SUBSAMPLING IN LARGE GRAPHS USING RICCI CURVATURE

Anonymous authors
Paper under double-blind review

ABSTRACT

In the past decades, many large graphs with millions of nodes have been collected/constructed. The high computational cost and significant visualization difficulty hinder the analysis of large graphs. To overcome the difficulties, researchers have developed many graph subsampling approaches to provide a rough sketch that preserves global properties. By selecting representative nodes, these graph subsampling methods can help researchers estimate the graph statistics, e.g., the number of communities, of the large graph from the subsample. However, the available subsampling methods, e.g., degree node sampler and random walk sampler, tend to leave out minority communities because nodes with high degrees are more likely to be sampled. To overcome the shortcomings of the existing methods, we are motivated to apply the community information hidden in the graph to the subsampling method. Though the community structure is unavailable, community structure information can be obtained by applying geometric methods to a graph. An analog of Ricci curvature in the manifold is defined for the graph, i.e., Ollivier Ricci curvature. Based on the asymptotic results about the within-community edge and between-communities edge's OR curvature, we propose a subsampling algorithm based on our theoretical results, the Ollivier-Ricci curvature Gradient-based subsampling (ORG-sub) algorithm. The proposed ORG-sub algorithm has two main contributions: First, ORG-sub provides a rigorous theoretical guarantee that the probability of ORG-sub taking all communities into the final subgraph converges to one. Second, extensive experiments on synthetic and benchmark datasets demonstrate the advantages of our algorithm.

1 INTRODUCTION

As we enter the big data era, our capacity to access large graphs (a.k.a. networks) has provided unprecedented opportunities and challenges. For example, (Wang et al., 2011) presented a Twitter social network, which has more than 190 million nodes (users) who generate more than 65 million edges (tweets) every day (Wang et al., 2011). Such huge networks enable researchers to tackle more complex problems. However, they pose great challenges to storing, visualizing, and analyzing since their sheer volumes render many computational methods infeasible.

Graph subsampling. Graph subsampling is a commonly used technique to address this scalability issue because of its simplicity and efficiency. Graph subsampling aims to take a subgraph that preserves critical features of the full graph. Various graph subsampling methods that preserve different graph properties have been proposed, including node sampling (Mall et al., 2013; Zeng et al., 2019), edge sampling (Krishnamurthy et al., 2005), and exploration sampling (Goodman, 1961; Leskovec et al., 2005; Hübler et al., 2008; Maiya & Berger-Wolf, 2010). Researchers evaluate the graph subsampling approaches by measuring the similarity between the features of the original graph and those of the subgraph. The features include, e.g., degree distribution (Adamic et al., 2001), minimum cut (Hu & Lau, 2013), and the number of triangles (Seshadhri et al., 2014).

An important graph feature that has received less attention is the number of communities (denoted by M), which is critical for various important tasks such as community detection and the network autoregression model. In particular, the number of communities M is required as input for many existing community detection algorithms, including modularity maximization (Newman, 2006; Good et al., 2010), spectral clustering (Von Luxburg, 2007; Rohe et al., 2011; Liu et al., 2018),

and pseudo-likelihood based methods (Amini et al., 2013; Wang et al., 2021). In addition, the number of communities M is also a required input in many dynamic network models, e.g., the network autoregression model (Jiang et al., 2020; Chen et al., 2020), and spectral clustering in dynamic networks (Liu et al., 2018). Thus, it is highly desirable for subsampling methods to yield subgraphs preserving the number of communities M. Despite many successful applications, existing subsampling methods tend to leave out minority communities (i.e., a community with a smaller number of nodes) because nodes with high degrees are more likely to be sampled into subgraphs. Consequently, these subsampling methods usually underestimate M, especially for graphs with imbalanced community structures.

To overcome the shortcomings of existing methods, we develop a graph subsampling method that yields subgraphs that can be used to accurately estimate M. Achieving this goal is challenging since the community structure is hidden and unavailable. Fortunately, recent studies indicate that the community structure is a geometric phenomenon by considering a graph as a Riemannian geometric object (Ni et al., 2015). Some insights into the community information can be obtained by applying some geometric methods to a graph (Ni et al., 2015; 2019; Sia et al., 2019).

Ollivier Ricci Curvature of Graph. In particular, a graph can be regarded as a discrete version of a Riemannian manifold (Ni et al., 2019). A node of the graph is analogous to a point on a Riemannian manifold, and a pair of connected nodes in a graph is analogous to two points connected by a geodesic on a manifold. The partition of a graph into communities is analogous to the geometric decomposition of a Riemannian manifold (Ni et al., 2019). Ricci curvature is a key tool for the geometric decomposition of a Riemannian manifold. It measures how the Riemannian manifold deviates from the flat manifold. Recently, (Lin et al., 2011) defines an analog of Ricci curvature for the graph, i.e., the Ollivier Ricci (abbreviated as OR) curvature. In this paper, we theoretically show that the OR curvatures of edges within a densely connected community are asymptotically larger than those of edges between two sparsely connected communities (Ni et al., 2015; Gosztolai & Arnaudon, 2021), as the size of the graph increases.

Ollivier Ricci Curvature Gradient Based Graph Subsampling. Based on our theoretical result, we propose an OR curvature gradient-based graph subsampling algorithm (abbreviated as ORG-sub). Specifically, ORG-sub randomly chooses one edge as the starting point of the subgraph and calculates the OR curvature of the selected edge. ORG-sub then gradually expands the subgraph by taking the next edge whose OR curvature shows the largest difference from the OR curvature of the previously taken edge. Here, we define the difference as the OR curvature gradient (ORG). We use the ORG to guide the expansion of ORG-sub, i.e., direct ORG-sub which edge to take next. All edges that ORG-sub took expand into the final subgraph.

Our proposed ORG-sub enjoys two main advantages. First, ORG-sub has a rigorous theoretical guarantee. In particular, we prove that the probability of ORG-sub taking all communities into the final subgraph converges to one, indicating that even nodes in the minority community are subsampled. More importantly, we theoretically show that the estimation of M by the subsampled graph converges to true M. Second, our extensive empirical experiments based on the simulated and real-world datasets show that the estimator based on ORG-sub subgraphs accurately estimates the number of communities while greatly reducing the computation cost.

2 Preliminaries

2.1 Notations and Definitions

A graph is made up of nodes that are connected by edges. The connected component is a set of nodes in a graph that are linked to each other by paths. Without loss of generality, we consider the undirected graph with only one connected component since it's trivial to categorize different components into different communities in a graph with multiple components. Denote a graph as $G = \langle V, E \rangle$, where V is an unordered set of nodes and E is an unordered set of edges, $E \subseteq \{(u, v)|u \in V, v \in V\}$. Operator $|\cdot|$ calculates the cardinality of the set. We then denote the cardinality of the node set in a graph by $|V|$ ($|V| = n$), the number of edges in a graph by $|E|$, and the neighborhood set of a node v by $\delta(v) = \{w|(v, w) \in E\}$. The degree of a node v is defined as the cardinality of its neighborhood: $d_v = |\delta(v)|$. For a subset of V, $S \subseteq V$, the subgraph induced by the subset S is denoted by $G[S] = (S, E_S)$, where the edge set $E_S = \{(v_S, w_S) \in E|v_S \in S, w_S \in S\}$. The

neighborhood of the node set S is defined by $N(S) = \bigcup_{v \in S} \delta(v)$, and the neighboring edge set of edge $e = (u, v)$ is $\Delta(e) = \{(x, y) | x \in \{u, v\}, y \in \delta(x) \setminus \{u, v\}\}$. The number of communities of the graph G is denoted by $M(G)$ (M is short for $M(G)$ when G is given), and the distribution of the communities is denoted by $\mathcal{D}(G) = \{n_i, i \in \{1, ..., M(G)\}\}$, where n_i is the size of the i-th community.

2.2 Ollivier Ricci Curvature of Graphs

A Riemannian manifold $(\mathcal{M}, g)$ is defined as a smooth manifold $\mathcal{M}$ equipped with a metric $g : \mathcal{TM} \times \mathcal{TM} \to \mathbb{R}$, where $\mathcal{TM}$ is the tangent space of $\mathcal{M}$ (Lee, 2018). Ricci curvature of Riemannian manifold measures the deviation of the manifold from a flat manifold (Lee, 2006; Ni et al., 2015; Samal et al., 2018). The "flat" means the distance between two nodes is the same as the distance between the two local spaces centered around the nodes. In the discrete version, the space defined on the node is discretely uniformly distributed in the node's neighborhood, then the distance between two spaces can be explained as the average distance between the neighborhoods of the two nodes. Through such an analogy, the counterpart of Ricci curvature, Ollivier Ricci curvature (Lin et al., 2011), is defined in graphs. Previous empirical results show that the OR curvature of graphs is related to the Ricci curvature of the underlying manifold: the graph with large (or small) OR curvature correspond to the underlying manifold with large (small) Ricci curvature (Ni et al., 2015; Samal et al., 2018; Ni et al., 2019; Gosztolai & Arnaudon, 2021). Theoretical results have shown the OR curvature of random geometric graphs converges to the Ricci curvature of the underlying Riemannian manifold (Boguna et al., 2021; van der Hoorn et al., 2021). Here we use a toy example to illustrate the relationship between the OR curvature of graphs and the corresponding underlying manifold. In Figure 1, the hyperbolic space (with negative curvature) is especially suited for tree-structured graphs (Nickel & Kiela, 2017), the Euclidean space (with zero curvature) is especially suited for grid-structured graphs, and the spherical space (with positive curvature) is especially suited for complete graphs (Ni et al., 2015).

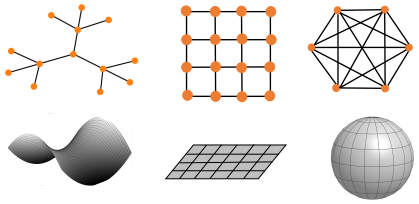

Figure 1: The left one is a tree-structured graph corresponding to a hyperbolic manifold with negative curvature. The middle one is a grid-structured graph corresponding to a Euclidean space with zero curvature. The right one is a complete graph K_6, which is embedded in a spherical surface.

We now introduce the definition of OR curvature of the graph. For $\alpha \in [0, 1]$ and any node u with degree d_u, we first define the probability distribution of node u by m_u^α:

$$m_u^\alpha(x) = \begin{cases} \alpha & \text{if } x = u \\ (1-\alpha)/d_u & \text{if } x \in \delta(u) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Let $d(u, v)$ be the geodesic distance between nodes u and v, which is the shortest path between two nodes of a graph. We shall present the Wasserstein distance of two nodes in the graph, which is defined as the transportation distance between two nodes' probability distributions m_u^α and m_v^α.

Definition 2.1 (Wasserstein distance)**.** Let X be the metric space with two probability distributions μ_1 and μ_2 with mass 1, respectively. A transportation plan from μ_1 to μ_2 is a mapping $\xi : X \times X \to [0, 1]$ satisfying $\int_y d\xi(x, y) = d\mu_1(x)$ and $\int_x d\xi(x, y) = d\mu_2(y)$. Then $W(\mu_1, \mu_2)$ is the infimum taken over all transportation plan ξ between μ_1 and μ_2:

$$W(\mu_1, \mu_2) = \inf_\xi \iint d(x, y) d\xi(x, y) \tag{2}$$

Substitute μ_1 and μ_2 by probability distributions of nodes m_u^α and m_v^α, the Wasserstein distance between two nodes is,

$$W(m_u^\alpha, m_v^\alpha) = \inf_\xi \sum_{u,v \in V} \xi(u, v) d(u, v) \tag{3}$$

The Wasserstein distance $W(m_u^\alpha, m_v^\alpha)$ can be computed by linear programming Lin et al. (2011).

Then the OR curvature (Ollivier, 2007; Lin et al., 2011) over the edge (u, v) is defined as follows.

Definition 2.2 (Ollivier Ricci Curvature)**.** Given the Wasserstein distance $W(m_u^\alpha, m_v^\alpha)$ and the geodesic distance $d(u, v)$, we can define the Ollivier Ricci curvature as:

$$\kappa(u, v) = 1 - \frac{W(m_u^\alpha, m_v^\alpha)}{d(u, v)} \tag{4}$$

When $\alpha = 1$, $W(m_u^1, m_v^1) = d(u, v)$, and the OR curvature is 0 for all edges in graph G. Thus, we set $\alpha = 0.5$ to avoid the OR curvature values being trivial.

3 Ollivier Ricci Curvature Gradient and Community Structure

The OR curvature of the graph reveals the properties of the underlying Riemannian manifold and provides insights into the community structure of graphs. Previous work has shown that the large (or small) curvature corresponds to the edge which is more (or less) connected than a grid (Ni et al., 2015; Samal et al., 2018; Ni et al., 2019; Gosztolai & Arnaudon, 2021). As shown in Figure 2, the left is a graph with five communities, and the right one is a graph with three communities generated by the stochastic block model. The warmer color indicates a smaller curvature, and the colder color indicates a larger curvature. Previous empirical results and the above toy example have related the OR curvature of graphs to the community structure of the graph. However, rigorous theoretical proofs that the OR curvature of the within-community edge is larger than the between-communities edge in a graph are not developed yet. In the Lemma 3.1, we theoretically prove the lower bound of within-community edges' OR curvatures are larger than the upper bound of the between-communities edges' OR curvatures in stochastic block models (SBM).

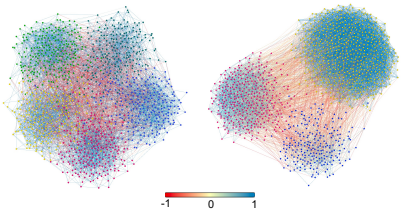

Figure 2: The cold-colored edges have large OR curvatures, and the warm-colored edges have small OR curvatures. Nodes with different colors belong to different communities.

3.1 Asymptotic Results of OR Curvature Gradient in SBM

Lemma 3.1 (Ollivier-Ricci curvature of random graphs generated by stochastic block models)**.** *Consider a graph with M communities generated from a stochastic block model (SBM) $\mathcal{B}\Big(\left\{p_{in}^{(i)}\right\}_{i=1}^{M}, p_{out}, \{n_i\}_{i=1}^{M}\Big)$, where $p_{in}^{(i)}$ represents the probability of an edge connecting nodes in the same community B_i, p_{out} represents the probability of an edge connecting nodes from different communities, and $\{n_i\}_{i=1}^{M}$ are the sizes of each community in the SBM. We assume that, for any $i = 1, ..., M$, $p_{in}^{(i)} > p_{out}$, and $\frac{n_i}{n}$ converges to $r_i (1 > r_i > 0)$, as n goes to infinity. Let $n_\vee = \max\{n_i\}_{i=1}^{M}$, and $p_\vee = \max\left\{p_{in}^{(i)}\right\}_{i=1}^{M}$. We denote the node x from the community B_i by x_{B_i}, the edge (x_{B_i}, y_{B_i}) where x is from B_i and y is from B_j, and the edge's OR curvature by $\kappa(x_{B_i}, y_{B_i})$. The following statements hold for the OR curvature.*

1. *If $p_{out} \leq \sqrt{\frac{(n_i-1)\ln n}{n \ln n_i}}$ and $p_{in}^{(i)} n_i + p_{out}(n - n_i) > \frac{8}{3}\ln n_i$, for the lower bound of within-community OR curvature, we almost surely have*

$$\kappa(x_{B_i}, y_{B_i}) \geq \frac{(n_i - 2){p_{in}^{(i)}}^2 + (n - n_i){p_{out}}^2}{(n_i - 1)p_{in}^{(i)} + (n - n_i)p_{out}} + \mathcal{O}\Big(\frac{\ln n}{p_{out}^2 n}\Big). \tag{5}$$

2. *If ${p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n - n_i) > 4\ln n_i$, for the upper bound of within-community OR curvature, we almost surely have*

$$\kappa(x_{B_i}, y_{B_i}) \leq p_{in}^{(i)} + \mathcal{O}\Big(\frac{\ln n_i}{n}\Big). \tag{6}$$

3. *If $p_{out} > \sqrt{(3\ln n_\vee)\Big/\left[\frac{p_{in}^{(i)}}{p_{out}}(n_i + n_j) + n - n_i - n_j\right]}$, for the upper bound of between-communities OR curvature, we almost surely have*

$$\kappa(x_{B_i}, y_{B_j}) \leq \frac{n - n_i - n_j}{n - 1}p_{out} + \frac{n_i - n_j - 2}{n - 1}p_\vee + \mathcal{O}\Big(\sqrt{\frac{p_\vee \ln n}{p_{out} n}}\Big). \tag{7}$$

Lemma 3.2. *Combining the conclusion of statement 1 and statement 2 in Lemma 3.1, almost surely, for any* $(x'_{B_i}, y'_{B_j}) \in \Delta((x_{B_i}, y_{B_i}))$*, we have*

$$|\kappa(x_{B_i}, y_{B_i}) - \kappa(x'_{B_i}, y'_{B_i})| \leq \frac{p_{out}(p_{in}^{(i)} - p_{out})}{\frac{n_i-1}{n-n_i}p_{in}^{(i)} + p_{out}} + \mathcal{O}\Big(\frac{1}{p_{out}n}\Big). \tag{8}$$

Statements 1 and 2 in Lemma 3.1 together show that under certain conditions, the probability that the OR curvature of within-community edges falls in $(\frac{(n_i-2)p_{in}^{(i)^2}+(n-n_i){p_{out}}^2}{(n_i-1)p_{in}^{(i)}+(n-n_i)p_{out}}, p_{in}^{(i)})$ goes to 1 as the graph size n goes to infinity. Statement 3 in Lemma 3.1 shows that under certain conditions, the probability that the OR curvature of between-communities edges being no larger than $\frac{n-n_i-n_j}{n-1}p_{out} + \frac{n_i-n_j-2}{n-1}p_\vee$ goes to 1 as n goes to infinity. Thus, we can conclude Lemma 3.2 that the probability of the maximum difference of two within-community edges' OR curvatures being $\frac{p_{out}(p_{in}^{(i)}-p_{out})}{\frac{n_i-1}{n-n_i}p_{in}^{(i)}+p_{out}}$ goes to 1 as n goes to infinity. Since the largest OR curvature difference between within-community edges can be bounded, we can prove the difference is small enough to distinguish from the difference between $\kappa(x_{B_i}, y_{B_i})$ and $\kappa(x^*_{B_i}, y^*_{B_j})$.

Theorem 3.3. *Combining the conclusion of statement 1 and statement 3 in Lemma 3.1, if* $p_\vee < \frac{p_{in}^{(i)}n}{n_i+n_j}$, $p_{in}^{(i)} < \frac{p_{out}(n-n_i)(n+n_i+n_j)}{n^2-(n_i+n_j)(n-2n_i)}$*, and conditions given in the lemma 3.1 are satisfied, almost surely, for any* $(x^*_{B_i}, y^*_{B_j}) \in \Delta((x_{B_i}, y_{B_i}))$*, we have*

$$\max_{(x'_{B_i}, y'_{B_i}) \in \Delta((x_{B_i}, y_{B_i}))} |\kappa(x_{B_i}, y_{B_i}) - \kappa(x'_{B_i}, y'_{B_i})| < \kappa(x_{B_i}, y_{B_i}) - \kappa(x^*_{B_i}, y^*_{B_j}). \tag{9}$$

The Theorem 3.3 shows the OR curvature of the within-community edge $\kappa(x_{B_i}, y_{B_i})$ is larger than the OR curvature of the between-communities edge $\kappa(x_{B_i}, y_{B_j})$ by the maximum of $|\kappa(x_{B_i}, y_{B_i}) - \kappa(x'_{B_i}, y'_{B_i})|$.

The theorem holds under mild conditions. When p_{out} is small enough, and $p_\vee$ is not too large compared to p_{out}, the conditions can be satisfied. When the graph size is large, e.g., thousands of or millions of nodes, the constraint of p_{out} and $p_{in}^{(i)}$ is loosened, and the inequality in Theorem 3.3 can be easily satisfied.

3.2 OR Curvature Gradient and Communities

We define the OR curvature gradient (ORG) as $|\kappa(x^{(j)}, y^{(j)}) - \kappa(x^{(i)}, y^{(i)})|$, where $(x^{(j)}, y^{(j)}) \in \Delta((x^{(i)}, y^{(i)}))$. It is the difference between the OR curvature of two adjacent edges. According to the theoretical results, the within-community ORG $|\kappa(x_{B_i}, y_{B_i}) - \kappa(x'_{B_i}, y'_{B_i})|$ is significantly different from the between-communities ORG $|\kappa(x_{B_i}, y_{B_i}) - \kappa(x^*_{B_i}, y^*_{B_j})|$, this motivates us to propose a subsampling algorithm based on ORG, which extracts the community information with theoretical guarantee.

4 OR Curvature Gradient Based Graph Subsampling

The existing graph subsampling methods usually use degree information and random walk through the graph. However, the subsampled graph tends to leave out minority communities (i.e., a community with a smaller number) since they tend to sample nodes with high degrees. Due to the lack of taking advantage of the community structure information during the expansion, the available subsampling algorithms usually underestimate M. From the previous work, the community structure relates to the geometric phenomenon of the underlying Riemannian Manifold. Thus, we develop a fast and efficient ORG-based subsampling algorithm that is able to traverse the communities of the graph instead of trapping in a single community. We then prove the probability of the proposed ORG-sub taking all communities of the full graph into the subsample converging to one. Consequently, the subsampled graph attained by the ORG-sub algorithm can get a larger proportion of the minority communities than other subsampling algorithms and helps estimate M more accurately, as shown in Figure 3. In addition, we theoretically show the estimation of M by the subsampling algorithm converges to true M.

4.1 SUBSAMPLING ALGORITHM

The ORG carries the community structure information since it differentiates within-community edges and between-communities edges with theoretical guarantees in Theorem 3.3. We can use ORG as a guide for expanding the subgraph in the direction of more communities. First, we randomly sample an edge $(x, y) \in E$ as the starting edge. The probability of obtaining an edge (x, y) at the start is:

$$P((x^{(1)}, y^{(1)})) = \frac{1}{|E|}. \tag{10}$$

To take advantage of the theoretical properties of the ORG, the ORG-based subsampler expands to the next edge whose OR curvature shows the greatest difference from the OR curvature of the previously taken edge. The edge subsampled in the $i+1$-th step $(x^{(i+1)}, y^{(i+1)})$ given the edge subsampled in the i-th step $(x^{(i)}, y^{(i)})$ can be expressed by:

$$(x^{(i+1)}, y^{(i+1)}) = \underset{(x,y)\in\Delta((x^{(i)},y^{(i)}))}{\arg\max} |\kappa(x, y) - \kappa(x^{(i)}, y^{(i)})|. \tag{11}$$

Since the difference between the within-community ORG is smaller than the between-community ORG, the proposed subsampler will expand to another community instead of trapping in one community. After the subsampler stops expanding or the subsampling budget is used up, we get subsampled nodes. Then we can get the subgraph induced by the subsampled nodes. Details of the algorithm are in Algorithm 1.

Algorithm 1

Input: Graph G; Number of nodes to be subsampled $\tilde{n}$; OR curvature of the graph G calculated by parameter α; the subsampled node set $S = \emptyset$
Initialization: Randomly choose a node v_0 as the start of sampling.
Cold Start: Among all neighbors of v_0, randomly add v_1 to S.
While: $|S| < \tilde{n}$

- Given the edge $e_t = (v_{t-1}, v_t)$ selected in the t-th step, get the edge curvatures of the e_t's neighboring edges $\Delta(e_t)$.
- Select the edge e_{t+1} in the edge set $\Delta(e_t)$ that has the greatest OGR.
- Add the end point of the edge e_{t+1} to the subsampled node set S.
- $t = t + 1$.

Output: Subsampled node set S and the induced subgraph $G[S]$

4.2 ORG-SUB FOR ESTIMATING M

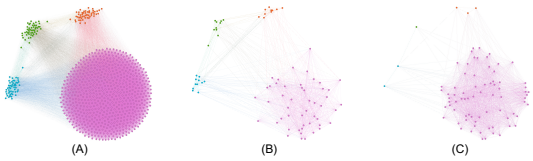

Figure 3: (A): A graph generated by SBM with community size $(650, 50, 50, 50)$, $p_{in} = 0.8$ and $p_{out} = 0.1$. (B): The subsampled graph by the proposed ORG-sub algorithm. The proportion of subsampling is 10%. (C) The subsampled graph by degree-based subsampling algorithm. The proportion of subsampling is 10%.

Applying the proposed ORG-sub algorithm to the full graph, as shown in Figure 3 (A), we can get a subsample as shown in Figure 3 (B). Compared with the subsample obtained by the degree-based subsampler in Figure 3 (C), the proposed algorithm can subsample more nodes in minority communities. The toy example illustrates that the proposed ORG-sub algorithm can preserve the community structure information better. Thus, the subsampled graph can help estimate M of the full graph.

We take multiple subgraphs by applying ORG-sub r times. For each subsampled graph $G[S]$ given the sample size $\tilde{n}$, we get an estimation of the number of communities, $\hat{M}(G[S])$, then the mean of $\hat{M}(G[S])$ is calculated as our final estimation of M. As for the choice of the algorithm estimating $\hat{M}(G[S])$, we use the state-of-the-art network cross-validation method (Li et al., 2020).Compared with other methods (Wang & Bickel, 2017; Chen & Lei, 2018), network cross-validation is more robust to complicate settings, i.e., the number of communities is large or the community structure is not conspicuous.

Theoretically, we prove the probability of the ORG-sub algorithm expanding to all communities converging to one as the expanding steps increase.

Theorem 4.1 (Probability of the ORG-sub's Expansion to Communities)**.** *Given the graph G generated by Theorem 3.3, we apply the ORG-sub algorithm 1 to the graph G and take graph subsamples with size $\tilde{n}$ denoted by $G[S]$. Under the conditions in Theorem 3.3, $\frac{\tilde{n}}{n} \to 1$, we can prove the probability of the final subgraph $G[S]$ taken by ORG-sub carrying nodes in the community B_i is*

$$P(v \in B_i \mid v \in G[S]) \to 1 \tag{12}$$

Theorem 4.1 also tells us that the probability that the final subgraph taken by the ORG-sub algorithm converges to one.

Corollary 4.1.1. *If the subsample with size $\tilde{n}$ satisfies $\frac{\tilde{n}\rho}{\log \tilde{n}} \to \infty$, where ρ controls the expected node degree of subsampled graph, and as the subsampling times $r \to \infty$, the probability the subsampled graph underestimates the true M is:*

$$Pr(\frac{1}{r}\sum_{i=1}^{r} \hat{M}(G[S_i]) < M) \to 0, \tag{13}$$

Theorem 4.1 theoretically guarantees a certain proportion of each community in the full graph is subsampled. Thus, the convergence of the network cross-validation method in (Li et al., 2020) holds for the subsampled graphs. we can then prove the convergence of the ORG-sub-based estimation $\hat{M}(G[S_i])$, referring to the Theorem 3 in (Li et al., 2020).

4.3 Computational Analysis of the Algorithms

The computation complexity of the Algorithm 1. Aside from the computation complexity of the OR curvature of the graph (which is near-linear by Sinkhorn's algorithm), we analyze the time complexity of the sampling procedure. The time complexity of querying the neighboring edges and their corresponding curvatures is of order $\mathcal{O}(|E|)$. The time complexity of the sorting step to attaining the most different curvature from the neighbors is $\mathcal{O}(d \log d)$, where d is the degeneracy of graph G (The degeneracy of the graph is smaller than or equal to the maximum degree of the graph). Thus, the time complexity of taking one step is about $\mathcal{O}(|E| + d \log d)$. To sample $\tilde{n}$ nodes, we need to run about $\tilde{n}$ steps and get the induced subgraph from the sampled node set S. Getting induced subgraph has the time complexity of order $\mathcal{O}(\tilde{n}d)$. Thus the total complexity of the sampling procedure is $\mathcal{O}\left(\tilde{n}(|E| + d \log d)\right)$. Considering the worst case, if the graph is highly connected and the degeneracy is in the same order as the number of nodes, n, the complexity of the sampling procedure is $\mathcal{O}(\tilde{n}(|E| + n \log n))$.

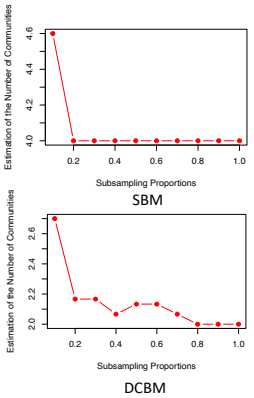

Figure 4: The function of the estimation of the number of communities with respect to the subsampling proportion for two block models.

The computation complexity of the algorithm estimating M. The computation complexity of the network cross-validation procedure is of order $\mathcal{O}(\tilde{n}^3)$. Thus, the number of communities estimation step's computation complexity is of order $\mathcal{O}(r\tilde{n}^3)$. We noticed that if we use the whole graph to detect the number of communities, the time complexity is $\mathcal{O}(n^3)$, which would be larger than the combined computational complexity of the subsampling procedure and the estimation procedure only by the subsample, $\mathcal{O}(\tilde{n}^3) + \mathcal{O}\left(\tilde{n}(|E| + d \log d)\right)$, conditioning on the subsample size $\tilde{n}$ is small enough.

Choice of the Subsample Size. When implementing our algorithm, there are some hyper-parameters to be specified. One important hyper-parameter is the size of the subsamples which determines the performance and efficiency of the algorithm.

Empirically, we plot the estimation of the number of communities as the function of the subsampling proportion. As shown in Figure 4, we select the elbow where the estimation of the number of communities starts to converge to save computational resources and stabilize the estimation results.

5 Experiment

We evaluate the performance of our algorithm on both synthetic datasets and real-world datasets. We use the metric, MAE= $\frac{1}{r}\sum_{i=1}^{r}\left|M(G) - \hat{M}(G[S_i])\right|$ to evaluate the accuracy of subsampling

based estimation. We also compare the proposed method with one node sampler, degree-based node sampler (DBN) (Adamic et al., 2001), and six benchmark exploration-based graph subsampling methods, including Random Walk Sampler (RW) (Gjoka et al., 2010), Multi-dimensional Random Walk Sampler (MDRW) (Ribeiro & Towsley, 2010), Metropolis Hasting Random Walk Sampler (MHRW) (Hübler et al., 2008), Foreset Fire Sampler (FFS) (Leskovec et al., 2005), Snowball Sampler (Goodman, 1961), one community structure preserved sampler, and Community Structure Expansion Sampler (CSE) (Maiya & Berger-Wolf, 2010). All the experiments are conducted on a machine with an 18-core NVIDIA Tesla V100 GPU (3.00 GHz).

5.1 Datasets

Synthetic Dataset We generate synthetic datasets by the stochastic block models (SBM) and degree-corrected block models (DCBM), which can assign the community distribution to each node. We set the community proportion as $(3/4, 1/10, 1/12, 1/15)$ with 900 nodes in total. The out-in-ratio (the ratio of between-communities edges over within-community edges) controls the ratio of the number of edges between the communities and within a community. A higher ratio represents a noisier graph. The degree corrected model corrects the node degree by a power-law distribution. Given the probability of an edge within a community ($p_{in} = 0.8$), we vary the probability of an edge between communities (p_{out}) ($\{0.06, 0.08, 0.10, 0.12\}$) and subsampling proportions ($\{0.1, 0.12, 0.14, 0.16\}$) from low to high for both block models to observe how our method performs as the setting becomes more challenging compared with others. More details about the generation of the DCBM datasets are presented in Appendix.

Real-world Dataset We use four read-world graph datasets with true labels to validate the performance of our method. The datasets (Rossi & Ahmed, 2015; Rozemberczki et al., 2020a; Sen et al., 2008) are widely used as the benchmark datasets to test the performance of the graph community detection. All the graphs are considered unweighted and undirected. Besides, all the self-loop edges and isolated nodes are removed. *Cora* constructs the relationship among different scientific publications. There is an edge connecting two publications if they belong to the same category. *Polbooks* and *Polblogs* construct the relationship among books and blogs about US politics, respectively. The edges in *Polbooks* represent the frequent co-purchasing of books; the edges in *Polbooks* represent one of the blogs cited by the other. *Facebook* constructs a social media network on Facebook. Table A.7 in Appendix provides the main features of the four datasets, recording the number of the nodes, edges, and communities of the graph, sparsity, and the clustering coefficient of the whole graph. These datasets cover different levels of size, density, and community size, which guarantee a robust test for the performance of our method.

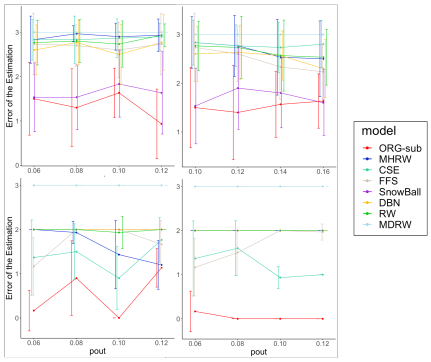

Figure 5: The function of the performance of MAE with respect to pout and prop for DCBM dataset.

5.2 Results of Synthetic Dataset

We use synthetic datasets with hidden community structures to evaluate the performance of our method. We set the hyper-parameter to calculate the OR curvature of graphs as default $\alpha = 0.5$. The subsampling times r for estimating the number of communities is set as 3, which is enough to get a stable estimation result. As for other methods, we set the hyper-parameters as the default in package *Little Ball of Fur* (Rozemberczki et al., 2020b). We replicate the experiments 30 times under each setting and compare the performance in terms of MAE for all methods. The results of SBM and DCBM datasets are reported in Figure 5. The error bar is the standard deviation of 30 replications. We set subsampling proportion (prop) as 0.1 when varying the probability of an edge between communities (p_{out}) and set $p_{out} = 0.06$ when varying the prop. Our method can get a more accurate estimation of the number of communities than other methods. More details about the table of results, including each combination of pout and prop values, are presented in Table A.5 and A.2 in Appendix. We also compare the computation time for estimation of the number of communities by using the whole graph and the subgraph. It turns out the time for estimating on the subgraph together

with the time for subsampling is still much shorter than the time for estimation on the whole graph. It is consistent with our conclusion about the complexity of the algorithm. Details about computational time are presented in Appendix.

5.3 Results of Real-world Dataset

We compare the performance of the proposed ORG-sub algorithm with subsamples generated by different algorithms by MAE defined above. Table 1 records the average error of the estimation of the number of communities over 30 replications. Still, our algorithm outperforms other algorithms in most cases. To be noted, the *Polbogs* and *Facebook* datasets' community sizes are more imbalanced (the ratio of majority and minority community size is larger) than the *Polbooks* and *Cora* datasets. We found that the proposed ORG-sub algorithm performs better when the community of the graph is more imbalanced. This is consistent with our intuition that other algorithms might leave out the minority communities while ours can expand to minority communities. The true M of each dataset is recorded in the first column of Table 1. Similar to the Synthetic Dataset, we also compare the computation time for estimating the number of communities on the subgraph and the whole graph. The computation time on the subgraph (together with the subsampling time) is still much shorter than on the whole graph. More details are presented in the Appendix.

Dataset	Prop.	ORG-sub	MHRW	CSE	FFS	SnowBall	DBN	RW	MDRW
Polblogs True: 2	10%	**0.00**	0.40	0.33	0.20	0.03	0.03	0.07	0.87
	12%	**0.00**	1.23	0.23	0.10	0.00	0.23	0.13	-0.77
	14%	**0.00**	1.13	0.13	0.10	0.00	0.33	0.03	0.83
	16%	**0.00**	0.67	0.57	0.33	0.00	0.57	1.60	-0.90
Polbooks True: 3	10%	**0.00**	1.20	0.62	2.68	0.48	0.60	0.33	0.00
	12%	**0.00**	1.20	0.62	2.68	0.48	0.60	0.33	0.00
	14%	1.41	1.17	1.29	0.36	**0.21**	0.71	1.25	2.00
	16%	**0.15**	0.89	0.15	1.57	2.00	0.29	1.00	1.00
Cora True: 7	10%	4.40	**1.53**	4.97	3.40	3.00	3.37	3.87	5.80
	12%	**1.27**	2.63	4.83	3.93	3.73	4.67	4.60	5.63
	14%	**1.93**	2.87	4.90	4.33	3.97	4.90	4.50	5.47
	16%	**1.40**	3.63	4.83	3.13	4.57	4.50	3.73	5.27
Facebook True: 11	10%	**5.27**	6.83	7.97	6.87	8.77	7.57	5.77	10.00
	12%	**4.90**	6.10	7.93	6.67	7.53	8.53	4.90	9.97
	14%	5.90	6.80	8.13	6.40	7.17	6.93	**4.93**	9.90
	16%	**5.50**	6.77	7.57	6.03	6.77	6.93	5.73	9.93

Table 1: Comparison of the performance on the error of the estimation of the number of communities for each dataset and subsample method.

6 Conclusion

In this paper, we propose a novel Ollivier-Ricci Curvature Gradient-based graph subsampling (ORG-sub) algorithm. We found our algorithm performs better than other subsampling algorithms in terms of subsampling-based number of communities estimation. The proposed method is the first attempt to utilize the graph's internal topological information to subsample a large graph. From the OR curvature gradient we defined based on properties of OR curvature of graphs, we can discover between-communities edges, thus, extracting community information from the graph.

In terms of both theoretical and empirical results, we show the proposed method can effectively traverse different communities and avoid trapping in one community by maximizing OR curvature gradient. The extensive experiments of the synthetic and real-world data demonstrate that our proposed method outperforms other benchmark graph subsampling methods in terms of estimating the number of communities. To be noted, our algorithm can perform better, especially on datasets with small-sized communities.

REFERENCES

Lada A Adamic, Rajan M Lukose, Amit R Puniyani, and Bernardo A Huberman. Search in power-law networks. *Physical review E*, 64(4):046135, 2001.

Arash A Amini, Aiyou Chen, Peter J Bickel, and Elizaveta Levina. Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics*, 41(4):2097–2122, 2013.

Marian Boguna, Ivan Bonamassa, Manlio De Domenico, Shlomo Havlin, Dmitri Krioukov, and M Serrano. Network geometry. *Nature Reviews Physics*, 3(2):114–135, 2021.

Elynn Y Chen, Jianqing Fan, and Xuening Zhu. Community network auto-regression for high-dimensional time series. *arXiv preprint arXiv:2007.05521*, 2020.

Kehui Chen and Jing Lei. Network cross-validation for determining the number of communities in network data. *Journal of the American Statistical Association*, 113(521):241–251, 2018.

Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. Walking in facebook: A case study of unbiased sampling of osns. In *2010 Proceedings IEEE Infocom*, pp. 1–9. Ieee, 2010.

Benjamin H Good, Yves-Alexandre De Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.

Leo A Goodman. Snowball sampling. *The annals of mathematical statistics*, pp. 148–170, 1961.

Adam Gosztolai and Alexis Arnaudon. Unfolding the multiscale structure of networks with dynamical ollivier-ricci curvature. *Nature Communications*, 12(1):1–11, 2021.

Pili Hu and Wing Cheong Lau. A survey and taxonomy of graph sampling. *arXiv preprint arXiv:1308.5865*, 2013.

Christian Hübler, Hans-Peter Kriegel, Karsten Borgwardt, and Zoubin Ghahramani. Metropolis algorithms for representative subgraph sampling. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 283–292. IEEE, 2008.

Binyan Jiang, Jailing Li, and Qiwei Yao. Autoregressive networks. *arXiv preprint arXiv:2010.04492*, 2020.

Vaishnavi Krishnamurthy, Michalis Faloutsos, Marek Chrobak, Li Lao, J-H Cui, and Allon G Percus. Reducing large internet topologies for faster simulations. In *International Conference on Research in Networking*, pp. 328–341. Springer, 2005.

John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

John M Lee. *Introduction to Riemannian manifolds*. Springer, 2018.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 177–187, 2005.

Tianxi Li, Elizaveta Levina, and Ji Zhu. Network cross-validation by edge sampling. *Biometrika*, 107(2):257–276, 2020.

Yong Lin, Linyuan Lu, and Shing-Tung Yau. Ricci curvature of graphs. *Tohoku Mathematical Journal, Second Series*, 63(4):605–627, 2011.

Fuchen Liu, David Choi, Lu Xie, and Kathryn Roeder. Global spectral clustering in dynamic networks. *Proceedings of the National Academy of Sciences*, 115(5):927–932, 2018.

Arun S Maiya and Tanya Y Berger-Wolf. Sampling community structure. In *Proceedings of the 19th international conference on World wide web*, pp. 701–710, 2010.

Raghvendra Mall, Rocco Langone, and Johan AK Suykens. Furs: Fast and unique representative subset selection retaining large-scale community structure. *Social Network Analysis and Mining*, 3 (4):1075–1095, 2013.

Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

Chien-Chun Ni, Yu-Yao Lin, Jie Gao, Xianfeng David Gu, and Emil Saucan. Ricci curvature of the internet topology. In *2015 IEEE conference on computer communications (INFOCOM)*, pp. 2758–2766. IEEE, 2015.

Chien-Chun Ni, Yu-Yao Lin, Feng Luo, and Jie Gao. Community detection on networks with ricci flow. *Scientific reports*, 9(1):1–12, 2019.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

Yann Ollivier. Ricci curvature of metric spaces. *Comptes Rendus Mathematique*, 345(11):643–646, 2007.

Bruno Ribeiro and Don Towsley. Estimating and sampling graphs with multidimensional random walks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 390–403, 2010.

Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL `https://networkrepository.com`.

Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, pp. 3125–3132. ACM, 2020a.

Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Little Ball of Fur: A Python Library for Graph Sampling. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, pp. 3133–3140. ACM, 2020b.

Areejit Samal, RP Sreejith, Jiao Gu, Shiping Liu, Emil Saucan, and Jürgen Jost. Comparative analysis of two discretizations of ricci curvature for complex networks. *Scientific reports*, 8(1):1–16, 2018.

Prithviraj Sen, Galileo Mark Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

C Seshadhri, Ali Pinar, and Tamara G Kolda. Wedge sampling for computing clustering coefficients and triangle counts on large graphs. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(4):294–307, 2014.

Jayson Sia, Edmond Jonckheere, and Paul Bogdan. Ollivier-ricci curvature-based method to community detection in complex networks. *Scientific reports*, 9(1):1–12, 2019.

Pim van der Hoorn, William J Cunningham, Gabor Lippner, Carlo Trugenberger, and Dmitri Krioukov. Ollivier-ricci curvature convergence in random geometric graphs. *Physical Review Research*, 3(1): 013211, 2021.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Jiangzhou Wang, Jingfei Zhang, Binghui Liu, Ji Zhu, and Jianhua Guo. Fast network community detection with profile-pseudo likelihood methods. *Journal of the American Statistical Association*, pp. 1–14, 2021.

Tianyi Wang, Yang Chen, Zengbin Zhang, Tianyin Xu, Long Jin, Pan Hui, Beixing Deng, and Xing Li. Understanding graph sampling algorithms for social network analysis. In *2011 31st international conference on distributed computing systems workshops*, pp. 123–128. IEEE, 2011.

YX Rachel Wang and Peter J Bickel. Likelihood-based model selection for stochastic block models. *The Annals of Statistics*, 45(2):500–528, 2017.

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graph-saint: Graph sampling based inductive learning method. In *International Conference on Learning Representations*, 2019.

Appendix for "Subsampling for Large Graphs Using Ricci Curvature"

The appendix shows the details of the proof, details of experiments including the parameter settings of generating the synthetic datasets, a description of real-world datasets, and more experiment results on synthetic and real-world datasets.

A Proof Details

Lemma A.1 (Ollivier-Ricci curvature of random graphs generated by stochastic block models)**.** *Consider a graph with M communities generated from a stochastic block model (SBM) $\mathcal{B}\Big(\Big\{p_{in}^{(i)}\Big\}_{i=1}^{M}, p_{out}, \{n_i\}_{i=1}^{M}\Big)$, where $p_{in}^{(i)}$ represents the probability of an edge connecting nodes in the same community B_i, p_{out} represents the probability of an edge connecting nodes from different communities, and $\{n_i\}_{i=1}^{M}$ are the sizes of each community in the SBM. We assume that, for any $i = 1, ..., M$, $p_{in}^{(i)} > p_{out}$, and $\frac{n_i}{n}$ converges to $r_i (1 > r_i > 0)$, as n goes to infinity. Let $n_\vee = \max\{n_i\}_{i=1}^{M}$, and $p_\vee = \max\Big\{p_{in}^{(i)}\Big\}_{i=1}^{M}$. We denote the node x from the community B_i by x_{B_i}, the edge (x_{B_i}, y_{B_i}) where x is from B_i and y is from B_j, and the edge's OR curvature by $\kappa(x_{B_i}, y_{B_i})$. The following statements hold for the OR curvature.*

1. *If $p_{out} \leq \sqrt{\frac{(n_i-1)\ln n}{n \ln n_i}}$, for the lower bound of within-community OR curvature, we almost surely have*

$$\kappa(x_{B_i}, y_{B_i}) \geq \frac{(n_i - 2){p_{in}^{(i)}}^2 + (n - n_i){p_{out}}^2}{(n_i - 1)p_{in}^{(i)} + (n - n_i)p_{out}} + \mathcal{O}\Big(\frac{\ln n}{p_{out}^2 n}\Big). \tag{14}$$

2. *As for the upper bound of within-community OR curvature, we almost surely have*

$$\kappa(x_{B_i}, y_{B_i}) \leq p_{in}^{(i)} + \mathcal{O}\Big(\frac{\ln n_i}{n}\Big). \tag{15}$$

3. *If $n_j > n_i$, for the upper bound of between-communities OR curvature, we almost surely have*

$$\kappa(x_{B_i}, y_{B_j}) \leq \frac{n - n_i - n_j}{n - 1} p_{out} + \frac{n_i - n_j - 2}{n - 1} p_\vee + \mathcal{O}\Big(\sqrt{\frac{p_\vee \ln n}{p_{out} n}}\Big). \tag{16}$$

Lemma A.2. *Combining the conclusion of statement 1 and statement 2 in Lemma A.1, almost surely, for any $(x'_{B_i}, y'_{B_j}) \in \Delta((x_{B_i}, y_{B_i}))$, we have*

$$|\kappa(x_{B_i}, y_{B_i}) - \kappa(x'_{B_i}, y'_{B_i})| \leq \frac{p_{out}(p_{in}^{(i)} - p_{out})}{\frac{n_i - 1}{n - n_i} p_{in}^{(i)} + p_{out}} + \mathcal{O}\Big(\frac{1}{p_{out} n}\Big). \tag{17}$$

We will use Chernoff's inequality to prove the lemmas.

Lemma A.3. *(Chernoff's inequality) Let $X_1, ..., X_n$ be the independent random variables with*

$$P(X_i = 1) = p_i, P(X_i = 0) = 1 - p_i \tag{18}$$

We consider the sum $X = \sum_{i=1}^{n} X_i$, with expectation $E(X) = \sum_{i=1}^{n} p_i$. Then we have

$$(Lowertail) P(X \leq E(X) - \lambda) \leq e^{-\lambda^2/2\mathrm{E}(X)} \tag{19}$$

$$(Uppertail) P(X \geq E(X) + \lambda) \leq e^{-\lambda^2/(2\mathrm{E}(X) + 2\lambda/3)} \tag{20}$$

WLOG, the proof of the theorem considers the situation when $M = 2$. The conclusion can be easily generated to any arbitrary M by setting one block as block B_1, and the other blocks as another block B_2. Before we prove the theorem, we need a few lemmas.

Lemma A.4. *If $p_{in}^{(i)} n_i + p_{out}(n - n_i) > \frac{8}{3} \ln n_i$, then the degree of all nodes in block B_i fall in the range $\Big(p_{in}^{(i)}(n_i - 1) + p_{out}(n - n_i)) - \sqrt{4(p_{in}^{(i)} n_i + p_{out}(n - n_i)) \ln n_1}, p_{in}^{(i)}(n_i - 1) + p_{out} n_i) + \sqrt{6(p_{in}^{(i)} n_i + p_{out}(n - n_i)) \ln n_i}\Big)$ with probability at least $1 - 2/n_i$.*

Proof. Without loss of generality, we just prove the first statement and the second statement can be proved in the same way.

For each vertex v in block B_i, it is easy to show that the expected degree of vertex v is $E(d_v^{B_i}) = p_{in}^{(i)}(n_i - 1) + p_{out}(n - n_i)$. Applying Chernoff's inequality with the lower tail $\lambda = \sqrt{4(p_{in}^{(i)} n_i + p_{out}(n - n_i)) \ln n_i}$, we have

$$P\Big(d_v^{B_i} - (p_{in}^{(i)}(n_i - 1) + p_{out}(n - n_i)) < -\sqrt{4(p_{in}^{(i)} n_i + p_{out}(n - n_i)) \ln n_i}\Big) \tag{21}$$

$$\leq e^{\frac{-4(p_{in}^{(i)} n_i + p_{out}(n - n_i)) \ln n_i}{2(p_{in}^{(i)}(n_i - 1) + p_{out}(n - n_i))}} < \frac{1}{n_i^2} \tag{22}$$

Applying Chernoff's inequality with the upper tail $\lambda = \sqrt{6(p_{in}^{(i)} n_i + p_{out}(n - n_i)) \ln n_i}$, we have

$$P\Big(d_v^{B_i} - (p_{in}^{(i)}(n_i - 1) + p_{out}(n - n_i)) > \sqrt{6(p_{in}^{(i)} n_i + p_{out}(n - n_i)) \ln n_i}\Big) \tag{23}$$

$$\leq e^{\frac{-6(p_{in}^{(i)} n_i + p_{out} n_i) \ln n_i}{2(p_{in}^{(i)}(n_i - 1) + p_{out} n_i) + 2/3\sqrt{6(p_{in}^{(i)} n_i + p_{out}(n - n_i)) \ln n_i}}} < \frac{1}{n_i^2} \tag{24}$$

In the last step, we used the assumption $p_{in}^{(i)}(n - n_i) + p_{out} n_i > \frac{8}{3} \ln(n - n_i)$. The probability that there is a vertex v in block B_i so that $d_v^{B_i} \notin \Big(p_{in}^{(i)}(n_i - 1) + p_{out}(n - n_i)) - \sqrt{4(p_{in}^{(i)} n_i + p_{out}(n - n_i)) \ln n_i}, p_{in}^{(i)}(n_i - 1) + p_{out}(n - n_i)) + \sqrt{6(p_{in}^{(i)} n_i + p_{out}(n - n_i)) \ln n_i}\Big)$ is at most $n_i\Big(\frac{1}{n_i^2} + \frac{1}{n_i^2}\Big) = \frac{2}{n_i}$. □

The co-degree d_{xy} of a pair of vertices (x, y) is the cardinality of the common neighborhood of x and y. Roughly speaking, when $p_{in}^{(i)}$ and p_{out} is large, the co-degree follows the binomial distribution; when $p_{in}^{(i)}$ and p_{out} is small, the co-degree follows the Poisson distribution. We can expect that all co-degrees within a block and between blocks are concentrated around a small interval.

Lemma A.5. *If ${p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n - n_i) > 4 \ln n_i$, then with probability at least $1 - 1/n_i$, all co-degrees of a pair of vertices in block B_i fall in the range $\Big({p_{in}^{(i)}}^2(n_i - 2) + {p_{out}}^2(n - n_i) - \sqrt{6({p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n - n_i)) \ln n_i}, {p_{in}^{(i)}}^2(n_i - 2) + {p_{out}}^2(n - n_i) + \sqrt{9({p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n - n_i)) \ln n_i}\Big)$.*

If $p_{in}^{(i)} p_{out} n_i + p_{out} p_{in}^{(i)} n_j + {p_{out}}^2(n - n_i - n_j) > 3 \ln n_\vee$, then with probability at least $1 - 1/n_\vee$, all co-degrees of a pair of vertices in blocks B_i and B_j fall in the range $\Big(p_{in}^{(i)} p_{out}(n_i - 1) + {p_{out}}^2(n - n_j - n_i) + p_{in}^{(i)} p_{out}(n_j - 1) - \sqrt{6(p_{in}^{(i)} p_{out} n_i + p_{out} p_{in}^{(i)} n_j + {p_{out}}^2(n - n_i - n_j)) \ln n_\vee}, p_{in}^{(i)} p_{out}(n_i - 1) + {p_{out}}^2(n - n_j - n_i) + p_{in}^{(i)} p_{out}(n_j - 1) + \sqrt{9(p_{in}^{(i)} p_{out} n_i + p_{out} p_{in}^{(i)} n_j + {p_{out}}^2(n - n_i - n_j)) \ln n_\vee}\Big)$.

Proof. For a pair of vertices $x \in B_i$ and $y \in B_j$, the co-degree $d_{x^{B_i}y^{B_j}}$ is the sum of $n-2$ independent random variables $X_1, ..., X_{n-2}$ with expectation $E(d_{x^{B_i}y^{B_i}}) = {p_{in}^{(i)}}^2(n_i - 2) + {p_{out}}^2(n - n_j)$. Applying Chernoff's inequality with the lower tail $\lambda = \sqrt{6({p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n-n_j))\ln n_i}$, we have

$$P\Big(d_{x^{B_i}y^{B_i}} - ({p_{in}^{(i)}}^2(n_i-2) + {p_{out}}^2(n-n_i)) < -\sqrt{6({p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n-n_i))\ln n_i}\Big) \tag{25}$$

$$\leq e^{-\frac{6({p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n-n_i))\ln n_i}{2({p_{in}^{(i)}}^2(n_i-2)+{p_{out}}^2(n-n_i))}} < \frac{1}{n_i^3}. \tag{26}$$

If ${p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n-n_i) > 4\ln n_i$, we apply Chernoff's inequality with the upper tail $\lambda = \sqrt{9({p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n-n_i))\ln n_i}$,

$$P\Big(d_{x^{B_i}y^{B_i}} - ({p_{in}^{(i)}}^2(n_i-2) + {p_{out}}^2(n-n_i)) > \sqrt{9({p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n-n_i))\ln n_i}\Big) \tag{27}$$

$$\leq e^{-\frac{9({p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n-n_i))\ln n_i}{2({p_{in}^{(i)}}^2(n_i-2)+{p_{out}}^2(n-n_i))+2/3\sqrt{9({p_{in}^{(i)}}^2 n_i + {p_{out}}^2(n-n_i))\ln n_i}}} < \frac{1}{n_i^3}. \tag{28}$$

Now the number of pairs is at most $n_i^2/2$. the sum of the probabilities of small events is at most $\frac{n_i^2}{2}\left(\frac{1}{n_i^3} + \frac{1}{n_i^3}\right) = \frac{1}{n_i}$.

For a pair of vertices $x \in B_i$ and $y \in B_j$, the co-degree $d_{x^{B_i}y^{B_j}}$ is the sum of $n-2$ independent random variables $X_1, ..., X_{n-2}$ with expectation $E(d_{x^{B_i}y^{B_j}}) = p_{in}^{(i)}p_{out}(n_i - 1) + {p_{out}}^2(n - n_j - n_i) + p_{in}^{(i)}p_{out}(n_j - 1)$. Applying Chernoff's inequality with the lower tail $\lambda = \sqrt{6(p_{in}^{(i)}p_{out}n_i + p_{out}p_{in}^{(i)}n_j + {p_{out}}^2(n-n_i-n_j))\ln n_\vee}$, we have

$$P\Big(d_{x^{B_1}y^{B_2}} - (p_{in}^{(i)}p_{out}(n_i-1) + {p_{out}}^2(n-n_j-n_i) + p_{in}^{(i)}p_{out}(n_j-1)) \tag{29}$$

$$< -\sqrt{6(p_{in}^{(i)}p_{out}n_i + p_{out}p_{in}^{(i)}n_j + {p_{out}}^2(n-n_i-n_j))\ln n_\vee})\Big) \tag{30}$$

$$\leq e^{-\frac{(6\ln n_\vee)(p_{in}^{(i)}p_{out}(n_i-1)+{p_{out}}^2(n-n_j-n_i)+p_{in}^{(i)}p_{out}(n_j-1))}{2(p_{in}^{(i)}p_{out}(n_i-1)+{p_{out}}^2(n-n_j-n_i)+p_{in}^{(i)}p_{out}(n_j-1))}} < \frac{1}{{n_\vee}^3}. \tag{31}$$

If $p_{in}^{(i)}p_{out}n_i + p_{out}p_{in}^{(i)}n_j + {p_{out}}^2(n-n_i-n_j) > 3\ln n_\vee$, we apply Chernoff's inequality with the upper tail $\lambda = \sqrt{9(p_{in}^{(i)}p_{out}n_i + p_{out}p_{in}^{(i)}n_j + {p_{out}}^2(n-n_i-n_j))\ln n_\vee}$,

$$P\Big(d_{x^{B_i}y^{B_j}} - (p_{in}^{(i)}p_{out}(n_i-1) + {p_{out}}^2(n-n_j-n_i) + p_{in}^{(i)}p_{out}(n_j-1)) \tag{32}$$

$$> \sqrt{9(p_{in}^{(i)}p_{out}n_i + p_{out}p_{in}^{(i)}n_j + {p_{out}}^2(n-n_i-n_j))\ln n_\vee}\Big) \tag{33}$$

$$\leq e^{-\frac{9(p_{in}^{(i)}p_{out}n_i + p_{out}p_{in}^{(i)}n_j + {p_{out}}^2(n-n_i-n_j))\ln n_\vee}{2(p_{in}^{(i)}p_{out}n_i + p_{out}p_{in}^{(i)}n_j + {p_{out}}^2(n-n_i-n_j))+2/3\sqrt{9(p_{in}^{(i)}p_{out}n_i + p_{out}p_{in}^{(i)}n_j + {p_{out}}^2(n-n_i-n_j))\ln n_\vee}}} \tag{34}$$

$$< \frac{1}{{n_\vee}^3}. \tag{35}$$

The number of pairs is at most ${n_\vee}^2$. the sum of the probabilities of small events is at most $({n_\vee}^2)\left(\frac{1}{{n_\vee}^3}\right) = \frac{2}{n_\vee}$. □

Referring to the lemma mentioned in Lin et al. (2011), we can construct the lower and upper bound of the Ricci curvature of graphs generated by the stochastic block model.

Lemma A.6. *Suppose that $\phi : \Gamma(x)\backslash N(y) \to \Gamma(y)\backslash N(x)$ is an injective mapping. Then we have*

$$\kappa(x,y) \geq 1 - \frac{1}{d_y} \sum_{u \in \Gamma(x) \backslash N(y)} d(u, \phi(u)) + \frac{1}{d_x} - \frac{3(d_y - d_x)}{d_y}.$$

$$\kappa(x,y) = \lim_{\alpha \to 1} \frac{1 - W(m_x^\alpha, m_y^\alpha)}{1-\alpha} \leq \frac{d_{xy}+1}{d_x} + \frac{1}{d_y}.$$

Lemma A.7. *For the lower bound, we will construct a matching M from $\Gamma(x)\backslash N(y)$ to $\Gamma(y)\backslash N(x)$ as follows. Let $U_0 = \Gamma(x)\backslash N(y)$ and $V_0 = \Gamma(y)\backslash N(x)$. Pick up a vertex $u_1 \in U_0$. Reveal the neighborhood of u_1 in V_0. Pick a vertex in the neighborhood, and denote it by v_1. Let $U_1 = U_0 \backslash \{u_1\}$ and $V_1 = V_0 \backslash \{v_1\}$ and continue this process. The process ends when $\Gamma(u_{i+1}) \cap V_i = \emptyset$. The probability that the maximum matching between U_0 and V_0 is at most k is less than*

$$\sum_{i=1}^{k} (1-p_{out})^{|V_0|-i} < \frac{1}{p}(1-p_{out})^{|V_0|-k} \leq \sqrt{\frac{n}{\ln n}} e^{-p_{out}(|V_0|-k)} \leq n e^{-p_{out}(|V_0|-k)}.$$

Choose $k = \lfloor |V_0| - (3\ln n)/p \rfloor$. With probability at least $1 - 1/n^2$, there is a matching M of size k between $\Gamma(x)\backslash N(y)$ and $\Gamma(y)\backslash N(x)$.

Now we extend the matching M to an injective mapping $\phi : \Gamma(x)\backslash N(y) \to \Gamma(y)\backslash N(x)$ arbitrarily. Applying Lemmas A.7, with probability at least $1 - 4/n$, we have

$$\begin{aligned}
\kappa(x,y) &\geq 1 - \frac{1}{d_y} \sum_{u \in \Gamma(x)\backslash N(y)} d(u, \phi(u)) + \frac{1}{d_x} - \frac{3(d_y - d_x)}{d_y} \\
&\geq 1 - \frac{1}{d_y}(k + 3(|V_0| - k)) + \frac{1}{d_x} - \frac{3(d_y - d_x)}{d_y} \\
&\geq \frac{d_{xy}}{d_y} - \frac{2(3\ln n/p_{out})}{d_y} - \frac{3(d_y - d_x)}{d_y}.
\end{aligned}$$

Proof. Given the range of the degrees and co-degrees, lower bound and upper bound of Ricci curvatures with respect to d_x and d_{xy}, we can prove the theorem **??** as follows.

For vertex $x \in B_i, y \in B_i$, given $p_{in}^{(i)} n_i + p_{out}(n - n_i) > \frac{8}{3} \ln n_i$, we have

$$\begin{aligned}
\kappa(x_{B_i}, y_{B_i}) &\leq \frac{(n_i - 2){p_{in}^{(i)}}^2 + (n - n_j)p_{out}^2 + \sqrt{9\ln n_1 (n_1 {p_{in}^{(i)}}^2 + n_2 p_{out}^2)} + 2}{(n_i - 1)p_{in}^{(i)} + (n - n_i)p_{out} - \sqrt{4(n_i p_{in}^{(i)} + (n - n_i)p_{out})\ln n_i}} \\
&= p_{in}^{(i)} + \mathcal{O}(\frac{\ln n_i}{n}).
\end{aligned}$$

For vertex $x \in B_i, y \in B_i$, given ${p_{in}^{(i)}}^2 n_i + {p_{out}}^2 (n - n_i) > 4 \ln n_i$, we have

$$
\begin{aligned}
\kappa(x_{B_i}, y_{B_i}) \geq & \frac{(n_i-2){p_{in}^{(i)}}^2 + (n-n_i){p_{out}}^2 - \sqrt{6\ln n_i(n_i{p_{in}^{(i)}}^2 + (n-n_i)p_{out}^2)}}{(n_i-1)p_{in}^{(i)} + (n-n_i)p_{out} + \sqrt{6\ln n_i(n_i p_{in}^{(i)} + (n-n_i)p_{out})}} \\
& - \frac{6\ln n/p_{out}}{(n_i-1)p_{in}^{(i)} + n_2 p_{out} + \sqrt{4\ln n_i(n_i p_{in}^{(i)} + (n-n_i)p_{out})}} \\
& - \frac{6\sqrt{6\ln n_i(n_i p_{in}^{(i)} + (n-n_i)p_{out})}}{(n_i-1)p_{in}^{(i)} + (n-n_i)p_{out} + \sqrt{4\ln n_i(n_i p_{in}^{(i)} + (n-n_i)p_{out})}} \\
\geq & \frac{(n_i-2){p_{in}^{(i)}}^2 + (n-n_i){p_{out}}^2}{(n_i-1)p_{in}^{(i)} + (n-n_i)p_{out}} - \mathcal{O}(\frac{\ln n}{p_{out}^2 n}) - \mathcal{O}(\sqrt{\frac{p_{in}^{(i)}\ln n_i}{p_{out}^2 n}}) \\
= & \frac{(n_i-2){p_{in}^{(i)}}^2 + (n-n_i){p_{out}}^2}{(n_i-1)p_{in}^{(i)} + (n-n_i)p_{out}} - \mathcal{O}(\frac{\ln n}{p_{out}^2 n}) - \mathcal{O}(\sqrt{\frac{p_{in}^{(i)}\ln n_i}{p_{out}^2 n}}).
\end{aligned}
$$

If ${p_{out}}^2 p_{in}^{(i)} \leq \frac{(\ln_n)^2}{n\ln n_i}$, we have $\kappa(x_{B_i}, y_{B_i}) = \frac{(n_i-2){p_{in}^{(i)}}^2 + (n-n_i){p_{out}}^2}{(n_i-1)p_{in}^{(i)} + (n-n_i)p_{out}} - \mathcal{O}(\frac{\ln n}{p_{out}^2 n})$.

For vertex $x \in B_i, y \in B_j$, let $n_\vee = \max\{n_i\}_{i=1}^M$, and $p_\vee = \max\left\{p_{in}^{(i)}\right\}_{i=1}^M$. If $p_{out} > \sqrt{(3\ln n_\vee)\Big/\left[\frac{p_{in}^{(i)}}{p_{out}}(n_i+n_j) + n - n_i - n_j\right]}$, we have

$$
\begin{aligned}
\kappa(x_{B_i}, y_{B_j}) \leq & \frac{3\sqrt{\ln n_\vee[(n_i-1)p_{in}^{(i)}p_{out} + (n-n_i-n_j)p_{out}^2 + (n_j-1)p_{in}^{(j)}p_{out}]}}{(n_i-1)p_{in}^{(i)} + n_j p_{out} - 2\sqrt{\ln n_\vee(n_i p_{in}^{(i)} + n_j p_{out})}} \\
& + \frac{1 + (n_i-1)p_{in}^{(i)}p_{out} + (n-n_i-n_j)p_{out}^2 + (n_j-1)p_{in}^{(j)}p_{out}}{(n_i-1)p_{in}^{(i)} + n_j p_{out} - 2\sqrt{\ln n_\vee(n_i p_{in}^{(i)} + n_j p_{out})}} \\
& + \frac{1}{(n_j-1)p_{in}^{(j)} + n_i p_{out} - 2\sqrt{\ln n_\vee(n_j p_{in}^{(j)} + n_i p_{out})}} \\
\leq & \frac{3\sqrt{\ln n_\vee[(n_i-1)p_{in}^{(i)}p_{out} + (n-n_i-n_j)p_{out}^2 + (n_j-1)p_{in}^{(j)}p_{out}]} + 2}{(n-1)p_{out} - 2\sqrt{np_\vee\ln n}} \\
& + \frac{(n_i-1)p_{in}^{(i)}p_{out} + (n-n_i-n_j)p_{out}^2 + (n_j-1)p_{in}^{(j)}p_{out}}{(n-1)p_{out} - 2\sqrt{np_\vee\ln n}} \\
\leq & \mathcal{O}(\sqrt{\frac{n_\vee(p_{in}^{(i)} + p_{in}^{(j)} + p_{out})\ln n_\vee}{(n-1)^2 p_{out}}}) + \frac{n_i+n_j-2}{n-1}p_\vee + \frac{n-n_i-n_j}{n-1}p_{out}
\end{aligned}
$$

□

Proof. We can prove the Lemma A.2 by the lower bound and upper bound of the within-community edges' OR curvature from Lemma A.1. □

Theorem A.8. *Combining the conclusion of statement 1 and statement 3 in Lemma A.1, if* $p_\vee < \frac{p_{in}^{(i)}n}{n_i+n_j}$, $p_{in}^{(i)} < \frac{p_{out}(n-n_i)(n+n_i+n_j)}{n^2-(n_i+n_j)(n-2n_i)}$ *and conditions given in Lemma A.1, almost surely, for any* $(x_{B_i}^*, y_{B_j}^*) \in \Delta((x_{B_i}, y_{B_i}))$, *we have*

$$
\max_{(x'_{B_i}, y'_{B_i}) \in \Delta((x_{B_i}, y_{B_i}))} |\kappa(x_{B_i}, y_{B_i}) - \kappa(x'_{B_i}, y'_{B_i})| < \kappa(x_{B_i}^*, y_{B_j}^*) - \kappa(x_{B_i}, y_{B_i}). \tag{36}
$$

Proof. We first get the lower bound of the OR curvature difference of within community edge $\kappa(x_{B_i}, y_{B_i})$ and between community edge $\kappa(x^*_{B_i}, y^*_{B_j})$, which can be obtained by the lower bound of $\kappa(x_{B_i}, y_{B_i})$ and upper bound of $\kappa(x^*_{B_i}, y^*_{B_j})$ from the Lemma A.1.

Given the upper bound of the OR curvature difference of edges in the same community, we can prove under certain conditions, the upper bound is smaller than the lower bound we obtained in the last step. Denote $r_i = n_i/n$ and $r_j = n_j/n$.

$$\kappa(x_{B_i}, y_{B_i}) - \kappa(x^*_{B_i}, y^*_{B_j}) - |\kappa(x_{B_i}, y_{B_i}) - \kappa(x'_{B_i}, y'_{B_i})| \tag{37}$$

$$= \frac{(n_i-2){p_{in}^{(i)}}^2 + (n-n_i){p_{out}}^2}{(n_i-1)p_{in}^{(i)} + (n-n_i)p_{out}} - \Big(\frac{n-n_i-n_j}{n-1}p_{out} + \frac{n_i-n_j-2}{n-1}p_\vee\Big) - \frac{p_{out}(p_{in}^{(i)} - p_{out})}{\frac{n_i-1}{n-n_i}p_{in}^{(i)} + p_{out}} \tag{38}$$

$$= \frac{r_i{p_{in}^{(i)}}^2 + (1-r_i){p_{out}}^2}{r_i p_{in}^{(i)} + (1-r_i)p_{out}} - p_{out} - (r_i+r_j)(p_\vee - p_{out}) - \frac{(1-r_i)p_{out}(p_{in}^{(i)} - p_{out})}{r_i p_{in}^{(i)} + (1-r_i)p_{out}} + \mathcal{O}(\frac{1}{n}) \tag{39}$$

$$= \frac{r_i{p_{in}^{(i)}}^2 + 2(1-r_i)p_{out}^2 - (1-r_i)p_{out}p_{in}^{(i)} - r_i p_{in}^{(i)}p_{out} - (1-r_i)p_{out}^2}{r_i p_{in}^{(i)} + (1-r_i)p_{out}} \tag{40}$$

$$- \frac{(r_i+r_j)r_i(p_\vee - p_{out})p_{in}^{(i)} - (r_i+r_j)(1-r_i)p_{out}(p-p_{out})}{r_i p_{in}^{(i)} + (1-r_i)p_{out}} + \mathcal{O}(\frac{1}{n}) \tag{41}$$

$$> \frac{p_{in}^{(i)} r_i(p_{in}^{(i)} - p_\vee(r_i+r_j)) + p_{out}(p_{out}(1-r_i)(1+r_i+r_j) - p_\vee(1-(r_i+r_j)(1-2r_i)))}{r_i p_{in}^{(i)} + (1-r_i)p_{out}} \tag{42}$$

$$+ \mathcal{O}(\frac{1}{n}) > 0 \tag{43}$$

□

Theorem A.9 (Probability of the ORG-sub's Expansion to Communities). *Given the graph G generated by Theorem A.8, we apply the ORG-sub Algorithm 1 to the graph G and take graph subsamples with size $\tilde{n}$ denoted by $G[S]$. Under the conditions in Theorem A.8, $\frac{\tilde{n}}{n} \to 1$, we can prove that, the probability of the final subgraph $G[S]$ taken by ORG-sub containing nodes in the community B_i is:*

$$P(v \in B_i \mid v \in G[S]) \to 1 \tag{44}$$

Proof. Without loss of generality, we show the theorem holds for a two-block stochastic block model (SBM).

We assume the graph is generated from a two-block stochastic block model (SBM), where the probability of an edge within a block is p_{in}, and the probability of an edge between the blocks is p_{out}. We denote the subsampled graph from the original graph G by $G[S]$, where S is a subset of the full vertices set V of graph G, and $G[S]$ is the subgraph induced by the subset S. Denote the probability matrix of generating the adjacency matrix A by $\mathbf{P}$:

$$\begin{aligned} \mathbf{P}^* &= \begin{bmatrix} p_{in}\mathbf{1}_{n_1}\mathbf{1}_{n_1}^T & p_{out}\mathbf{1}_{n_1}\mathbf{1}_{n_2}^T \\ p_{out}\mathbf{1}_{n_2}\mathbf{1}_{n_1}^T & p_{in}\mathbf{1}_{n_2}\mathbf{1}_{n_2}^T \end{bmatrix}, \\ \mathbf{P} &= \mathbf{P}^* - diag\left\{\mathbf{P}^*\right\}, \end{aligned} \tag{45}$$

where the size of the first community is n_1, the size of the second community is n_2, and the number of nodes of the graph is $n = n_1 + n_2$. The node set belonging to the community 1 is $B_1 = \{v_1, ..., v_{n_1}\}$, and the node set belonging to the community 2 is $B_2 = \{v_{n_1+1,...,n}\}$. Denote each entry of the probability matrix P by p_{ij}, then the observed A is generated from:

$$A_{ij} \overset{\text{i.i.d.}}{\sim} \text{Bernouli}(p_{ij}), \tag{46}$$

Define the block of the matrix A as $A^{o_1 o_2} = \{e_{ij} = (v_i, v_j) | v_i \in B_{o_1}, v_j \in B_{o_2}\}$, where $o_1, o_2 \in \{1, 2\}$.

We denote the edge subsampled at the l-th step by $e_{ij}^{(l)}$, and the neighboring edge set of edge e_{ij} by $\Delta(e_{ij})$. For step 1, given the edge subsampled in the initialization step $e_{ij}^{(0)}$, we have:

$$\begin{aligned}
P(e_{ij}^{(1)} \in A^{11}) =& \sum_{e_{ij}^{(0)} \in \Delta(e_{ij}^{(1)})} P(e_{ij}^{(1)} | e_{ij}^{(0)}) P(e_{ij}^{(0)}) \\
=& \sum_{e_{ij}^{(0)} \in \Delta(e_{ij}^{(1)}) \cap (A^{12} \cup A^{21})} P(e_{ij}^{(1)}) P(e_{ij}^{(0)}) \\
=& 2n_2 \cdot p_{in} \cdot \frac{2p_{out}}{n(n-1)} = \frac{4n_2 p_{in} p_{out}}{n(n-1)},
\end{aligned}$$

$$\begin{aligned}
P(e_{ij}^{(1)} \in A^{22}) =& \sum_{e_{ij}^{(0)} \in \Delta(e_{ij}^{(1)})} P(e_{ij}^{(1)} | e_{ij}^{(0)}) P(e_{ij}^{(0)}) \\
=& \sum_{e_{ij}^{(0)} \in \Delta(e_{ij}^{(1)}) \cap (A^{12} \cup A^{21})} P(e_{ij}^{(1)}) P(e_{ij}^{(0)}) \\
=& 2n_1 \cdot p_{in} \cdot \frac{2p_{out}}{n(n-1)} = \frac{4n_1 p_{in} p_{out}}{n(n-1)},
\end{aligned}$$

$$\begin{aligned}
P(e_{ij}^{(1)} \in A^{12}) = P(e_{ij}^{(1)} \in A^{21}) =& \sum_{e_{ij}^{(0)} \in \Delta(e_{ij}^{(1)})} P(e_{ij}^{(1)} | e_{ij}^{(0)}) P(e_{ij}^{(0)}) \\
=& \sum_{e_{ij}^{(0)} \in \Delta(e_{ij}^{(1)}) \cap (A^{11} \cup A^{22})} P(e_{ij}^{(1)}) P(e_{ij}^{(0)}) \\
=& (n_1 + n_2 - 2) \cdot p_{in} \cdot \frac{2p_{out}}{n(n-1)} \\
=& \frac{2(n_1 + n_2 - 2) p_{in} p_{out}}{n(n-1)}.
\end{aligned}$$

Iteratively, we get the probability of subsampling edge e_{ij} at the l-th step:

$$\begin{aligned}
P(e_{ij}^{(l)} \in A^{11}) =& \sum_{e_{ij}^{(l-1)} \in \Delta(e_{ij}^{(l)})} P(e_{ij}^{(l)} | e_{ij}^{(l-1)}) P(e_{ij}^{(l-1)}) \\
=& \sum_{e_{ij}^{(l-1)} \in \Delta(e_{ij}^{(l)}) \cap (A^{12} \cup A^{21})} P(e_{ij}^{(l)}) P(e_{ij}^{(l-1)}),
\end{aligned}$$

$$\begin{aligned}
P(e_{ij}^{(l)} \in A^{22}) =& \sum_{e_{ij}^{(l-1)} \in \Delta(e_{ij}^{(l)})} P(e_{ij}^{(l)} | e_{ij}^{(l-1)}) P(e_{ij}^{(l-1)}) \\
=& \sum_{e_{ij}^{(l-1)} \in \Delta(e_{ij}^{(l)}) \cap (A^{12} \cup A^{21})} P(e_{ij}^{(l)}) P(e_{ij}^{(l-1)}),
\end{aligned}$$

$$\begin{aligned}
P(e_{ij}^{(l)} \in A^{12}) = P(e_{ij}^{(l)} \in A^{21}) =& \sum_{e_{ij}^{(l-1)} \in \Delta(e_{ij}^{(l)})} P(e_{ij}^{(l)} | e_{ij}^{(l-1)}) P(e_{ij}^{(l-1)}) \\
=& \sum_{e_{ij}^{(l-1)} \in \Delta(e_{ij}^{(l)}) \cap (A^{11} \cup A^{22})} P(e_{ij}^{(l)}) P(e_{ij}^{(l-1)}).
\end{aligned}$$

By the above equations, for each step the probability of subsampling an edge in a block is a constant. After subsampling one edge in each step, the nodes connected by the subsampled edges are

subsampled into node set S. We denote the probability of adding node $\tilde{v}$ at the step l belonging to community B_o, where $o \in \{1, 2\}$, by $P(\tilde{v}^{(l)} \in B_o)$. After $\tilde{n}$ steps, the probability of subsampling node $\tilde{v}$ from the node set B_1 is:

$$\begin{aligned} P(\tilde{v} \in B_1) =& 1 - P(\tilde{v}^{(1)} \notin B_1) \cdot P(\tilde{v}^{(2)} \notin B_1) \cdots P(\tilde{v}^{(\tilde{n})} \notin B_1) \\ =& 1 - P(e_{ij}^{(0)} \in A^{22}) \cdot P(e_{ij}^{(1)} \in A^{22}) \cdots P(e_{ij}^{(\tilde{n}-1)} \in A^{22}) \end{aligned}$$

Analogously, the probability of subsampling node $\tilde{v}$ belonging to the node set B_2 after n steps is:

$$\begin{aligned} P(\tilde{v} \in B_2) =& 1 - P(\tilde{v}^{(1)} \notin B_2) \cdot P(\tilde{v}^{(2)} \notin B_2) \cdots P(\tilde{v}^{(\tilde{n})} \notin B_2) \\ =& 1 - P(e_{ij}^{(0)} \in A^{11}) \cdot P(e_{ij}^{(1)} \in A^{11}) \cdots P(e_{ij}^{(\tilde{n}-1)} \in A^{11}) \end{aligned}$$

As the subsampler expand the subgraph, we can prove that the probability of the proposed ORG-sub algorithm taking all communities into the final subgraph converges to one under the two-stochastic-block model.

In general cases, the graph is generated by SBM with multiple communities, we can always think of the community that the subsampler has never been to as one block, and other communities as the other block. Given the conclusion under the two-stochastic-block model, the ORG-sub can gradually expand the subsampled graph to the block we have never been to. Thus, we can prove the ORG-sub can expand to any community under a multiple-stochastic-block model. □

Corollary A.9.1. *If the subsample with size $\tilde{n}$ satisfies $\frac{\tilde{n}\rho}{\log \tilde{n}} \to \infty$, where ρ controls the expected node degree of subsampled graph, and as the subsampling times $r \to \infty$, the probability the subsampled graph underestimates the true M is:*

$$Pr(\frac{1}{r}\sum_{i=1}^{r} \hat{M}(G[S_i]) < M) \to 0, \tag{47}$$

Proof. From the theorem A.9, we know that the true number of communities $M(G[S])$ of subgraph $G[S]$ is the same as the true number of communities M of the original graph G. Theorem 3 of Li et al. (2020) proves the one-sided consistency of estimating the number of communities of the stochastic block model under the following assumptions. First, the expected node degree $\lambda_n = n\rho$ has the order $\frac{n}{\log n}$, where n is the number of nodes of the graph G. Second, there exists a constant γ such that $\min_k n_k \geq \gamma N$, where n_k is the size of the k-th community. Since our subsampled graph satisfies $\frac{\tilde{n}\rho}{\log \tilde{n}}$, we can easily verify that the subsampled subgraph satisfies the assumptions in Theorem 3 of Li et al. (2020). The estimate of the number of communities $\hat{M}_i$ corresponding to S_i, the i-th subsample, satisfies:

$$Pr(\hat{M}_i < M(G[S_i])) \to 0,\ Pr(\hat{M}_i < M) \to 0, \tag{48}$$

As the replications of the subsamples increase, it's trivial to show:

$$Pr(\frac{1}{r}\sum_{i=1}^{r} \hat{M}_i < M) \to 0. \tag{49}$$

□

B Details of Experiments

We evaluate the performance of our algorithm on both synthetic datasets and real-world datasets using the metrics presented in the main context. We also compare the proposed method with seven benchmark exploration-based graph subsampling methods, including Metropolis Hasting Random Walk Sampler (MHRW), Forest Fire Sampler (FFS), Snowball Sampler, Community Structure Expansion Sampler (CSE), degree-based node sampler (DBN), Random Walk Sampler (RW) and Multi-dimensional Random Walk Sampler (MDRW). We set the hyper-parameters of the seven methods as the default in the package *Little Ball of Fur*. The rejection constraint of the MHRW method is set as 1; the burning probability of FFS is 0.4; the bound on the degree of Snowball Sampler is set as 50. The hyper-parameter used to calculate the Ricci curvature of graphs is set as $\alpha = 0.5$. The times of subsampling r in Algorithm 2 for estimation of the number of communities is set as 3, which is enough to get a stable estimation result. All the experiments are conducted on a workstation with an 18-core NVIDIA Tesla V100 GPU (3.00 GHz).

B.1 EXPERIMENTS ON SYNTHETIC DATASETS

B.1.1 PARAMETERS SETTINGS

We use stochastic block models (SBM), and degree corrected block models (DCBM), which can assign the community distribution to each node. To create graphs with unbalanced communities, we set the community proportion as $(3/4, 1/10, 1/12, 1/15)$ with 900 nodes. The out-in-ratio controls the ratio of an edge's probability of between-communities and within-community. A higher out-in-ratio represents a noisier graph, i.e., harder to distinguish the community from the graph. We set the probability of an edge within a block as 0.8 and vary the probability of an edge between blocks (pout) for both SBM and DCBM ($\{0.06, 0.08, 0.10, 0.12\}$). The higher the pout value (higher the out-in-ratio) is, the noisier the subsampled graph is. We also change the subsampling proportion ($\{0.1, 0.12, 0.14, 0.16\}$) from low to high for both block models to observe how our method performs compared with others as the settings change. The degree corrected model corrects the node degree using a power-law distribution. More parameters need to be set before generating the graphs following DCBM. We set the average node degree as 40, with is consistent with the setting in Li et al. (2020), in which the assumptions in our algorithm hold. The node degree follows a power-law distribution with the lower bound as 1 and the scaling parameter as 5.

B.1.2 RESULTS OF SYNTHETIC DATASETS

Pout	Prop	ORG-sub	MHRW	CSE	FFS	SnowBall	DBN	RW	MDRW
0.06	0.10	**1.50**	2.83	2.83	2.73	1.53	2.60	2.77	3.00
	0.12	**1.40**	2.77	2.77	2.60	1.90	2.63	2.73	3.00
	0.14	**1.57**	2.53	2.73	2.33	1.80	2.57	2.57	3.00
	0.16	**1.63**	2.50	2.80	2.23	1.80	2.30	2.53	3.00
0.08	0.10	**1.30**	2.97	2.83	2.70	1.53	2.77	2.80	3.00
	0.12	**1.57**	2.77	2.83	2.77	1.80	2.53	2.53	3.00
	0.14	**1.57**	2.90	2.83	2.70	1.80	2.40	2.67	3.00
	0.16	**1.63**	2.77	2.63	2.53	1.80	2.40	2.53	3.00
0.10	0.10	**0.93**	2.83	2.93	2.83	1.37	2.63	2.80	3.00
	0.12	**1.63**	2.90	2.87	2.60	1.83	2.50	2.73	3.00
	0.14	**1.53**	2.80	2.77	2.70	1.93	2.47	2.77	3.00
	0.16	**1.73**	2.83	2.77	2.57	1.70	2.57	2.63	3.00
0.12	0.10	**0.93**	2.93	2.90	2.73	1.6	2.77	2.93	3.00
	0.12	**1.40**	3.00	2.77	2.73	1.60	2.73	2.80	3.00
	0.14	**1.57**	2.93	2.73	2.47	1.80	2.73	2.77	3.00
	0.16	**1.60**	2.90	2.67	2.50	1.87	2.60	2.67	3.00

Table A.2: **DCBM**: Error of the estimation of the number of communities for different subsampling methods under different settings.

We use synthetic datasets with hidden community structures to evaluate the performance of our method. We replicate the experiments 30 times under each setting and compare the performance of the error of estimation of the number of communities and the computation time. The average of 30 replications of each setting is recorded in Table A.2 and Table A.5. In these tables, Column **Prop** denotes the subsampling proportion, and column **Pout** denotes the probability of an edge between communities. The rest of the columns are the results of different subsampling methods. The performance of our algorithm is better than that of other methods. As for the results for SBM in Table A.5, the error of estimating the number of communities decreases as the subsampling proportion increases, and the error increases as the observed graphs are noisier (pout increases). As for the computational time, we observe that the estimation time of the full sample is two orders of magnitude larger than the time of subsampling and estimation. As the subsampling proportion increases, the computational time of subsampling methods increases. Though our method's computational time is slightly larger than our subsampling methods, our method has better accuracy.

Pout	Prop	ORG-sub	MHRW	CSE	FFS	SnowBall	DBN	RW	MDRW
0.06	0.10	**0.47**	1.70	1.93	2.00	2.33	2.00	2.00	3.00
	0.12	**0.33**	1.67	1.80	1.93	2.20	2.00	2.00	3.00
	0.14	**0.10**	1.77	1.77	1.93	2.13	2.00	2.00	3.00
	0.16	**0.00**	1.73	1.30	1.97	2.07	2.00	2.00	3.00
0.08	0.10	**0.63**	1.77	1.60	1.87	2.23	2.00	2.00	3.00
	0.12	**0.37**	1.73	1.70	1.83	2.17	2.00	2.00	3.00
	0.14	**0.03**	1.87	1.53	1.80	2.07	2.00	2.00	3.00
	0.16	**0.07**	1.83	1.67	1.83	2.03	2.00	2.00	3.00
0.10	0.10	**0.80**	1.77	1.73	1.93	2.20	2.00	1.97	3.00
	0.12	**0.33**	1.73	1.67	2.00	2.17	2.00	1.93	3.00
	0.14	**0.13**	1.80	1.63	1.90	2.03	2.00	1.93	3.00
	0.16	**0.03**	1.73	1.60	1.97	2.03	2.00	1.93	3.00
0.12	0.10	**0.73**	1.67	1.83	1.87	2.07	2.00	2.00	3.00
	0.12	**0.53**	1.60	1.57	1.93	2.03	2.00	2.00	3.00
	0.14	**0.33**	1.63	1.53	1.93	2.00	2.00	2.00	3.00
	0.16	**0.20**	1.63	1.40	1.97	2.00	2.00	2.00	3.00

Table A.5: **SBM**: Error of the estimation of the number of communities for different subsampling methods under different settings.

B.2 Experiments on Real-world Datasets

B.2.1 Description of the Datasets

We use four real-world datasets to evaluate the performance of our method as well. These datasets are widely used as benchmark datasets in the study of community detection for graphs. They can be easily found at KONECT, UCI network data repository and Stanford Network Analysis Project Rossi & Ahmed (2015); Rozemberczki et al. (2020a); Sen et al. (2008). All datasets are considered undirected and unweighted graphs, and all self-loop edges and isolated nodes are removed. More details about the datasets are as follow.

Polblogs: The political blogs network dataset was collected by Adamic and Glance in 2005. A network is constructed among all the posts published by liberal or conservative bloggers (two communities). Each node represents one post. An edge connects two nodes if one of them is cited by the other.

Polbooks: This is a network of books about US politics published around the 2004 presidential election and sold by online bookseller *Amazon.com*. All the books are divided into four communities by NI-LPA. Edges between books represent frequent co-purchasing of books by the same buyers.

Facebook: This is an ego-network dataset of the "friend circles" of one anonymous user on Facebook. The network forms friend circles such as family members, high school friends or other friends that are "hand-labeled" by the user.

Cora: The Cora dataset describes the citation relationship among scientific publications classified into seven classes. After preprocessing, there remain 1,873 nodes and 5,033 links.

In Table A.7, Node, Edge, and Community record the number of the nodes, edges, and communities of the graph, respectively. Sparsity records how dense the graph is, calculated by the ratio of the number of edges in the actual and complete graphs. CCF is the clustering coefficient of the whole graph. These datasets cover different levels of size and density which guarantee a robust test for the performance of our method.

Dataset	Node	Edge	Sparsity	Community	CCF
Polblogs	1331	19182	0.0108	2	0.105
Polbooks	102	429	0.0416	3	0.153
Facebook	736	13847	0.0.025	11	0.114
Cora	1873	5033	0.0014	7	0.038

Table A.7: Key features of the real-world datasets

B.2.2 RESULTS OF REAL-WORLD DATASETS

The comparisons of errors in estimating the number of communities are presented in the main context. The computation time is also obtained. Similar to the synthetic datasets, the estimation time of the full sample is two orders of magnitude larger than the time of subsampling (including the time for computing OR curvature for each edge in the graph) and estimation.

The result also shows that the computation time for estimating the number of communities using the subsampling method is much shorter than that using the full dataset. The complexity of all methods is influenced by the size of the graph, especially by the number of nodes. It is no wonder that the computation time of all methods in estimating the number of communities for dataset *Cora* is much longer than other datasets.