node) should be modified, but all the proof of the unbiasedness and error bound remains same. We summarize the state transition matrix and the isomorphic state coefficient of RWOMRN at Table 5 and Table 6, respectively.

**Stationary Distributions.** Our first step is to compute the stationary distribution of all the Markovian states.

$$
\begin{align*}
\pi(X_m, X_{m+1}, X_{m+2}) &= \frac{1}{MB_{X_{m+1}}} \\
\pi(X_m, X_{m+1}, Y_{m+2}) &= \frac{1}{M(r_{X_{m+1}} + b_{X_{m+1}})} \\
\pi(X_m, Y_{m+1}, Y_{m+2}) &= \frac{1}{M(r_{X_{m+1}} + b_{X_{m+1}})}
\end{align*}
$$

where $M = \sum_{v \in V_B} \frac{b_v - r_v + 3b_v r_v}{r_v}$, $V_B$ is the node set of blue level.
State transition matrix.
The state transition matrix is different from what appears before. Now, when we are in the state \((X_m, X_{m+1}, Y_{m+2})\), in next step, we must explore one more red node and obtain a new state with two red nodes and the probability is \(\frac{1}{r_{m+2}}\). Note that the denominator cannot be zero, because \(Y_{m+2}\) has at least one red neighbor that is the corresponding node of \(X_{m+1}\) on the red layer. When we are in a state \((X_{m+1}, Y_{m+2}, Y_{m+3})\), in next step we must return to blue layer by the mechanism of the algorithm. Consequently, two neighbors of \(X_{m+1}\) are picked randomly, and we attain a state with three blue nodes.

Isomorphism Coefficients.
The result is different from the coefficient for \(RWNbN\). For example, for the fifth graphlet, assume those three nodes on the blue level are \(u, v, w\) from the left side to the right side and their corresponding nodes on the red level are \(u', v', w'\) respectively. Under \(RWNbN\), there are four ways to obtain this graphlet, \((u, v, w)\), \((w, v, u)\), \((u, v, w')\), \((w, v, u')\). But under \(RWOMRN\), we have two more ways \((u, v', w')\), \((w, v', u')\) because we change the random walk mechanism and allow the random walker to sample twice on the red level.

**Algorithm 3:** Sampling One More Red Node: \(RWOMRN\)

**Input:** two-layer network \(G\), random walk steps \(n\)

**Output:** graphlets concentration estimation \(\hat{d}_i\)

|\(\hat{C}_i\) | 0, \(\forall i\); Random walk counter \(t = 1\) |
|---|---|
|Randomly pick a valid initial state \(S_1 = (X_1, X_2, X_3)\) // \(X1\) and \(X2\) and \(X3\) are blue nodes that induce a three-node graphlet |
|while \(t \leq n\) do |
| \(i = \) type ID of induced subgraph of \(S_t\) |
| \(|\hat{C}_i| = |\hat{C}_i| + \frac{1}{\alpha_t \pi(S_t)}\) |
| \(t = t + 1\) |
|if \(S_t\) is in the form \((X_m, X_{m+1}, X_{m+2})\) then |
| Uniformly pick a blue node that is adjacent to \(X_{m+2}\) by chance \(\frac{1}{r_{X_{m+2}} + b_{X_{m+2}}}\), then obtain the next state |
| Or uniformly pick a red node that is adjacent to \(X_{m+2}\) by chance \(\frac{1}{r_{X_{m+2}} + b_{X_{m+2}}}\), then obtain the next state |
|else if \(S_t\) is in the form \((X_m, X_{m+1}, Y_{m+2})\) then |
| Uniformly pick a red node that is adjacent to \(Y_{m+2}\) in red level, then obtain the next state |
|else |
| \(S_t\) is in the form \((X_m, Y_{m+1}, Y_{m+2})\) |
| Uniformly pick 2 blue nodes \(X_{m-1}\) and \(X_{m+1}\) that are adjacent to \(X_m\), and obtain the next state |
|end if |
|end while |

\[\hat{d}_i = \frac{|\hat{C}_i|}{\sum_{j=1}^{14} |\hat{C}_j|}, \quad i \in \{1, 2, \ldots, 14\}\]

4.2 Mixed Algorithm

Previously, we have already shown the algorithm of sampling one node in red level and sampling two nodes in red level. The latter has better performance on those graphlets that have more red edges, but the former is better on graphlets with more blue edges. Actually, given a total of \(n\) sampling steps, if more red nodes are sampled, the graphlets with more red edges can be estimated
more accurately. However, the estimation accuracy of the graphlets with more blue edges degrades. Consequently, we obtain a balance between those two algorithms. Because we want to estimate concentration, so roughly speaking, making the error some kind of evenly can obtain a better estimation of concentration.

So, when in the state \((X_m, X_{m+1}, Y_{m+2})\), instead of directly sampling a red node that is neighbor of \(Y_{m+2}\), we choose to do so by probability, and we can also return to blue level by probability. Consequently, a balance between the two former algorithms is achieved. Compared with two former algorithms, we sample a moderate number of red nodes as well as blue nodes given a fixed number of sampling steps. We summarize the the isomorphic state coefficient and the state transition matrix at the Table 7 and Table 8, respectively.

We show the algorithm in Algorithm 4.

Stationary distribution and Coefficient. Similarly, we directly show the stationary distributions of our state space and coefficients.

\[
\begin{align*}
\pi(X_m, X_{m+1}, X_{m+2}) &= \frac{1}{MB_{X_{m+1}}} \\
\pi(X_m, X_{m+1}, Y_{m+2}) &= \frac{M(X_{m+1} + bX_{m+1})}{bX_m} \\
\pi(X_m, Y_{m+1}, Y_{m+2}) &= \frac{bY_{m+2}}{M(X_{m} + bX_{m})(Y_{m+1} + bX_{m})}
\end{align*}
\]

where

\[
M = 2|E_B| + \sum_{v \in V_B} \frac{b_v r_v}{b_v + r_v} + \sum_{v \in V_B} \sum_{u \in RN(v)} \frac{b_v r_u}{(b_u + r_u)(r_u + b_v)}
\]

\(RN(v)\) denotes the set of red neighbors of \(v\).

State transition matrix

Here, different from the state transition matrix of RWOMRN, when in a state \((X_m, X_{m+1}, Y_{m+2})\), it is possible to explore one more red node, but we may also directly return to the blue layer. The possible candidates are all blue neighbors of \(X_{m+1}\) and all red neighbors of \(Y_{m+2}\), so the denominator is \(bX_{m+1} + rY_{m+2}\).

In the above section, we introduced two extensions as a complement of the two algorithms mentioned previously. We show that if we can do one more node/edge sampling on the red layer, then graphlets with more red edges may be estimated more accurately. However, given the fixed random walk steps, there is a tradeoff between sampling blue edges or red edges. So, the algorithms mentioned in this section actually provide more options for one who cares more about the estimation accuracy of graphlets with more red edges.

5 EXPERIMENTAL EVALUATION

We evaluate the performance of the proposed algorithms (RWNbN, RWEbE, RWOMRN, and RWMix) on a set of multiplex networks. There are two parts in the experiment section. This first one is qualitative analysis, where we want to figure out what factors have an influence on the performance of our estimators. For simplicity, we only consider two dimensions, one is the type of the network, another one is the way that two networks are connected. In this part, we use synthetic networks because they are easy to control. The second part is quantitative analysis, where we want to evaluate the performances of our estimators.

The baseline algorithm is the random walk that can traverse both layers without restrictions (RWNb), and is thus statistically more accurate than the proposed algorithms. We claim that our proposed algorithms are effective if they yield the comparable accuracy as the baseline. The algorithms are implemented in Python and we run experiments on a Linux machine with Intel 3.70 GHz CPU and 8G memory. Our purpose is to answer the following questions:

ACM Transactions on the Web, Vol. 15, No. 4, Article 18. Publication date: June 2021.
**ALGORITHM 4:** Mixed Algorithm : RWMix

**Input:** two-layer network $G$, random walk steps $n$

**Output:** graphlets concentration estimation $\hat{d}_i$

$|C_i| = 0, \forall i$; Random walk counter $t = 1$

Randomly pick a valid initial state $S_1 = (X_1, X_2, X_3)$ // $X_1$ and $X_2$ and $X_3$ are blue nodes that induce a three-node graphlet

while $t \leq n$

i = type ID of induced subgraph of $S_t$

$|\hat{C}_i| = |\hat{C}_{i-1}| + \frac{1}{a_j(\pi(S_t))}$

$t = t + 1$

if $S_t$ is in the form $(X_m, X_{m+1}, X_{m+2})$ then

Uniformly pick a blue node that is adjacent to $X_{m+2}$ by chance $\frac{1}{r_{X_{m+2}} + b_{X_{m+2}}}$, then obtain the next state

Or uniformly pick a red node that is adjacent to $X_{m+2}$ by chance $\frac{1}{r_{X_{m+2}} + b_{X_{m+2}}}$, then obtain the next state

else if $S_t$ is in the form $(X_m, X_{m+1}, Y_{m+2})$ then

Uniformly pick a red node that is adjacent to $Y_{m+2}$ in red level by chance $\frac{1}{r_{Y_{m+2}} + b_{Y_{m+2}}}$, then obtain the next state

Or uniformly pick a blue node that is adjacent to $X_{m+1}$ by chance $\frac{1}{r_{Y_{m+2}} + b_{Y_{m+2}}}$, then obtain the next state

else

$(S_t$ is in the form $(X_m, Y_{m+1}, Y_{m+2})$)

Uniformly pick 2 blue nodes $X_{m-1}$ and $X_{m+1}$ that are adjacent to $X_m$, and obtain the next state

end if

end while

$\hat{d}_i = \frac{|\hat{C}_i|}{\sum_{j=1}^{14} |C_j|}, i \in \{1, 2, \ldots, 14\}$

- Q1: How does the heterogeneity in the structure of different layers influence the accuracy of sampling?
- Q2: How accurate are our sampling algorithms in synthetic and real-world two-layer graphs?
- Q3: Can we balance the sampling accuracies of different types of graphlets?

### 5.1 Experimental Setup

We concatenate two single-layer networks into a two-layer multiplex network under two scenarios: (i) both layers are synthetic networks and (ii) one layer is from the real-world and the other layer is synthetic. Besides that, we also find a real-world two-layer network ff-tw [13, 38] consisted of two social media (Friendfeed and Twitter), which is suitable for our setting. We subsample it to get a moderate connected subgraph with about 1 million edges. With the synthetic two-layer networks, we are able to qualitatively explore the relationship between the sampling accuracy and the network structure; with the real-world and synthetic OSNs, we can evaluate the accuracy of our algorithms in the wild.

**Error Metrics.** We consider two commonly used metrics to characterize the errors of graph sampling.
Table 6. Coefficient $\alpha_i$

| $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | $\alpha_4$ | $\alpha_5$ | $\alpha_6$ | $\alpha_7$ |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 2         | 1         | 3         | 3         | 6         | 6         | 4         |
| $\alpha_8$ | $\alpha_9$ | $\alpha_{10}$ | $\alpha_{11}$ | $\alpha_{12}$ | $\alpha_{13}$ | $\alpha_{14}$ |
| 4         | 8         | 8         | 7         | 12        | 12        | 18        |

- **Mean of relative error (MRE)** is defined as: $E[|\hat{C}_i| - |C_i|]/|C_i|$ over 1,000 independent runs. It is used to measure the closeness of estimation to the ground truth. If our estimators have a low MRE, then that means the estimators have a low relative error in average.

- **Normalized root mean square error (NRMSE)** is defined as:

\[
NRMSE(|\hat{C}_i|) = \sqrt{\frac{E[(|\hat{C}_i| - |C_i|)^2]}{|C_i|}} = \sqrt{\frac{\text{Var}[|\hat{C}_i|] + E[(|\hat{C}_i| - |C_i|)^2]}{|C_i|}}.
\] (10)

It is used to measure the variance and bias of the estimators jointly. Both of them are very important to characterize the performance of our method. If the estimators we proposed have a low NRMSE, then that means the expectations of the estimators are very close to the ground truth and also they have a low variance.

The number of random walk steps is set to 20k that only visits a small fraction of the nodes. The number of independent runs is set to 1,000 to show the convergence of NRMSE as the number of random walks progresses.

### 5.2 Experiment Results

Qualitative and quantitative analyses are presented in this section. The former is based on experiments on synthetic datasets while the latter corresponds to the real and synthetic datasets.

**Qualitative analysis on synthetic graphs.**

In an attempt to study the impact of various factors on the sampling performance, we run experiments on several synthetic multiplex networks with two generated layers. Generally speaking, we focus on two dimensions. One is the type of network (for example, there are several famous types of network, like small-world network or scale-free network). We want to know whether the types of networks have notable influences on the sampling performance. Another dimension is the way that the two layers are connected. The combinations of two layers can be numerous, give the choice of network types and their inter-connecting ways. For simplicity, we choose three famous types of networks and three methods to connect two layers. So, finally, we obtain nine synthetic two-layer networks.

The blue layer is generated from three categories: **Erdos-Renyi (ER)** random graph [11], **small workl (SW)** graph [50], and **Barabasi-Albert (BA)** scale-free graph [6] while the red layer is generated to balance the concentrations of all graphlets. For each category, the two layers are interconnected in three ways. The first is that every node in the blue layer has a corresponding node in the red layer (i.e., all the nodes have a one-to-one mapping). The second is that the blue layer and the red layer are of the same scale (i.e., the same number of nodes), while half of the nodes in the blue layer connect to their counterparts in the red layer. The third is that the blue graph is two times larger than the red graph and all the red nodes connect to the half of the nodes in the blue graph. Note that we cannot exhaust all the possible combinations of two layers, and we believe that the sampling is meaningful if the sizes of the two layers are not too much different.
Table 7. Coefficient $\alpha_i$

| $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | $\alpha_4$ | $\alpha_5$ | $\alpha_6$ | $\alpha_7$ |
|---|---|---|---|---|---|---|
| 2 | 1 | 3 | 3 | 6 | 6 | 4 |

$\alpha_8$ $\alpha_9$ $\alpha_{10}$ $\alpha_{11}$ $\alpha_{12}$ $\alpha_{13}$ $\alpha_{14}$

$\alpha_8$ $\alpha_9$ $\alpha_{10}$ $\alpha_{11}$ $\alpha_{12}$ $\alpha_{13}$ $\alpha_{14}$

4 8 8 7 12 12 18

Table 8. State Transition Matrix of RWMix

| Current state | Next state | $(X_{m}, X_{m+1}, X_{m+2})$ | $(X_{m}, X_{m+1}, Y_{m+2})$ | $(X_{m+1}, Y_{m+2}, Y_{m+3})$ |
|---|---|---|---|---|
| $(X_{m-1}, X_{m}, X_{m+1})$ | $\frac{1}{r_{X_{m+1}} + b_{X_{m+1}}}$ | $\frac{1}{r_{X_{m+1}} + b_{X_{m+1}}}$ | 0 |
| $(X_{m}, X_{m+1}, Y_{m+2})$ | $\frac{1}{b_{X_{m+1}} + r_{Y_{m+2}}}$ | 0 | $\frac{1}{r_{Y_{m+2}} + b_{X_{m+1}}}$ |
| $(X_{m+1}, Y_{m+2}, Y_{m+3})$ | $\frac{1}{b_{X_{m+1}}}$ | 0 | 0 |

Table 9. Information of Nine Synthetic Datasets

| Graph | $|\mathcal{V}|$ | $|\mathcal{E}|$ | $#8(10^{-6})$ | $#10(10^{-6})$ | $#12(10^{-6})$ |
|---|---|---|---|---|---|
| ER1 | 100K | 948K | 8.06 | 5.27 | 5.67 |
| ER2 | 100K | 753K | 3.97 | 6.7 | 2.11 |
| ER3 | 100K | 753K | 3.85 | 4.97 | 1.86 |
| SW1 | 100K | 757K | 7.88 | 3.78 | 1764 |
| SW2 | 120K | 824K | 961 | 413 | 1843 |
| SW3 | 100K | 651K | 7.37 | 10.2 | 1800 |
| BA1 | 100K | 946K | 3.25 | 2.06 | 54 |
| BA2 | 120K | 1068K | 12.7 | 2.16 | 44 |
| BA3 | 100K | 957K | 38.5 | 24.9 | 32.8 |

We denote these three two-layer graphs by #1, #2, #3. Then, ER1 (respectively, ER2, ER3) indicates #1 (respectively, #2, #3) Erdos-Renyi graph, and the other notations are the same. We summarize the basic information of those nine datasets in Table 9, and enumerate the ground-truth concentrations of some representative graphlets. For example, the number #8 refers to the 8th graphlet. One can observe that the ground-truth concentrations of the graphlets are highly diverse, and a rare graphlet is believed to have a high estimation error, and vice versa. In this set of experiments, we mainly focus on the relative change that our algorithms perform in different datasets, rather than the absolute magnitudes of the estimation errors.

In what follows, we fix the sampling algorithm while changing the graph topology. Due to lack of space, we stick to RWEBE, and similar conclusions can be drawn from other sampling algorithms. First, the network topology does influence the accuracy of our sampling algorithm, but through the concentration of graphlets. Figure 5, Figure 6, and Figure 7 show the MREs of different graphlets on ER, SW, and BA networks, respectively. The MSEs of the graphlets \{1, 2, 3, 4, 5\} are always of the order of $10^{-2}$, indicating the high sampling accuracy. Those of the graphlets \{6, 7, 8, 9, 10, 11, 12, 13, 14\} are relatively higher, which are less accurately estimated. The decrease in sampling accuracy is originated from the very low graphlet concentration (as shown in the Appendix). In Figure 5, one can see that the MREs of graphlets \{2, 4\} are obviously higher on
BA1 while the MREs of graphlets \{6,9,12,14\} are much lower on SW1. Partial reasons attribute to that each node in SW1 has a more uniform number of edges than ER1 and BA1, thus the rarest graphlets still have higher concentrations compared with ER1 and BA1. However, no graph has the advantage on every graphlet. Similar observations can be strengthened in Figure 6 and Figure 7.

Second, changing the density of inter-layer links usually does not influence the performance of our sampling algorithm. Figure 8 evaluates the MREs of three ER graphs in which their differences are marginal. In Figure 9, the MREs of graphlets \{7,8,10,11,13\} on SW2 are much lower than on SW1 and SW3. In Figure 10, the MREs of graphlets \{7,8,10,11,13\}. We hereby argue that the extremely low concentration of graphlets, instead of the sampling algorithm itself, causes the increased mean of relative error. The relationships between the ground-truth graphlet concentration and the MRE on all the graphs are shown in Figure 11–Figure 19 where the x-coordinate indicates the logarithm of the concentration ratio and the y-coordinate is the logarithm of the MRE. One
can observe that all the rare graphlets (from 6th to 14th at the order of $10^{-6}$) encounter relatively large MREs on every network. As the concentration ratio increases, the MRE decreases accordingly. In Figure 15, the estimation on SW2 is more accurate simply because that the rare graphlets have much higher concentration ratios (at the order of $10^{-4}$) than the other networks. Therefore, we can conclude that the accuracy of our sampling algorithm is throttled by the ground-truth concentration ratio of a graphlet.

**Quantitative analysis with mixed and real-world graphs.**

We evaluate the accuracy and convergence of the proposed algorithms with mixed and real-world graph datasets. By mixed datasets, we mean a graph with the blue layer coming from the real-world, and the red layer synthetic. We also provide a real-world dataset coming from two social media. Table 10 lists the basic information of these datasets including Epinions [35], Facebook [32, 46], and ff-tw [13, 38].

Figure 20 shows the MRE of all the graphlets using the sampling approaches RWNbN, RWOMRN, RWMiX, RWEBE, and RWNR. The MREs of the Epinions1 network are shown in Figure 20. The first observation is that all the proposed algorithms achieve satisfactory accuracies on the set of graphlets \{1, 2, 3, 4, 5, 6, 9, 12, 14\}, with the MRE ranging from below 0.01 to around 0.1. To be specific, for graphlets with concentration around $10^{-3}$ (\{6, 9, 12, 14\}), the MREs are about 0.1. For graphlets with concentration around $10^{-2}$ (\{2, 4\}), the MREs are about 0.07. For graphlets

| Graph   | $|V|$ | $|E|$ | #7($10^{-6}$) | #9($10^{-6}$) | #11($10^{-6}$) |
|---------|------|------|--------------|--------------|--------------|
| Epinions1 | 76K  | 842K | 23           | 8577         | 40           |
| Epinions2 | 76K  | 990K | 20           | 6410         | 53           |
| Facebook1 | 63K  | 1532K| 55           | 18038        | 68           |
| Facebook2 | 63K  | 1841K| 43           | 13269        | 96           |
| ff-tw    | 44K  | 978K | 152          | 702          | 311          |
with concentration higher than $10^{-2}$ ($\{1, 3, 5\}$), the MREs are around 0.02 - 0.03. The MREs on the set of graphlets $\{7, 8, 10, 11, 13\}$ are much higher due to their concentrations below $10^{-5}$. The second observation is that RWNbN has a better MRE than RWOMRN with more blue edges, and underperforms RWOMRN with more red edges, and the MRE of RWMix is usually in between. Hence, given a fixed sampling budget, we can balance the sampling accuracy of different types of graphlets by splitting this budget appropriately. As the third observation, the MRE of RWNbN, RWOMRN, RWMix, and RWEbE is comparable to that of RWNR, which allows the random walk on both layers. This implies that even though the random walk on the red graph is restricted, the sampling of two-layer graphlets can still be achieved with high accuracy. Last but not the least, no algorithm dominates the others on every graphlet.

We select two representative algorithms, RWNbN and RWMix, for analyzing the convergence of sampling. Figure 21 illustrates the NRMSE of different graphlets with RWNbN when the number of sampling steps increases. For most of the graphlets, the NRMSE reduces quickly in the beginning, and tends to converge when the sampling step reaches $20k$. The graphlets in the set $\{1, 3, 5\}$ see the NRMSE at the order of $10^{-2}$, and those in the set $\{2, 4, 6, 9, 12, 14\}$ have the NRMSE at the order of $10^{-1}$. The 8th and 10th graphlets contain two red edges that are “non-overlapping” with the blue ones so that they are difficult to be sampled under the red layer restriction. At the same time, those two are two of the rarest graphlets in this dataset, which increases the difficulty to sample them. Consequently, the NRMSE of the 8th and 10th graphlets are high, indicating both large errors and large variance. Figure 22 illustrates the NRMSE of different graphlets with RWMix.

We observe similar trends as those of RWNbN. Since RWMix opportunistically samples more red nodes than RWNbN, the estimation on the graphlets with more red nodes is expected to be better. For instance, the NRMSE of the 4th graphlet in RWMix is obviously lower than that in RWNbN, and the NRMSE of the 8th and 10th graphlets are also slightly lower. We further evaluate the MSE of the proposed algorithms on the Epinions2 network in Figure 23. The NRMSEs of RWEbE and RWOMRN are shown in Figure 24 and Figure 25, respectively. The experimental results validate the accuracy and convergence of our sampling algorithms in general except for a few extremely rare graphlets. For example, the NRMSE curve of 8th graphlet, the rarest one, has the
highest variance. Whereas the estimation of more frequent graphlets in the set \{1, 3, 5\} is highly accurate and fast convergent. This implies that sampling rare graphlets is even more challenging in multiplex and restricted graphs.

Figure 26 and 27 show the MRE of the proposed algorithms with Facebook graphs, i.e., Facebook1 and Facebook2. Similar patterns regarding the MRE are observed as those of the Epinions graphs, whereas the accuracy of Facebook graphs is better than the Epinions graphs. Figures 28 and 29 demonstrate the NRMSE of all the graphlets as the sampling step increases from $2^k$ to $20^k$. The NRMSE of Facebook graphs is smaller than that of Epinion graphs for each graphlet, and the NRMSE curves of the rare graphlets are more smooth, indicating smaller variances in different sampling rounds. The underlying reason is that the rare graphlets in the Facebook graphs have relatively high concentration than their counterparts in the Epinions graphs.

Figure 30 shows the MRE of proposed algorithms with the real-world dataset ff-tw. Because the graphlet concentrations are relatively even (most sparse graphlet has concentration more than $10^{-3}$), the MRE of graphlet from 6–14 has significant improvement. From Figure 31 to Figure 35, we demonstrate the convergence results of all the proposed algorithms and the baseline algorithm. It can be found that RWNR has the best convergence. Its curves decrease quickly with the random walk steps increasing and finally converge to relatively small values. However, our proposed algorithms also show a comparable performance, which suggests that our methods can be used in realistic data and obtain satisfactory results.
At last, we summarize our answers to the raised questions. First, the network structure has a great influence on sampling accuracy. However, such an influence is not exerted through the type of each graph or the way of interconnecting two graphs, but the sparsity of graphlets. Second, our proposed algorithm accurately estimates the concentration of the graphlets whose ground-truth percentages are above $10^{-4}$, with an MRE below 0.04. They achieve comparable performance to the benchmark algorithm without random walk restrictions. Third, RWOMRN and RWMix usually sample more red nodes, thus leading to relatively higher accuracy for the graphlets with more red nodes.

6 CONCLUSION AND FUTURE WORK

In this article, we explore the random walk-based sampling on an interconnected two-layer network in which one layer allows random walk and the other layer only permits the one-hop node or edge sampling. We present a suite of three-node graphlets for two-layer multiplex networks, and propose a novel joint random work and node sampling approach to perform unbiased estimation of graphlet concentration. An analytic bound on the random work steps is derived to achieve asymptotic convergence. We observe the inherent tradeoff in the two-layer network with a fixed amount of random walk steps. The concentration of the graphlets with more nodes on one layer can be better estimated when more sampling steps are assigned to this layer. We further present two variants to balance the tradeoff between the accuracies of different graphlets. Experimental
results on the real-world and synthetic graphs manifest that the proposed algorithms can accurately estimate the two-layer graphlet concentration, and the level of accuracy is comparable to the random walk without layer restriction.

In our future work, we will evaluate the proposed algorithms on more realistic settings and also look into the stream sampling of two-layer networks. The streaming sampling has been studied in traditional single-layer networks where the restriction is the limited memory size of storing the incoming edges. The major challenge is the sparsity of certain types of graphlets, e.g., many more wedges than triangles, and this phenomenon is aggravated in multiplex networks. In Reference [1], the authors introduced the importance sampling method to improve the sampling accuracy of triangles. On this basis, we plan to design novel stream sampling algorithms to capture the graphlet concentration of two-layer multiplex networks when the edges arrive sequentially and the storage space is very limited.

APPENDICES

A STATIONARY DISTRIBUTION

We have the following two equations:

\[ \pi \times P = \pi, \]  
\[ \sum \pi = 1. \]

The set of all the blue neighbors of \( X_m \) is denoted by \( B(X_m) \) and the set of all the red neighbors of \( X_m \) is denoted by \( R(X_m) \). According to Equation (11), we have the following:

\[ \pi(X_m, X_{m+1}, Y_{m+2}) = \sum_{a \in B(X_m)} \pi(a, X_m, X_{m+1}) \times \frac{1}{r_{X_{m+1}} + b_{X_{m+1}}}, \]  
\[ \pi(X_m, X_{m+1}, X_{m+2}) = \sum_{a \in B(X_m)} \pi(a, X_m, X_{m+1}) \times \frac{1}{r_{X_{m+1}} + b_{X_{m+1}}} \]

\[ + \sum_{b \in R(X_{m+1})} \pi(X_m, X_{m+1}, b) \times \frac{1}{b_{X_{m+1}}}. \]  

One can check that the following solution satisfies Equations (13) and (14), where \( M \) is an unknown constant,

\[ \begin{cases} 
\pi(X_m, X_{m+1}, Y_{m+2}) = \frac{1}{M(r_{X_{m+1}} + b_{X_{m+1}})} \\
\pi(X_m, X_{m+1}, X_{m+2}) = \frac{1}{Mb_{X_{m+1}}} 
\end{cases} \]

Now, we can use Equation (12) to solve the unknown constant \( M \).

According to Equation (12), we have the following:

\[ \sum_{v \in V_B} \sum_{a_1 \in B(v), a_2 \in B(v)} \pi(a_1, v, a_2) \]

\[ + \sum_{v \in V_B} \sum_{a_1 \in B(v), b_1 \in R(v)} \pi(a_1, v, b_1) \]

\[ = \sum_{v \in V_B} b_v^2 \frac{1}{Mb_v} + \sum_{v \in V_B} b_v r_v \frac{1}{M(b_v + r_v)} \]

\[ = \sum_{v \in V_B} \frac{1}{M} \left( b_v + \frac{b_v r_v}{b_v + r_v} \right) = 1, \]
which means:

$$M = \sum_{v \in V_n} \left( b_v + \frac{b_v r_v}{\Delta r + r_v} \right) = 2|E_B| + \sum_{v \in V_n} \frac{b_v r_v}{\Delta r + r_v}. \tag{16}$$

Finally, we have the stationary distribution:

$$\begin{align*}
\pi(X_m, X_{m+1}, Y_{m+2}) &= \frac{1}{M(x_m+1 + b_{x_{m+1}})} \\
\pi(X_m, X_{m+1}, X_{m+2}) &= \frac{1}{M(x_{x_{m+1}})}.
\end{align*}$$

It can be checked that the stationary distribution above is the solution of Equations (11) and (12).

## B ERROR BOUND

Here is the proof of Theorem 3.3. It needs several steps.

**Lemma B.1.** \( \forall 0 < \delta < 1, \exists \text{ constant } \xi, \text{ such that, when } n \geq \frac{\xi - H}{\alpha_i |C_i|} \frac{\tau}{e^2} \left( \ln \frac{|\phi|_\|C_i\|}{\delta} \right), \text{ we have} \)

$$\Pr \left( \frac{|\hat{C}_i|}{|C_i|} - 1 > \frac{\delta}{3} \right) < \frac{1}{2}. \tag{17}$$

Proof: Define \( f_i(S) = g_i(S)/\pi(S)H \), where \( f_i \in [0, 1] \). Let

$$\mu_i = \mathbb{E}_\pi[f_i] = \sum_{S \in W} f_i(S)\pi(S),$$

which is

$$\sum_{S \in W} g_i(S)/H = \alpha_i |C_i|/H. \tag{18}$$

If we have \( n \) valid states by random walking with sampling, which are \( \{ S_j \}_{j=1}^n \), then \( |\hat{C}_i| = \frac{1}{n} \sum_{j=1}^n \frac{g_i(S_j)}{\pi(S_j)/H} \).

Define \( Z = \sum_{j=1}^n f(S_j) = \sum_{j=1}^n \frac{g_i(S_j)}{\pi(S_j)/H} \). According to Theorem 3.2, we have

$$\Pr \left( \frac{1}{n} \sum_{j=1}^n \frac{g_i(S_j)}{\pi(S_j)/H} - \frac{\alpha_i |C_i|}{H} \frac{|\phi|_\|C_i\|}{3} \right) \leq c||\phi||_\pi e^{-e^2 \mu_i n}$$

where \( c \) is a constant. By simplifying, we have

$$\Pr \left( \frac{|\hat{C}_i|}{|C_i|} - 1 > \frac{\delta}{3} \right) \leq c||\phi||_\pi e^{-e^2 \mu_i n}.$$ 

Assuming that \( c||\phi||_\pi e^{-e^2 \mu_i n} \leq \frac{\delta}{2} \), we can obtain \( n \geq \frac{648r}{e^2 \mu_i} \ln \frac{2c||\phi||_\pi}{\delta} \). That means \( \exists \text{ constant } \xi, \text{ such that, when } n \geq \frac{\xi - H}{\alpha_i |C_i|} \frac{\tau}{e^2} \left( \ln \frac{|\phi|_\|C_i\|}{\delta} \right), \text{ we have the following:} \)

$$\Pr \left( \frac{|\hat{C}_i|}{|C_i|} - 1 \leq \frac{\delta}{2} \right). \tag{19}$$

**Lemma B.2.** \( \forall 0 < \delta < 1, \exists \text{ constant } \xi, \text{ such that, when } n \geq \frac{\xi - H}{\alpha_{min} |C_i|} \frac{\tau}{e^2} \left( \ln \frac{|\phi|_\|C_i\|}{\delta} \right), \text{ we have} \)

$$\Pr \left( \frac{|\hat{C}_i|}{|C_i|} - 1 > \frac{\delta}{3} \right) < \frac{1}{2}. \tag{20}$$

Proof: Define

$$f(S) = \frac{1}{\sum_{i=1}^{14} \alpha_i g_i(S)\pi(S)H} \left( \frac{\alpha_{min}}{H} \right).$$
Then
\[ \mu = \mathbb{E}_\pi[f] = \frac{\alpha_{\min}}{H} \sum_{S \in W} \frac{\mathbb{I}[|S| = 3]}{\sum_{i=1}^{14} \alpha_i g_i(S)}. \] (21)

Because \( \alpha_i|C_i| = \sum_{S \in W} g_i(S) \),
\[ |C| = \sum_{i=1}^{14} |C_i| = \sum_{S \in W} \frac{g_i(S)}{\alpha_i} \] (22)
we can find that
\[ \sum_{i=1}^{14} \sum_{S \in W} g_i(S) \] (23)

So, we have \( \mathbb{E}_\pi[f] = \frac{\alpha_{\min}|C|}{H} \).

Define \( Z = \sum_{j=1}^{n} f(S_j) \). According to Theorem 3.2, we have
\[ \Pr \left( \left| \frac{1}{n} \sum_{j=1}^{n} \frac{\mathbb{I}[|S| = 3]}{\sum_{i=1}^{14} \alpha_i g_i(S_j)} \pi(S_j) - \frac{\alpha_{\min}|C|}{H} \right| > \frac{\epsilon}{3} \right) \leq c |||\phi|||_\pi e^{-2\epsilon^2\mu n} \] (24)

We can find that
\[ \frac{1}{n} \sum_{j=1}^{n} \frac{\mathbb{I}[|S| = 3]}{\sum_{i=1}^{14} \alpha_i g_i(S_j)} = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{14} \frac{g_i(S_j)}{\alpha_i \pi(S_j)} = |\hat{C}|. \] (25)
So,
\[ \Pr \left( \left| \frac{\hat{C}}{|C|} - 1 \right| > \frac{\epsilon}{3} \right) \leq c |||\phi|||_\pi e^{-2\epsilon^2\mu n} \] (26)
Assuming that \( c |||\phi|||_\pi e^{-2\epsilon^2\mu n} \leq \frac{\delta}{2} \), we can obtain \( n \geq \frac{648r}{e^2 \mu} \ln \frac{2c |||\phi|||_\pi}{\delta} \). That means \( \exists \) constant \( \xi \), such that, when \( n \geq \frac{e^2 H}{\alpha_{\min} |C|} \frac{\xi}{\epsilon^2} \ln \frac{|||\phi|||_\pi}{\delta} \), we have
\[ \Pr \left( \left| \frac{\hat{C}}{|C|} \right| > \frac{\delta}{2} \right) < \frac{\delta}{2}. \] (27)

THEOREM B.3. \( \forall 0 < \delta < 1, \exists \) constant \( \xi \), such that, when \( n \geq \xi \frac{H}{\alpha_{\min} |C|} \frac{\epsilon^2}{\epsilon^2} \ln \frac{|||\phi|||_\pi}{\delta} \), we have
\[ \Pr((1 - \epsilon)d_i \leq \hat{d}_i \leq (1 + \epsilon)d_i) > 1 - \delta. \] (28)

Proof: If \( n \) satisfies the condition in this Theorem, then it must satisfy the conditions in Lemma B.1 and Lemma B.2. Let \( A_1 \) denote the event that \( \left| \frac{\hat{C}}{|C|} - 1 \right| \leq \frac{\xi}{3} \), and \( A_2 \) denote the event that \( \left| \frac{\hat{C}}{|C|} - 1 \right| \leq \frac{\xi}{3} \). We have \( \Pr(A_1) > 1 - \frac{\delta}{2} \), and \( \Pr(A_2) > 1 - \frac{\delta}{2} \). If \( A_1 \) and \( A_2 \) happen, then
\[ (1 - \epsilon)d_i \leq \frac{1}{1 + \frac{\xi}{3}} |C_j| \leq |\hat{C}_j| \leq \frac{1}{1 - \frac{\xi}{3}} |C_j| \leq (1 + \epsilon)d_i. \] (29)
Let \( F \) denote the event that \( (1 - \epsilon)d_i \leq \hat{d}_i \leq (1 + \epsilon)d_i \), then \( A_1 \cap A_2 \subset F \). So, we have
\[ \Pr(F) \geq \Pr(A_1 \cap A_2) \geq \Pr(A_1) + \Pr(A_2) - 1 = 1 - \delta. \] (30)


C GROUND TRUTH OF DATASETS

Here we present the ground truth of all the concentrations of each dataset in Table 11 and Table 12.

Table 11. Information of Datasets

| Graph | |V| |E| #1(10^-1) | #2(10^-1) | #3(10^-1) | #4(10^-1) | #5(10^-2) | #6(10^-4) |
|-------|--------|--------|--------|------------|------------|------------|------------|------------|------------|
| ER1   | 100K   | 948K   | 1.85   | 3.03       | 2.39       | 1.96       | 7.73       | 0.0378     |
| ER2   | 100K   | 753K   | 4.14   | 2.48       | 1.68       | 1.27       | 4.34       | 0.135      |
| ER3   | 100K   | 753K   | 4.13   | 2.47       | 1.69       | 1.27       | 4.34       | 0.122      |
| SW1   | 100K   | 757K   | 1.72   | 3.19       | 2.23       | 2.06       | 7.25       | 14         |
| SW2   | 120K   | 824K   | 2.17   | 3.98       | 1.57       | 1.73       | 4.55       | 10.8       |
| SW3   | 100K   | 651K   | 3.25   | 2.61       | 1.58       | 1.93       | 5.62       | 33.2       |
| BA1   | 100K   | 946K   | 2.83   | 1.39       | 3.68       | 0.899      | 12         | 0.415      |
| BA2   | 120K   | 1068K  | 3.58   | 2.36       | 2.92       | 0.503      | 6.35       | 0.448      |
| BA3   | 100K   | 957K   | 2.62   | 3.3        | 2.15       | 1.46       | 4.69       | 0.327      |
| Epinions1 | 76K   | 842K   | 2.66   | 0.378      | 4.51       | 0.321      | 19.2       | 33.5       |
| Epinions2 | 76K   | 990K   | 1.59   | 0.4        | 4.32       | 0.548      | 29.4       | 15.7       |
| Facebook1 | 63K   | 1532K  | 2.68   | 1.49       | 3.4        | 0.94       | 10.7       | 95.3       |
| Facebook2 | 63K   | 1841K  | 1.47   | 1.56       | 3.18       | 1.69       | 17.3       | 41         |
| ff-tw | 44K    | 978K   | 3.8    | 1.2        | 2.5        | 1.0        | 9.4        | 104.3      |

Table 12. Information of Datasets (Continued)

| Graph | #7(10^-5) | #8(10^-5) | #9(10^-7) | #10(10^-7) | #11(10^-6) | #12(10^-5) | #13(10^-5) | #14(10^-5) |
|-------|------------|------------|------------|------------|------------|------------|------------|------------|
| ER1   | 0.836      | 80.6       | 0.756      | 52.7       | 11.5       | 0.567      | 3.68       | 0.0995     |
| ER2   | 0.857      | 39.7       | 0.584      | 67.1       | 5.84       | 0.211      | 3.48       | 0.0994     |
| ER3   | 1.14       | 38.5       | 0.77       | 49.7       | 5.09       | 0.186      | 4.72       | 0.0745     |
| SW1   | 0.542      | 78.8       | 0.796      | 37.8       | 8.37       | 176        | 3.12       | 37.7       |
| SW2   | 147        | 9620       | 213        | 4130       | 1300       | 184        | 282        | 65.2       |
| SW3   | 0.586      | 73.7       | 141        | 102        | 7.75       | 180        | 5.29       | 75.6       |
| BA1   | 1.37       | 32.5       | 8.02       | 20.6       | 20         | 5.48       | 4.86       | 0.917      |
| BA2   | 2.43       | 127        | 7.26       | 21.6       | 16.2       | 4.4        | 5.68       | 1.03       |
| BA3   | 4.87       | 385        | 5.3        | 249        | 62         | 3.28       | 21         | 0.769      |
| Epinions1 | 2.31 | 16         | 858        | 14.2       | 40.5       | 730        | 16.2       | 207        |
| Epinions2 | 1.99     | 26.4       | 641        | 32.3       | 53.3       | 876        | 36.8       | 398        |
| Facebook1 | 5.5     | 148        | 1800       | 87.9       | 68.4       | 1140       | 22.1       | 240        |
| Facebook2 | 4.38     | 230        | 1330       | 257        | 96.5       | 1430       | 53.6       | 515        |
| ff-tw | 152.0      | 74650.0    | 701.5      | 44026.3    | 3110.9     | 685.9      | 1696.7     | 252.4      |

REFERENCES

[1] Nesreen K. Ahmed, Nick Duffield, Jennifer Neville, and Ramana Kompella. 2014. Graph sample and hold: A framework for big-graph analytics. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1446–1455.
[2] N. K. Ahmed, N. Duffield, T. Willke, and R. A. Rossi. 2017. On sampling from massive graph streams. arXiv:1703.02625. Retrieved from https://arxiv.org/abs/1703.02625.
[3] N. K. Ahmed, N. Duffield, T. L. Willke, and R. A. Rossi. 2017. On sampling from massive graph streams. VLDB J. 10, 11 (2017), 1430–1441.
[4] N. K. Ahmed, J. Neville, and R. Kompella. 2013. Network sampling: From static to streaming graphs. *ACM Trans. Knowl. Discov. Data* 8, 2 (2013), 1–56.

[5] N. K. Ahmed, J. Neville, R. A. Rossi, and N. Duffield. 2015. Efficient graphlet counting for large networks. In *Proceedings of the International Conference on Data Mining*. IEEE, 1–10.

[6] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74, 1 (2002), 47.

[7] Massoud Amin. 2002. Toward secure and resilient interdependent infrastructures. *J. Infrastruct. Syst.* 8, 3 (2002), 67–75.

[8] Luca Becchetti, Paolo Boldi, Carlos Castillo, and Aristides Gionis. 2008. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 16–24.

[9] M. A. Bhuiyan, M. Rahman, M. Rahman, and Al. H. Mohammad. 2012. Guise: Uniform sampling of graphlets for large graph analysis. In *Proceedings of the International Conference on Data Mining*. IEEE, 91–100.

[10] Hanjo D. Boekhout, Walter A. Kosters, and Frank W. Takes. 2018. Counting multilayer temporal motifs in complex networks. In *Proceedings of the International Conference on Complex Networks and Their Applications*. Springer, 565–577.

[11] Béla Bollobás and Béla Bollobás. 2001. *Random Graphs*. Number 73. Cambridge University Press.

[12] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca. 2009. Network analysis in the social sciences. *Science* 323, 5916 (2009), 892–895.

[13] Fabio Celli, F. Marta L. Di Lascio, Matteo Magnani, Barbara Pacelli, and Luca Rossi. 2010. Social network data and practices: The case of Friendfeed. In *Proceedings of the International Conference on Social Computing, Behavioral Modeling and Prediction*. Lecture Notes in Computer Science. Springer, Berlin.

[14] X. Chen, Y. Li, P. Wang, and J. Lui. 2016. A general framework for estimating graphlet statistics via random walk. *VLDB J.* 10, 3 (2016), 253–264.

[15] Charles J. Geyer. 2005. Markov chain Monte Carlo lecture notes.

[16] M. Gjoka, C. T. Butts, M. Kurant, and A. Markopoulou. 2011. Multigraph sampling of online social networks. *IEEE J. Select. Areas Commun.* 29, 9 (2011), 1893–1905.

[17] J. W. Godfrey. 1969. The mechanism of a road network. *Traffic Eng. Contr.* 8, 8 (1969).

[18] Hong Huang, Jie Tang, Lu Liu, JarDer Luo, and Xiaoming Fu. 2015. Triadic closure pattern analysis and prediction in social networks. *IEEE Trans. Knowl. Data Eng.* 27, 12 (2015), 3374–3389.

[19] M. Jha, C. Seshadhri, and A. Pinar. 2013. A space efficient streaming algorithm for triangle counting using the birthday paradox. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*. ACM, 589–597.

[20] M. Jha, C. Seshadhri, and A. Pinar. 2015. Path sampling: A fast and provable method for estimating 4-vertex subgraph counts. In *Proceedings of the International Conference on World Wide Web*. 495–505.

[21] J. Kuszczyszyn, K. Musial, and M. Budka. 2011. Link prediction based on subgraph evolution in dynamic social networks. In *Proceedings of the International Conference on Social Computing*. IEEE, 27–34.

[22] L. Katzir and S. J. Hardiman. 2015. Estimating clustering coefficients and size of social networks via random walk. *ACM Trans. Web* 9, 4 (2015), 19.

[23] Peter Klimmek and Stefan Thurmer. 2013. Triadic closure dynamics drives scaling laws in social multiplex networks. *New J. Phys.* 15, 6 (2013), 063008.
Sampling Graphlets of Multiplex Networks: A Restricted Random Walk Approach

[32] Jérôme Kunegis. 2013. Konect: The Koblenz network collection. In Proceedings of the 22nd International Conference on World Wide Web. 1343–1350.

[33] J. Kunegis, A. Lommatzsch, and C. Bauckhage. 2009. The slashdot zoo: Mining a social network with negative edges. In Proceedings of the International Conference on World Wide Web. ACM, 741–750.

[34] C. H. Lee, X. Xu, and D. Y. Eun. 2012. Beyond random walk and metropolis-hastings samplers: Why you should not backtrack for unbiased graph sampling. In Proceedings of the ACM Special Interest Group on Performance Evaluation (SIGMETRICS’12). Vol. 40. ACM, 319–330.

[35] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. Retrieved from http://snap.stanford.edu/data.

[36] J. Y. Li and M. Y. Yeh. 2011. On sampling type distribution from heterogeneous social networks. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 111–122.

[37] R. H. Li, J. X. Yu, L. Qin, R. Mao, and T. Jin. 2015. On random walk based graph sampling. In Proceedings of the International Conference on Data Engineering. IEEE, 927–938.

[38] Matteo Magnani and Luca Rossi. 2011. The ML-model for multi-layer social networks. In Proceedings of the Advances in Social Network Analysis and Mining (ASONAM’11). IEEE Computer Society, 5–12.

[39] Abdelaziz Mohaisen, Aaram Yun, and Yongdae Kim. 2010. Measuring the mixing time of social graphs. In Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. 383–389.

[40] J. D. Noh and H. Rieger. 2004. Random walks on complex networks. Phys. Rev. Lett. 92, 11 (2004), 118701.

[41] N. Pržulj. 2007. Biological network comparison using graphlet degree distribution. Bioinformatics 23, 2 (2007), e177–e183.

[42] M. Rahman, M. Bhuiyan, and M. Al. Hasan. 2012. Graft: An approximate graphlet counting algorithm for large graph analysis. In Proceedings of the International Conference on Information and Knowledge Management. ACM, 1467–1471.

[43] J. Scott. 1988. Social network analysis. Sociology 22, 1 (1988), 109–127.

[44] C. Seshadhri, A. Pinar, and T. G. Kolda. 2013. Triadic measures on graphs: The power of wedge sampling. In Proceedings of the International Conference on Data Mining. SIAM, 10–18.

[45] S. Suri and S. Vassilvitskii. 2011. Counting triangles and the curse of the last reducer. In Proceedings of the International Conference on World Wide Web. 607–614.

[46] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. 2009. On the evolution of user interaction in Facebook. In Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN’09).

[47] J. S. Vitter. 1985. Random sampling with a reservoir. ACM Trans. Math. Softw. 11, 1 (1985), 37–57.

[48] P. Wang, J. Lui, B. Ribeiro, D. Towsley, J. Zhao, and X. Guan. 2014. Efficiently estimating motif statistics of large networks. ACM Trans. Knowl. Discov. Data 9, 2 (2014), 8.

[49] P. Wang, J. Tao, J. Zhao, and X. Guan. 2015. Moss: A scalable tool for efficiently sampling and counting 4-and 5-node graphlets. arXiv:1509.08089. Retrieved from https://arxiv.org/abs/1509.08089.

[50] Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of ‘small-world’ networks. Nature 393, 6684 (1998), 440–442.

[51] Sebastian Wernicke and Florian Rasche. 2006. FANMOD: A tool for fast network motif detection. Bioinformatics 22, 9 (2006), 1152–1153.

[52] O. Younis, M. Krunz, and S. Ramanan. 2006. Node clustering in wireless sensor networks: Recent developments and deployment challenges. IEEE Netw. 20, 3 (2006), 20–25.

[53] Jing Zhang, Zhanpeng Fang, Wei Chen, and Jie Tang. 2015. Diffusion of “following” links in microblogging networks. IEEE Trans. Knowl. Data Eng. 27, 8 (2015), 2093–2106.

Received May 2020; revised December 2020; accepted March 2021