

---

# Streaming Belief Propagation for Community Detection

---

**Yuchen Wu**  
Stanford University  
wuyc14@stanford.edu

**Jakab Tardos**  
EPFL  
jakab.tardos@epfl.ch

**MohammadHossein Bateni**  
Google Research  
bateni@google.com

**André Linhares**  
Google Research  
linhares@google.com

**Filipe Miguel Gonçalves de Almeida**  
Google Research  
filipea@google.com

**Andrea Montanari**  
Stanford University  
montanari@stanford.edu

**Ashkan Norouzi-Fard**  
Google Research  
ashkannorouzi@google.com

## Abstract

The community detection problem requires to cluster the nodes of a network into a small number of well-connected ‘communities’. There has been substantial recent progress in characterizing the fundamental statistical limits of community detection under simple stochastic block models. However, in real-world applications, the network structure is typically dynamic, with nodes that join over time. In this setting, we would like a detection algorithm to perform only a limited number of updates at each node arrival. While standard voting approaches satisfy this constraint, it is unclear whether they exploit the network information optimally. We introduce a simple model for networks growing over time which we refer to as *streaming stochastic block model* (StSBM). Within this model, we prove that voting algorithms have fundamental limitations. We also develop a streaming belief-propagation (STREAMBP) approach, for which we prove optimality in certain regimes. We validate our theoretical findings on synthetic and real data.

## 1 Introduction

Given a single realization of a network  $G = (V, E)$ , the community detection problem requires to find a partition of its vertices into a small number of clusters or ‘communities’ [GN02, MO04, JTZ04, For10, PKVS12, JYL<sup>+</sup>18].

Numerous methods have been developed for community detection on static networks [GN02, CNM04, GA05, RCC<sup>+</sup>04, LF09, VLBB08, WZCX18]. However, the network structure evolves over time in most applications. For instance, in social networks new users can join the network; in online commerce new products can be listed; see [WGKM18, GLZ08, KvB<sup>+</sup>14] for other examples. In such dynamic settings, it is desirable to have algorithms that perform only a limited number of operations each time a node joins or leaves, possibly revising the labels of nodes in a neighborhood of the new node. (These notions will be formalized below.) Several groups have developed algorithms of this type in the recent past [HS12, CSG16, ZWCY19, MKV<sup>+</sup>20]. The present paper aims at characterizing the fundamental statistical limits of community detection in the dynamic setting, and proposing algorithms that achieve those limits.

As usual, establishing fundamental statistical limits requires introducing a statistical model for the network  $G = (V, E)$ . In the case of static networks, precise characterizations have only been achieved recently, and almost uniquely for a simple network model, namely the stochastic block model (SBM) [HLL83, ABFX08, KN11, RCY<sup>+</sup>11, DKMZ11, Mas14, ABH15, AS15, MNS18]. In this paper we build on these recent advances and study a dynamic generalization of the SBM, which we refer to as *streaming SBM* (StSBM).

The SBM can be defined as follows. Each vertex  $v$  is given a label  $\tau(v)$  drawn independently from a fixed distribution over the set  $[k] = \{1, 2, \dots, k\}$ . Edges are conditionally independent given vertex labels. Two vertices  $u, v$  are connected by an edge with probability  $W_{\tau(u), \tau(v)}$ . The analyst is given a realization of the graph and required to estimate the labels  $\tau$ . We will assume in addition that the analyst has access to some noisy version of the vertex labels, denoted by  $\tilde{\tau} = (\tilde{\tau}(v))_{v \in V}$ . This is a mathematically convenient generalization: the special case in which no noisy observations  $\tilde{\tau}$  are available can be captured by letting  $\tilde{\tau}$  be independent of  $\tau$ . Further, such a generalization is useful to model cases in which covariate information is available at the nodes [MX16].

Informally, the *streaming SBM* (StSBM) is a version of SBM in which nodes are revealed one at a time in random order (see below for a formal definition). In order to model the notion that only a limited number of updates is performed each time a new node joins the network, we introduce a class of ‘local streaming algorithms.’ These encompass several algorithms in earlier literature, e.g. [ZGL03, CSG16]. Our definition is inspired and motivated by the more classical definition of local algorithms for static graphs. Local algorithms output estimates for one vertex based only on a small neighborhood around it, and thus scale well to large graphs; see [Suo13] for a survey. A substantial literature studies the behavior of local algorithms for sparse graphs [GT12, HLS14, GS14, Mon15, MX16, FM17].

Our results focus on the sparse regime in which the graph’s average degree is bounded. This is the most challenging regime for the SBM, and it is also relevant for real-world applications where networks are usually sparse. We present the following contributions:

*Fundamental limitations of local streaming algorithms.* We prove that, in the absence of side information, in streaming symmetric SBM (introduced in Section 2), local streaming algorithms (introduced in Section 3) do not achieve any non-trivial reconstruction: their accuracy is asymptotically the same as random guessing; see Corollary 1. This holds despite the fact that there exist polynomial time non-local algorithms that achieve significantly better accuracy. From a practical viewpoint, this indicates that methods with a small ‘locality radius’ (the range over which the algorithm updates its estimates) are ineffective at aggregating information: they perform poorly unless strong local side information is available.

*Optimality of streaming belief propagation.* On the positive side, in Section 4 we define a streaming version of belief propagation (BP), a local streaming algorithm that we call STREAMBP, parameterized by a locality radius  $R$ . We prove that, for any non-vanishing amount of side information, STREAMBP achieves the same reconstruction accuracy as offline BP; see Theorem 2. The latter is in turn conjectured to be the optimal offline polynomial-time algorithm [DKMZ11, Abb17, HS17]. Under this conjecture, there is no loss of performance in restricting to local streaming algorithms as long as (1) local side information is available; (2) the locality radius is sufficiently large; and (3) information is aggregated optimally via STREAMBP.

Let us emphasize that we do not claim (nor do we expect) STREAMBP to outperform offline BP. We use offline BP as an ‘oracle’ benchmark (as it has the full graph information available, and is not constrained to act in a streaming fashion).

*Implementation and numerical experiments.* In Section 5 and Appendix A–C, we validate our results both on synthetic data, generated according to the StSBM, and on real datasets. Our empirical results are consistent with the theory; in particular, STREAMBP substantially outperforms simple voting methods. However, we observe that it can behave poorly with large locality radius  $R$ . In order to overcome this problem, we introduce a ‘bounded distance’ version of STREAMBP, called STREAMBP\*, which appears to be more robust. (STREAMBP\* can be shown to enjoy the same theoretical guarantees as STREAMBP.)

## 2 Streaming stochastic block model

In this section we present a formal definition of the proposed model. The streaming stochastic block model is a probability distribution  $\text{StSBM}(n, k, p, W, \alpha)$  over triples  $(\tau, \tilde{\tau}, \mathbf{G})$  where  $\tau \in [k]^n$  is a vector of labels (here  $[k] \triangleq \{1, \dots, k\}$ ),  $\tilde{\tau} \in [k]^n$  are noisy observations of the labels  $\tau$ , and  $\mathbf{G} = (G(0), G(1), \dots, G(n))$  is a sequence of undirected graphs. Here  $G(t) = (V(t), E(t))$  is a graph over  $|V(t)| = t$  vertices and, for each  $0 \leq t \leq n-1$ ,  $V(t) \subseteq V(t+1)$  and  $E(t) \subseteq E(t+1)$ . We will assume, without loss of generality, that  $V(n) = [n]$ , and interpret  $\tau(v)$  as the label associated to vertex  $v \in [n]$ . For each  $0 \leq t \leq n-1$ ,  $G(t)$  is the subgraph induced in  $G(t+1)$  by  $V(t)$ ; equivalently, all edges in  $E(t+1) \setminus E(t)$  are incident to the unique vertex in  $V(t+1) \setminus V(t)$ .

The distribution  $\text{StSBM}(n, k, p, W, \alpha)$  is parameterized by a scalar  $\alpha \in [0, \frac{k-1}{k}]$ , a probability vector  $p = (p_1, \dots, p_k) \in \Delta_k \triangleq \{x \in [0, 1]^k, \langle x, \mathbf{1} \rangle = 1\}$ , and a symmetric matrix  $W \in [0, 1]^{k \times k}$ . We draw the coordinates of  $\tau$  independently with distribution  $p$ , and set  $\tilde{\tau}(v) = \tau(v)$  with probability  $1 - \alpha$ , and  $\tilde{\tau}(v) \sim \text{Unif}([k] \setminus \{\tau(v)\})$  otherwise, independently across vertices:

$$\mathbb{P}(\tau(v) = s) = p_s, \quad \mathbb{P}(\tilde{\tau}(v) = s_1 \mid \tau(v) = s_0) = \begin{cases} 1 - \alpha & \text{if } s_1 = s_0, \\ \alpha/(k-1) & \text{if } s_1 \neq s_0. \end{cases}$$

We then construct  $G(n)$  by generating conditionally independent edges, given  $(\tau, \tilde{\tau})$ , with

$$\mathbb{P}((u, v) \in E(n) \mid \tau, \tilde{\tau}) = W_{\tau(u), \tau(v)}. \quad (1)$$

Note that the labels  $\tilde{\tau}$  provide noisy ‘side information’ about the true labels  $\tau$ . This information  $\tilde{\tau}$  is conditionally independent of the graph  $G(n)$  given  $\tau$ . Finally we generate the graph sequence  $\mathbf{G}$  by choosing a uniformly random permutation of the vertices  $(v(1), v(2), \dots, v(n))$  and setting  $V(t) = \{v(1), \dots, v(t)\}$  and  $G(t)$  to the graph induced by  $V(t)$ . If  $v = v(t)$ , then we define  $t$  as the arrival order of vertex  $v$ . Note that, for each  $t$ ,  $G(t)$  is distributed according to a standard SBM with  $t$  vertices:  $G(t) \sim \text{SBM}(t, k, p, W)$ .

An equivalent description is that (conditional on  $\tau, \tilde{\tau}$ )  $\text{StSBM}(n, k, p, W, \alpha)$  defines a Markov chain over graphs. The new graph  $G(t+1)$  is generated from  $G(t)$  by drawing the vertex  $v(t+1)$  uniformly at random from  $[n] \setminus V(t)$ , and then the edges  $(u, v(t+1))$ ,  $u \in V(t)$ , independently with probabilities given by Equation (1).

We are interested in the behavior of large graphs with bounded average degree. In order to focus on this regime, we will consider  $n \rightarrow \infty$  and  $W = W^{(n)} \rightarrow 0$  with  $W^{(n)} = W_0/n$  for a matrix  $W_0 \in \mathbb{R}_{\geq 0}^{k \times k}$  independent of  $n$ .

A case of special interest is the streaming symmetric SBM,  $\text{StSSBM}(n, k, a, b, \alpha)$ , which corresponds to taking  $p = (1/k, \dots, 1/k)$  and  $W_0$  having diagonal elements  $a$  and non-diagonal elements  $b$ . Finally, the case  $\alpha = (k-1)/k$  corresponds to pure noise  $\tilde{\tau}$ : in this case we can drop  $\tilde{\tau}$  from the observations and we will drop  $\alpha$  from the distribution parameters.

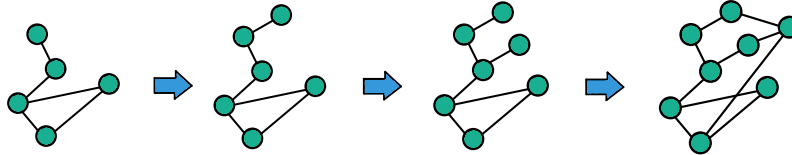


Figure 1: Evolving process of StSBM.

### 2.1 Definitions and notations

For two nodes  $v, v' \in V(t)$ , we denote by  $d_t(v, v')$  their graph distance in  $G(t)$ , i.e., the length of the shortest path in  $G(t)$  connecting  $v$  and  $v'$ , with  $d_t(v, v') = \infty$  if no such path exists. We also write  $d(v, v') = d_n(v, v')$  for the graph distance in  $G(n)$ . For  $v \in V(n)$  and  $R \in \mathbb{N}^+$ , let  $\mathbf{B}_R^t(v) = (V_R^t(v), E_R^t(v))$  denote the ball of radius  $R$  in  $G(t)$  centered at  $v$ , i.e., the subgraph induced in  $G(t)$  by nodes  $V_R^t(v) \triangleq \{v' \in V(t) : d_t(v, v') \leq R\}$  and edges  $(v_1, v_2) \in E(t)$  with

$v_1, v_2 \in V_R^t(v)$ . Furthermore, let  $D_R^t(v) \triangleq \{v' \in V(t) : d(v, v') = R\}$ . Throughout the paper, unless otherwise stated, we assume  $(v_1, v_2)$  is an undirected edge.

We consider an algorithm  $\mathcal{A}$  that takes as input the graph  $G(n)$  and side information  $\tilde{\tau}$ , and for each  $v \in V(n)$  outputs  $\mathcal{A}(v; G(n), \tilde{\tau}) \in [k]$  as an estimate for  $\tau(v)$ . Note that we always assume the arrival orders of vertices are observed (i.e., by observing  $v \in [n]$  we also observe the unique  $t \in [n]$  such that  $v = v(t)$ ), thus  $G(n)$  contains the arrival order of its vertices. We define the estimation accuracy of algorithm  $\mathcal{A}$  as

$$Q_n(\mathcal{A}) \triangleq \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{v \in V(n)} \mathbb{1}(\mathcal{A}(v; G(n), \tilde{\tau}) = \pi \circ \tau(v)) \right]. \quad (2)$$

Here  $\mathfrak{S}_k$  is the group of permutations over  $[k]$  and the expectation is with respect to  $G(n)$ ,  $\tau$ ,  $\tilde{\tau}$ , and the randomness of the algorithm (if  $\mathcal{A}$  is randomized).

### 3 Local streaming algorithms

In this section we introduce local streaming algorithms, which are a generalization of local algorithms in the dynamic network setting. An *R-local streaming algorithm* is an algorithm that at each vertex keeps some information available to that vertex. As a new vertex  $v(t)$  joins, information within the  $R$ -neighborhood  $B_R^t(v(t))$  is pulled. An estimate for  $\tau(v)$  is constructed based on information available to  $v$ . In order to accommodate randomized algorithms we assume that random variables  $(\xi_v)_{v \in V(n)} \stackrel{iid}{\sim} \text{Unif}([0, 1])$ , independent of the graph, are part of the local information available to the algorithm.

As an example, we can consider a simple voting algorithm. At each step  $t$ , this algorithm keeps in memory the current estimates  $\hat{\tau}(v) \in [k]$  for all  $v \in V(t)$ . As a new vertex  $v(t)$  joins, its estimated label is determined according to

$$\hat{\tau}(v(t)) = \operatorname{argmax}_{s \in [k]} \pi_t(s) \quad \pi_t(s) = \delta \mathbb{1}(s = \tilde{\tau}(v(t))) + \sum_{(v(t), u) \in E(t)} \mathbb{1}(s = \hat{\tau}(u)). \quad (3)$$

In words, the estimated label at  $v(t)$  is the winner of a voting procedure, where the neighbors of  $v(t)$  contribute one vote each, while the side information at  $v(t)$  contributes  $\delta$  votes.

For  $v \in V(n)$  and  $t \in [n]$ , we denote the subgraph accessible to  $v$  at time  $t$  by  $\mathcal{G}_v^t = (\mathcal{V}_v^t, \mathcal{E}_v^t)$ , with initialization  $\mathcal{G}_v^0 = (\{v\}, \emptyset)$ . At time  $t$ , we conduct the following updates:

$$\mathcal{V}_v^t \triangleq \begin{cases} \bigcup_{v' \in V_R^t(v(t))} \mathcal{V}_{v'}^{t-1} & \text{for } v \in V_R^t(v(t)), \\ \mathcal{V}_v^{t-1} & \text{for } v \notin V_R^t(v(t)). \end{cases}$$

We let  $\mathcal{G}_v^t$  be the subgraph induced in  $G(t)$  by  $\mathcal{V}_v^t$ , and denote by  $\bar{\mathcal{G}}_v^t = (\bar{\mathcal{V}}_v^t, \bar{\mathcal{E}}_v^t)$  the corresponding labeled graph with vertex labels  $\tilde{\tau}$  and randomness  $\xi$ . Namely  $\bar{\mathcal{V}}_v^t \triangleq \{(v', \tilde{\tau}(v'), \xi_{v'}) : v' \in \mathcal{V}_v^t\}$ ,  $\bar{\mathcal{E}}_v^t \triangleq \mathcal{E}_v^t$ . At time  $t$ , all nodes in the  $R$  neighborhood of  $v(t)$  share the same updated subgraph  $\mathcal{G}_{v(t)}^t$  while the rest of the subgraphs stay unchanged. Let us emphasize that the ‘neighborhoods’  $\mathcal{G}_v^t$  are not symmetric, in the sense that we can have  $v_1 \in \mathcal{G}_{v_2}^t$  but  $v_2 \notin \mathcal{G}_{v_1}^t$ .

**Definition 1** (*R*-local streaming algorithm). *An algorithm  $\mathcal{A}$  is an *R*-local streaming algorithm if, at each time  $t$  and for each vertex  $v \in V(t)$ , it outputs an estimate of  $\tau(v)$  denoted by  $\mathcal{A}^t(v; G(t), \tilde{\tau}) \in [k]$ , which is a function uniquely of  $\bar{\mathcal{G}}_v^t$ .*

Note that this class includes as special cases voting algorithms (which correspond to  $R = 1$ ) but also a broad class of other approaches. We will compare *R*-local streaming algorithms with *R*-local algorithms (non-streaming). In order to define the latter, given a neighborhood  $B_R^t(v)$ , we define the corresponding labeled graph as  $\bar{B}_R^t(v) \triangleq (\bar{V}_R^t(v), E_R^t(v))$ , with  $\bar{V}_R^t(v) \triangleq \{(v', \tilde{\tau}(v'), \xi_{v'}) : v' \in V_R^t(v(t))\}$ .

**Definition 2** (*R*-local algorithm). *An algorithm  $\mathcal{A}$  is an *R*-local algorithm if, at each time  $t$  and for each vertex  $v \in V(t)$ , it outputs an estimate of  $\tau(v)$  denoted by  $\mathcal{A}^t(v; G(t), \tilde{\tau}) \in [k]$ , which is a function uniquely of  $\bar{B}_R^t(v)$ .*

For simplicity, define the final output of an algorithm  $\mathcal{A}$  by  $\mathcal{A}(v; G(n), \tilde{\tau}) \triangleq \mathcal{A}^n(v; G(n), \tilde{\tau})$ . The next theorem states that, under StSSBM, any local streaming algorithm with fixed radius behaves asymptotically as a local algorithm. Here we focus on StSSBM—extension to asymmetric cases is straightforward.

**Theorem 1.** *Let  $\mathcal{G}$  be distributed according to  $\text{StSSBM}(n, k, a, b, \alpha)$ , and  $v_0 \sim \text{Unif}([n])$  be a vertex chosen independently of  $\mathcal{G}$ . Then, for any  $\epsilon > 0$ , there exist  $n_\epsilon, r_\epsilon \in \mathbb{N}^+$ , such that for every  $n \geq n_\epsilon$  with probability at least  $1 - \epsilon$ , the following properties hold: (1)  $\mathcal{G}_{v_0}^n$  is a subgraph of  $\mathcal{B}_{r_\epsilon}^n(v_0)$ ; and (2)  $v_0$  does not belong to  $\mathcal{G}_v^n$  for any  $v \in V(n) \setminus V_{r_\epsilon}^n(v_0)$ .*

Under the symmetric SBM, local algorithms without side information cannot achieve non-trivial estimation accuracy as defined in (2) [KMS16]. Therefore we have the following corollary of the first part of Theorem 1.

**Corollary 1.** *Under  $\text{StSSBM}(n, k, a, b)$  with no side information, no  $R$ -local streaming algorithm  $\mathcal{A}$  can achieve non-trivial estimation accuracy. That is,  $\lim_{n \rightarrow \infty} Q_n(\mathcal{A}) = 1/k$ .*

**Remark 1.** *Corollary 1 does not hold if side information is available. As we will see below, an arbitrarily small amount of side information (any  $\alpha < (k - 1)/k$ ) can be boosted to ideal accuracy using  $R$ -local streaming algorithms with sufficiently large  $R$ . On the other hand, for a fixed small  $R$ , a small amount of side information has only limited impact on accuracy. Our numerical simulations illustrate this for voting algorithms, which are  $R$ -local for  $R = 1$ : they do not provide substantial boost over the use of only side information (i.e., the estimated label  $\hat{\tau}(v) = \tilde{\tau}(v)$  at all vertices).*

**Remark 2.** *In practice we can imagine keeping a small memory containing global information and updating it each time a new vertex joins. This global information would be available at each node. For example, we could keep track of the estimated size of each community. For a suitable class of algorithms of this type, it can be shown that they cannot achieve non-trivial reconstruction in the symmetric model  $\text{StSSBM}(n, k, a, b)$  either. Due to space constraints we do not present this result here. Interested readers are referred to Section G in the appendix.*

## 4 Streaming belief propagation

In this section we focus on the symmetric model StSSBM. Notice that this model makes community detection more difficult compared to the asymmetric model, as in the latter case average degrees for vertices are different across communities, and one can obtain non-trivial estimation accuracy by simply using the degree of each vertex. For the symmetric model  $\text{StSSBM}(n, k, a, b, \alpha)$ , we proved that local streaming algorithms cannot provide any non-trivial reconstruction of the true labels if no side information is provided, i.e., if  $\alpha = (k - 1)/k$ . We also comment that, for a fixed (small)  $R$ , accuracy achieved by any algorithm is continuous in  $\alpha$ , and hence a small amount of side information will have a small effect.

In contrast, for any non-vanishing side information  $\alpha < (k - 1)/k$ , we conjecture that information-theoretically optimal reconstruction is possible using a local streaming algorithm, under two conditions: (i) the locality radius  $R$  is large enough; and (ii) the following Kesten-Stigum (KS) condition is met:

$$\lambda \triangleq \frac{(a - b)^2}{a + (k - 1)b} > 1. \quad (4)$$

We will refer to  $\lambda$  as to the ‘signal-to-noise ratio’ (SNR).

We provide evidence towards this conjecture by proposing a streaming belief propagation algorithm (STREAMBP) and showing that it achieves asymptotically the same accuracy as the standard offline BP algorithm. The latter is believed to achieve information-theoretically optimal reconstruction above the KS threshold [DKMZ11, MNS14]. We will describe STREAMBP in the setting of the symmetric model  $\text{StSSBM}(n, k, a, b, \alpha)$ , but its generalization to the asymmetric case is immediate.

The algorithm has a state which is given by a vector of messages indexed by directed edges in  $G(t)$ ,  $\mathbf{m}^t = \{\mathbf{m}_{u \rightarrow v}^t, \mathbf{m}_{v \rightarrow u}^t : (u, v) \in E(t)\}$ . Note that  $G(t)$  is an undirected graph, and each edge  $(u, v)$  corresponds to two messages indexed by  $u \rightarrow v$  and  $v \rightarrow u$ . Each message is a probability distribution over  $[k]$ :

$$\mathbf{m}_{u \rightarrow v} = (m_{u \rightarrow v}(1), m_{u \rightarrow v}(2), \dots, m_{u \rightarrow v}(k)) \in \Delta_k.$$

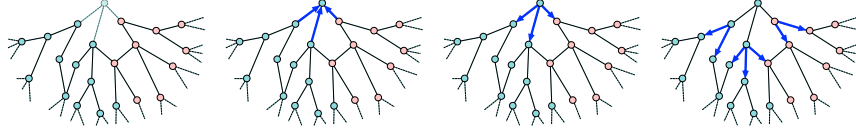


Figure 2: Update schedule of STREAMBP. Upon the arrival of a new vertex (shown in the leftmost figure), STREAMBP performs the belief propagation updates corresponding to the blue edges in the three other figures, in the order from left to right.

The BP update is a map  $\text{BP} : (\Delta_k)^* \times [k] \rightarrow \Delta_k$ , where  $(\Delta_k)^*$  denotes the finite sequences of elements of  $\Delta_k$ :

$$\text{BP}(\{\mathbf{m}_i\}_{i \leq \ell}; \tilde{\tau})(s) := \frac{\text{BP}_0(\tilde{\tau})(s)}{Z} \prod_{i=1}^{\ell} (b + (a - b)m_i(s)). \quad (5)$$

Here  $\text{BP}_0(\tilde{\tau})(s) \triangleq (\alpha + (k - 1 - k\alpha)\mathbb{1}_{\tilde{\tau}=s})/(k - 1)$  and the constant  $Z = Z(\{\mathbf{m}_i : i \leq \ell\}; \tilde{\tau})$  is defined implicitly by the normalization condition  $\sum_{s \in [k]} \text{BP}(\{\mathbf{m}_i : i \leq \ell\}; \tilde{\tau})(s) = 1$ . When a message  $v \rightarrow u$  is updated, we compute its new value by applying the function (5) to the incoming messages into vertex  $v$ , with the exception of  $u \rightarrow v$  (non-backtracking property):

$$\mathbf{m}_{v \rightarrow u} \leftarrow \text{BP}(\{\mathbf{m}_{w \rightarrow v} : w \in \partial v \setminus \{u\}\}; \tilde{\tau}(v)). \quad (6)$$

Here  $\partial v$  denotes the set of neighbors of vertex  $v$  in the current graph. When a new vertex  $v(t)$  joins at time  $t$ , we use the above rule to: (1) update all the messages incoming into  $v(t)$ , i.e.,  $w \rightarrow v(t)$ , for  $w$  a neighbor of  $v(t)$  in  $G(t)$ , and (2) update all messages at distance  $1 \leq \ell \leq R$  from  $v(t)$  in  $G(t)$ , along paths outgoing from  $v(t)$ , in order of increasing distance  $\ell$ . The pseudocode for STREAMBP is given in Algorithm 1, and an illustration in Figure 2. For the sake of simplicity, we analyze this algorithm in the two-group symmetric model  $\text{StSSBM}(n, 2, a, b, \alpha)$ . We believe that the extension of this analysis to other cases is straightforward, but we leave it out of this presentation.

Our main result is that STREAMBP achieves asymptotically at least the same accuracy as offline BP, as originally proposed in [DKMZ11] and analyzed, e.g., in [MX16]. Offline BP performs the updates via Equation (6) in parallel on all the edges of  $G(n)$  for  $R - 1$  iterations, and then computes vertex estimates using  $\mathbf{m}_u \leftarrow \text{BP}(\{\mathbf{m}_{v \rightarrow u}\}_{v \in \partial u}; \tilde{\tau}(u))$ ,  $\hat{\tau}(u) := \arg \max_{s \in [k]} m_u(s)$ . Note that, for each  $R$ , this defines an  $R$ -local algorithm, and hence we will refer to  $R$  as its radius.

**Theorem 2.** *For  $v \in V(n)$ , let  $\mathcal{A}_R(v; G(n), \tilde{\tau}) \in [k]$  be the estimate of  $\tau(v)$  given by Algorithm 1 (STREAMBP), and  $\mathcal{A}_R^{\text{off}}(v; G(n), \tilde{\tau}) \in [k]$  be the estimate given by offline BP with radius  $R$  (equivalently, BP with parallel updates, stopped after  $R$  iterations). Under the model  $\text{StSSBM}(n, 2, a, b, \alpha)$ , STREAMBP performs at least as well as offline BP:*

$$\liminf_{n \rightarrow \infty} (Q_n(\mathcal{A}_R) - Q_n(\mathcal{A}_R^{\text{off}})) \geq 0.$$

---

**Algorithm 1** Streaming  $R$ -local belief propagation

---

```

1: for  $t = 1, 2, \dots, n$  do
2:   // Update the incoming messages:
3:   for  $w \in D_1^t(v(t))$  do
4:      $\mathbf{m}_{w \rightarrow v(t)} \leftarrow \text{BP}(\{\mathbf{m}_{u \rightarrow w} : u \in \partial w \setminus \{v(t)\}\}; \tilde{\tau}(w))$ 
5:   end for
6:   // Update the outgoing messages:
7:   for  $r = 1, 2, \dots, R$  do
8:     for  $v \in D_r^t(v(t))$  do
9:       Let  $v' \in D_1^t(v)$  on a shortest path connecting  $v$  and  $v(t)$ .
10:       $\mathbf{m}_{v' \rightarrow v} \leftarrow \text{BP}(\{\mathbf{m}_{u \rightarrow v'} : u \in \partial v' \setminus \{v\}\}; \tilde{\tau}(v'))$ 
11:    end for
12:  end for
13: end for
14: for  $u \in V$  do
15:    $\mathbf{m}_u \leftarrow \text{BP}(\{\mathbf{m}_{v \rightarrow u}\}_{v \in \partial u}; \tilde{\tau}(u))$ 
16:   Output  $\hat{\tau}(u) := \arg \max_{s \in [k]} m_u(s)$  as an estimate for  $\tau(u)$ .
17: end for

```

---

It is conjectured that, in the presence of side information, i.e., for  $\alpha < (k - 1)/k$ , offline BP is optimal among all polynomial-time algorithms [DKMZ11] (provided  $R$  can be taken arbitrarily



large). Whenever this is the case, the above theorem implies that STREAMBP is optimal as well. In the case of the symmetric model, it is also believed that under the KS condition (Equation 4), and for  $\alpha < (k - 1)/k$ , offline BP does indeed achieve the information-theoretically optimal accuracy. This claim has been proven in certain cases by [MX16]: we can use the results of [MX16] in conjunction with Theorem 2 to obtain conditions under which STREAMBP is information-theoretically optimal.

**Corollary 2.** *Suppose one of the following conditions holds (for a sufficiently large absolute constant  $C$ ) in the two-group symmetric model  $\text{StSSBM}(n, k = 2, a, b, \alpha)$ : (1)  $|a - b| < 2$  and  $\alpha \in (0, 1/2)$ ; (2)  $(a - b)^2 > C(a + b)$  and  $\alpha \in (0, 1/2)$ ; or (3)  $\alpha \in (0, 1/C)$ . Then Algorithm 1 achieves optimal estimation accuracy:*

$$\limsup_{R \rightarrow \infty} \limsup_{n \rightarrow \infty} \left( Q_n(\mathcal{A}_R) - \sup_{\mathcal{A}} Q_n(\mathcal{A}) \right) = 0.$$

(Here the supremum is taken over all algorithms, not necessarily local or online.)

## 5 Empirical evaluation

In this section, we compare the empirical performance of two versions of our streaming belief propagation algorithm (Algorithm 1) with some baselines. We consider the following streaming algorithms:

- STREAMBP: proposed algorithm in this work with radius  $R$ , outlined in Algorithm 1.
- STREAMBP\*: proposed algorithm in this work, which is a ‘bounded-distance’ version of STREAMBP. It is described in more detail in Section 5.1, and its pseudocode is given in Algorithm 2.
- VOTE1X, VOTE2X, VOTE3X: simple plurality voting algorithms that give a weight  $\delta$  to the side information, as defined in Equation (3) (the numbering corresponds to  $\delta \in \{1, 2, 3\}$ ). Despite looking somewhat naïve, voting algorithms are common in industrial applications.

We also compare the streaming algorithms above with ORACLEBP, which is the *offline* belief propagation algorithm, parameterized by its radius (number of parallel iterations)  $R$ . Note that in contrast to the streaming algorithms, to which we reveal one vertex at a time (along with its side information and the edges connecting it to previously revealed vertices), ORACLEBP has access to *the entire graph* and the side information *of all the vertices* from the beginning.

Our experiments are based both on synthetic datasets (Section 5.2) and on real-world datasets (Section 5.3). Because the model considered in this paper assumes *undirected* graphs, in a preprocessing step we convert the input graphs of the real-world datasets into undirected graphs by simply ignoring edge directions. Table 1 shows statistics of the datasets used (after making the graphs undirected); the values used for  $a$  and  $b$  for the real-world datasets are discussed in Section 5.3.

	$ V $	$ E $	$k$	$a$	$b$
Citeseer	3,264	4,536	6	11.47	0.89
Cora	2,708	5,278	7	17.62	0.90
Polblogs	1,490	16,715	2	40.69	4.23
Synthetic	[10,000–50,000]	[20,000–700,000]	[2–5]	[2.5–18]	[0.05–1]

Table 1: Statistics of the synthetic and real-world datasets. For the synthetic datasets, various experiments with a range of parameters are performed.

As the measure of accuracy, we use the empirical fraction of labels correctly recovered by each algorithm, that is

$$\text{Acc} = \hat{\mathbb{E}} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{v \in V(n)} \mathbb{1}(\mathcal{A}(v; G(n), \tilde{\tau}) = \pi \circ \tau(v)) \right].$$

In synthetic data we observe that, as soon as  $\alpha < (k - 1)/k$ , the maximization over  $\pi$  is not necessary (as expected from theory), and hence we drop it.

## 5.1 Bounded-distance streaming BP

In our experimental results presented below, we observed that the simple implementation in Algorithm 1 exhibits undesirable behaviors for certain graphs. We believe this is caused by two factors. First, unlike ORACLEBP, we do not have an upper bound on the radius of influence of each vertex in STREAMBP; it may indeed use long paths. Second, the number of cycles in the graph increases as  $a, b$  grow. Large values of  $a$  and  $b$  can result in many paths being cycles, which negatively affects the performance of the algorithm.

In order to overcome these problems, we use two modifications. The first one is standard: we constrain messages so that  $m_{u \rightarrow v}(s) \in [\varepsilon, 1 - \varepsilon]$  for some fixed small  $\varepsilon > 0$  (we essentially constrain the log-likelihood ratios to be bounded).

The second modification defines a variant, presented in Algorithm 2, which we call STREAMBP\*. Here the estimate at node  $v$  is guaranteed to depend only on the graph structure and side information within  $B_R^n(v)$ , and not on the information outside this ball. This constraint can be implemented in a message-passing fashion, by keeping, on each edge,  $R + 1$  distinct messages  $m_{u \rightarrow v}^0, \dots, m_{u \rightarrow v}^R$ , corresponding to different locality radii.

## 5.2 Synthetic datasets

Figure 3 illustrates the effect of the radius on the performance of the algorithms.<sup>1</sup> We use various settings for  $k, a, b, \alpha$ . We observe that voting algorithms do not perform significantly better than the baseline  $1 - \alpha$  (dashed line). This is due both to the very small radius  $R = 1$  of these algorithms, and to the specific choice of the update rule. For  $R = 1$ , STREAMBP and STREAMBP\* perform significantly better than voting, showing that their update rule is preferable. Their accuracy improves with  $R$ , and is often close to the optimal accuracy (i.e. the accuracy of ORACLEBP for large  $R$ ) already for  $R \approx 5$ .

In Figure 4, we study the effect of the SNR parameter  $\lambda$ , defined in Equation (4), on the performance of the BP algorithms. The accuracy of the algorithms improves as  $\lambda$  increases. It is close to the baseline  $1 - \alpha$  when the SNR is close to the KS threshold at  $\lambda = 1$ , and then it improves for large  $\lambda$ . This is a trace of the phase transition at the KS threshold which is blurred because of side information. For large values of  $R$ , the accuracy of our streaming algorithms STREAMBP and STREAMBP\* nearly matches that of the optimal *offline* algorithm ORACLEBP.

Further experiments on synthetic datasets are reported in Appendix A. We then present in Appendix B the result of experiments where no side information is provided to the algorithms. We observe that in the streaming setting, and for small radius ( $R$ ), neither STREAMBP nor STREAMBP\* achieves high accuracy (above  $1/k$ ) in the absence of side information, as suggested by our theoretical results.

---

### Algorithm 2 STREAMBP\*: Bounded-distance streaming BP

---

```

1: for  $t = 1, 2, \dots, n$  do
2:   for  $v \in D_1^t(v(t))$  do
3:      $m_{v \rightarrow v(t)}^0 \leftarrow 1/k$ 
4:     for  $i = 1, 2, \dots, R$  do
5:        $m_{v \rightarrow v(t)}^i \leftarrow$ 
           BP( $\{m_{v' \rightarrow v}^{i-1}\}_{v' \in \partial v \setminus \{v(t)\}}; \tilde{\tau}(v)$ )
6:     end for
7:   end for
8:   for  $v \in D_1^t(v(t))$  do
9:      $m_{v(t) \rightarrow v}^0 \leftarrow 1/k$ 
10:    for  $i = 1, 2, \dots, R$  do
11:       $m_{v(t) \rightarrow v}^i \leftarrow$ 
           BP( $\{m_{v' \rightarrow v(t)}^{i-1}\}_{v' \in \partial v(t) \setminus \{v\}}; \tilde{\tau}(v(t))$ )
12:    end for
13:  end for
14:  for  $r = 2, 3, \dots, R$  do
15:    for  $v \in D_r^t(v(t))$  do
16:      Let  $v' \in D_1^t(v)$  on a shortest path
           connecting  $v$  and  $v(t)$ .
17:      for  $i = 1, 2, \dots, R$  do
18:         $m_{v' \rightarrow v}^i \leftarrow$ 
           BP( $\{m_{u \rightarrow v'}^{i-1}\}_{u \in \partial v' \setminus \{v\}}; \tilde{\tau}(v')$ )
19:      end for
20:    end for
21:  end for
22: end for
23: // Compute the estimates:
24: for  $u \in V$  do
25:    $m_u \leftarrow$  BP( $\{m_{v \rightarrow u}^R\}_{v \in \partial u}; \tilde{\tau}(u)$ )
26:   Output  $\hat{\tau}(u) := \arg \max_s m_u(s)$ 
27: end for

```

---

<sup>1</sup>Notice that the three voting algorithms do not depend on the radius, resulting in horizontal lines in the diagram.



### 5.3 Real-world datasets

We further investigate the performance of our algorithms on three real-world datasets: Cora [RA15], Citeseer [RA15], and Polblogs [AG05]. Cora and Citeseer are academic citation networks: vertices represent scientific publications partitioned into  $k = 7$  and  $k = 6$  classes respectively; directed edges represent citations of a publication by another. The Polblogs dataset represents the structure of the political blogosphere in the United States around its 2004 presidential election: vertices correspond to political blogs, and directed edges represent the existence of hyperlinks from one blog to another. The blogs are partitioned into  $k = 2$  classes based on their political orientation (liberal or conservative).

As mentioned in the beginning of Section 5, in a preprocessing step we convert the input graphs of the real-world datasets into undirected graphs by simply ignoring edge directions. Also, since the graphs do not stem from the models of Section 2, we use oracle estimates of parameters  $a$  and  $b$ , obtained by matching the density of intra- and inter-community edges with those in the model  $\text{StSSBM}(n, k, \tilde{a}, \tilde{b}, \alpha)$ . Namely, for a graph  $G = (V, E)$ , letting  $V_1, \dots, V_k \subseteq V$  be the ground-truth communities, we set

$$\tilde{a} = |V| \cdot \frac{\sum_{i \in [k]} \sum_{(u,v) \in E} \mathbb{1}\{u, v \in V_i\}}{\sum_{i \in [k]} \binom{|V_i|}{2}}, \quad \tilde{b} = |V| \cdot \frac{\sum_{\{i,j\} \in \binom{[k]}{2}} \sum_{(u,v) \in E} \mathbb{1}\{u \in V_i, v \in V_j\}}{\sum_{\{i,j\} \in \binom{[k]}{2}} |V_i| |V_j|}.$$

For each dataset, we run the streaming algorithms STREAMBP and STREAMBP\* using the parameters  $a = \tilde{a}$  and  $b = \tilde{b}$ . Notice that these estimates cannot be implemented in practice because we do not know the communities  $V_j$  to start with. However, the performances appear not to be too sensitive to these estimates; we provide empirical evidence of this in Appendix A.

Figure 5 shows the accuracy of different algorithms on these datasets, for selected values of  $\alpha$ . Although the graphs in these datasets are not random graphs generated according to StSBM, the empirical results generally align with the theoretical results we proved for that model. Specifically, we see that our streaming algorithm STREAMBP\* is approximately as accurate as the *offline* algorithm ORACLEBP, and significantly better than the voting algorithms. STREAMBP produces high-quality results for Cora and Citeseer datasets, but behaves erratically on the Polblogs dataset. As Polblogs is relatively dense compared with Cora and Citeseer thus contains more cycles, such phenomenon is likely due to the issues discussed in Section 5.1.

Appendix C provides additional details on these experiments, as well as results for other values of  $\alpha$ .

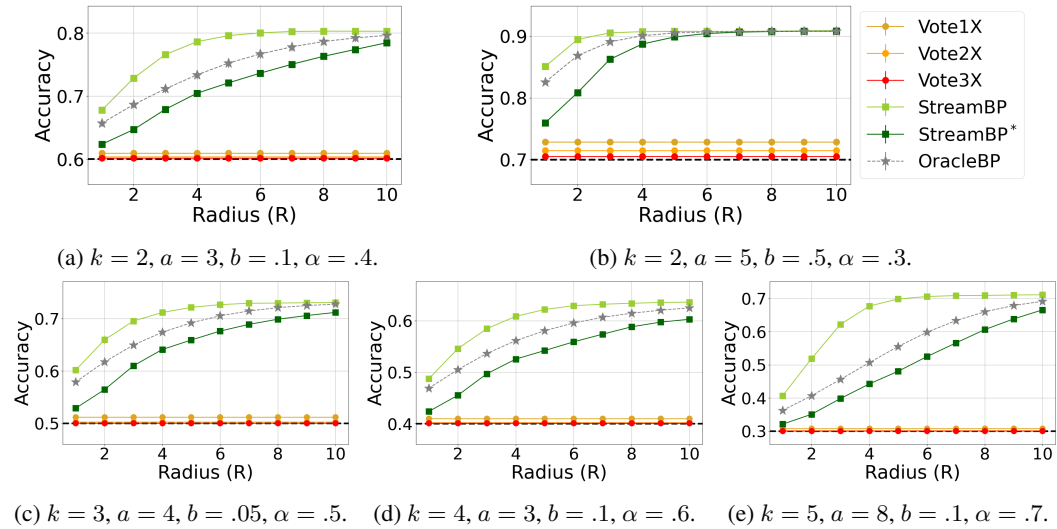


Figure 3: Results of the experiments on synthetic datasets with 50,000 vertices. The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely  $1 - \alpha$ .

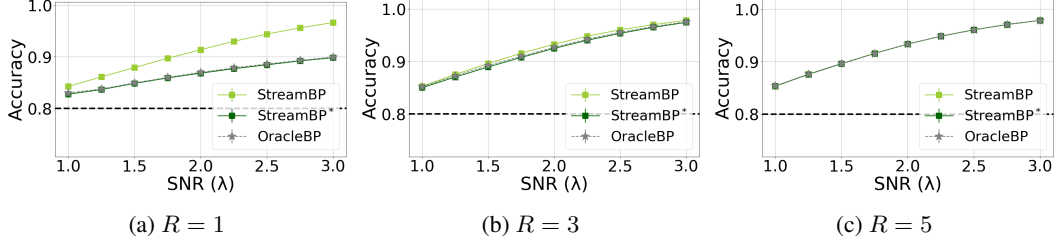


Figure 4: Results of the experiments on synthetic datasets with 50,000 vertices, with  $k = 2$ ,  $a + b = 8$ ,  $\alpha = 0.2$ , as we increase the ratio  $\lambda$  from 1.0 to 3.0. The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely  $1 - \alpha$ .

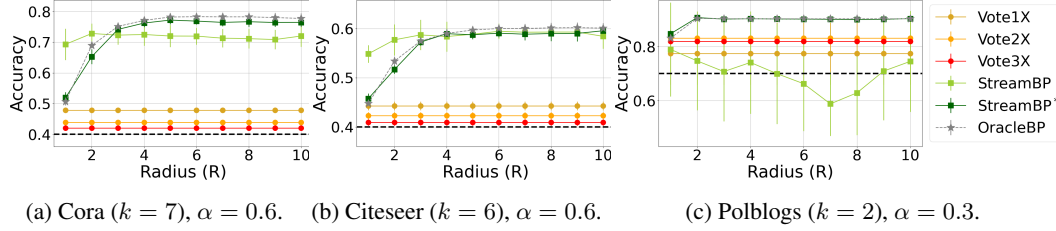


Figure 5: Results of experiments on real-world datasets. The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely  $1 - \alpha$ .

## Acknowledgments and Disclosure of Funding

A.M. and Y.W. were partially supported by NSF grants CCF-2006489, IIS-1741162 and the ONR grant N00014-18-1-2729.

J.T. has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No 759471).

## References

- [Abb17] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.
- [ABFX08] Edoardo Maria Airoidi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of machine learning research*, 2008.
- [ABH15] Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, 2015.
- [AG05] Lada A. Adamic and Natalie S. Glance. The political blogosphere and the 2004 u.s. election: divided they blog. In *LinkKDD*, pages 36–43, 2005.
- [AS15] Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 670–688. IEEE, 2015.
- [CNM04] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.
- [CSG16] Mário Cordeiro, Rui Portocarrero Sarmento, and Joao Gama. Dynamic community detection in evolving networks using locality modularity optimization. *Social Network Analysis and Mining*, 6(1):15, 2016.
- [DKMZ11] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.
- [FM17] Zhou Fan and Andrea Montanari. How well do local algorithms solve semidefinite programs? In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 604–614, 2017.
- [For10] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.
- [GA05] Roger Guimera and Luis A Nunes Amaral. Functional cartography of complex metabolic networks. *nature*, 433(7028):895–900, 2005.
- [GLZ08] Andrew B. Goldberg, Ming Li, and Xiaojin Zhu. Online manifold regularization: A new learning setting and empirical study. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD*, volume 5211 of *Lecture Notes in Computer Science*, pages 393–407. Springer, 2008.
- [GN02] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.
- [GS14] David Gamarnik and Madhu Sudan. Limits of local algorithms over sparse random graphs. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 369–376, 2014.
- [GT12] Shayan Oveis Gharan and Luca Trevisan. Approximating the expansion profile and almost optimal local graph clustering. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 187–196. IEEE, 2012.
- [HLL83] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.
- [HLS14] Hamed Hatami, László Lovász, and Balázs Szegedy. Limits of locally–globally convergent graph sequences. *Geometric and Functional Analysis*, 24(1):269–296, 2014.
- [HS12] Petter Holme and Jari Saramäki. Temporal networks. *Physics reports*, 519(3):97–125, 2012.

- [HS17] Samuel B Hopkins and David Steurer. Efficient bayesian estimation from few samples: community detection and related problems. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 379–390. IEEE, 2017.
- [JTZ04] Daxin Jiang, Chun Tang, and Aidong Zhang. Cluster analysis for gene expression data: a survey. *IEEE Transactions on knowledge and data engineering*, 16(11):1370–1386, 2004.
- [JYL<sup>+</sup>18] Muhammad Aqib Javed, Muhammad Shahzad Younis, Siddique Latif, Junaid Qadir, and Adeel Baig. Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*, 108:87–111, 2018.
- [KMS16] Varun Kanade, Elchanan Mossel, and Tselil Schramm. Global and local information in clustering labeled block models. *IEEE Transactions on Information Theory*, 62(10):5906–5917, 2016.
- [KN11] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.
- [KvB<sup>+</sup>14] Georg Kreml, Indre Žliobaite, Dariusz Brzeziński, Eyke Hüllermeier, Mark Last, Vincent Lemaire, Tino Noack, Ammar Shaker, Sonja Sievi, Myra Spiliopoulou, and Jerzy Stefanowski. Open challenges for data stream mining research. *SIGKDD Explor. Newsl.*, 16(1):1–10, September 2014.
- [LF09] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.
- [LP17] Russell Lyons and Yuval Peres. *Probability on trees and networks*, volume 42. Cambridge University Press, 2017.
- [Mas14] Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 694–703, 2014.
- [MKV<sup>+</sup>20] L-E Martinet, MA Kramer, W Viles, LN Perkins, E Spencer, CJ Chu, SS Cash, and ED Kolaczyk. Robust dynamic community detection with applications to human brain functional networks. *Nature communications*, 11(1):1–13, 2020.
- [MNS14] Elchanan Mossel, Joe Neeman, and Allan Sly. Belief propagation, robust reconstruction and optimal recovery of block models. In *Conference on Learning Theory*, pages 356–370, 2014.
- [MNS15] Elchanan Mossel, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3-4):431–461, 2015.
- [MNS18] Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. *Combinatorica*, 38(3):665–708, 2018.
- [MO04] Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM transactions on computational biology and bioinformatics*, 1(1):24–45, 2004.
- [Mon15] Andrea Montanari. Finding one community in a sparse graph. *Journal of Statistical Physics*, 161(2):273–299, 2015.
- [MX16] Elchanan Mossel and Jiaming Xu. Local algorithms for block models with side information. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 71–80, 2016.
- [PKVS12] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554, 2012.

- [RA15] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.
- [RCC<sup>+</sup>04] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the national academy of sciences*, 101(9):2658–2663, 2004.
- [RCY<sup>+</sup>11] Karl Rohe, Sourav Chatterjee, Bin Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *Annals of Statistics*, 39(4):1878–1915, 2011.
- [Suo13] Jukka Suomela. Survey of local algorithms. *ACM Computing Surveys (CSUR)*, 45(2):1–40, 2013.
- [Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.
- [VLBB08] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.
- [WGKM18] Tal Wagner, Sudipto Guha, Shiva Prasad Kasiviswanathan, and Nina Mishra. Semi-supervised learning on data streams via temporal label propagation. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 5082–5091. PMLR, 2018.
- [WZCX18] Feifan Wang, Baihai Zhang, Senchun Chai, and Yuanqing Xia. An extreme learning machine-based community detection algorithm in complex networks. *Complexity*, 2018, 2018.
- [ZGL03] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In Tom Fawcett and Nina Mishra, editors, *Machine Learning, Proceedings of the Twentieth International Conference ICML*, pages 912–919. AAAI Press, 2003.
- [ZWCY19] Xiangxiang Zeng, Wen Wang, Cong Chen, and Gary G Yen. A consensus community-based particle swarm optimization for dynamic community detection. *IEEE transactions on cybernetics*, 50(6):2502–2513, 2019.

## A Further experiments with synthetic datasets

**Results with different sets of parameters.** In Figures 6–9, we report more comprehensively the dependence of our algorithms’ performance on the radius, where each figure corresponds to one choice of  $k = 2, 3, 4, 5$ . Similarly to Figure 3, we can see that as the radius increases, the performance of the three belief propagation algorithms (STREAMBP, STREAMBP\*, and ORACLEBP) improves and converges to the same value. For some parameter settings, we observe that STREAMBP performs poorly when the radius is above a certain threshold; that is likely due to the issues discussed in Section 5.1.

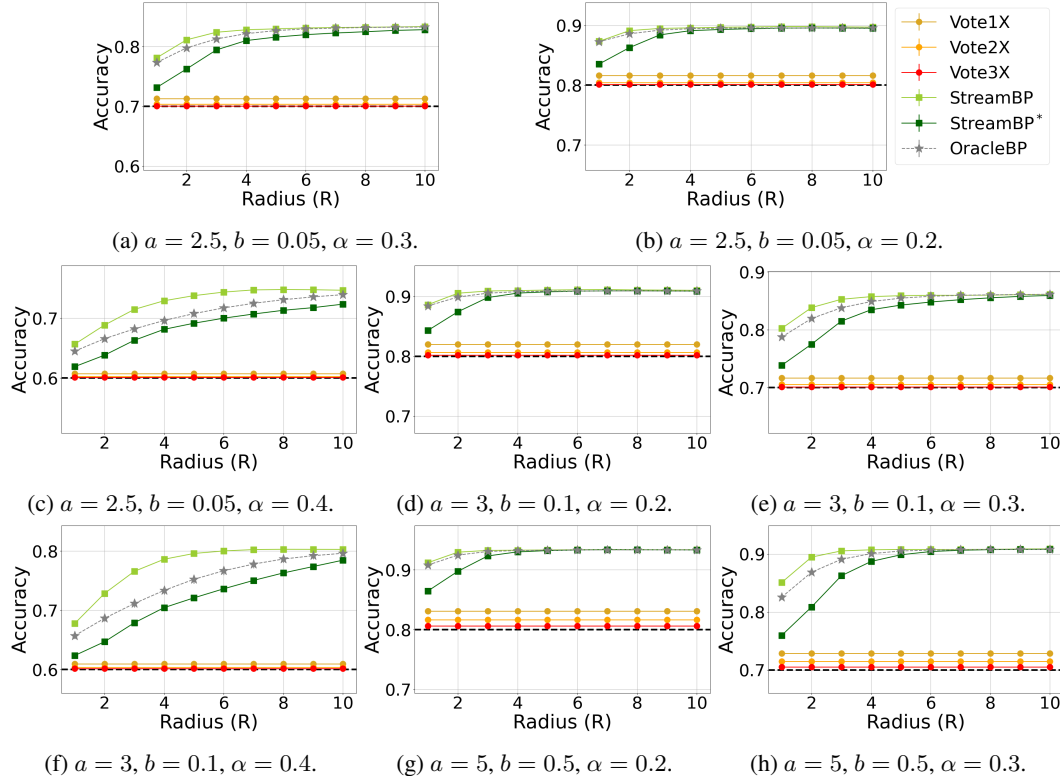


Figure 6: Results of the experiments on synthetic datasets with 50,000 vertices, and  $k = 2$ . The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely  $1 - \alpha$ .

**Variation of the accuracy of STREAMBP and STREAMBP\* during their execution.** We also investigate how the accuracy of the streaming algorithms STREAMBP and STREAMBP\* varies during their execution on a given graph, as new vertices arrive one at a time. Figure 10 shows the results obtained for graphs generated according to the distribution  $\text{StSSBM}(n, a, b, k, \alpha)$ , for various parameter settings.

We generally observe a very high accuracy (close to 1.0) for the first very few vertices, and then a sharp decline. This is not surprising given our use of the estimation accuracy  $Q_n(\mathcal{A})$  defined in Section 2; in particular, the accuracy upon arrival of the first vertex is always equal to 1.0. After this initial sharp decline, the arrival of more vertices almost steadily improves the accuracy, which eventually stabilizes. Note that this improvement accelerates at some point; this is a trace of the phase transition at the KS threshold which is blurred because of side information. Also note that the truncation of a triple  $(\tau, \tilde{\tau}, \mathbf{G}) \sim \text{StSSBM}(n, k, a, b, \alpha)$  to the first  $n' < n$  vertices induces the distribution  $\text{StSSBM}(n', k, \frac{n'}{n}a, \frac{n'}{n}b, \alpha)$ , whose signal-to-noise ratio is  $\lambda(n') \triangleq \frac{n'}{n}\lambda$ ; i.e., the signal-to-noise ratio increases linearly with  $n'$ . Each plot in Figure 10 has a vertical line showing the point where  $\lambda(n')$  crosses the threshold 1.0.



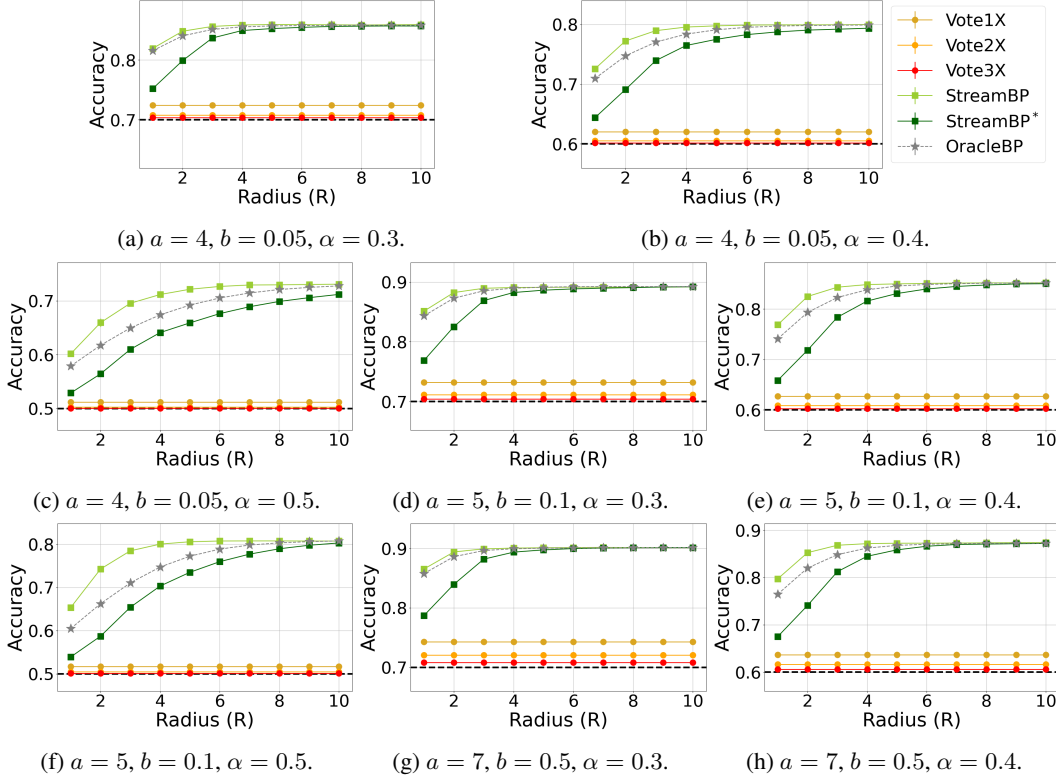


Figure 7: Results of the experiments on synthetic datasets with 50,000 vertices, and  $k = 3$ . The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely  $1 - \alpha$ .

**Robustness of STREAMBP and STREAMBP\* w.r.t. the parameters  $a$  and  $b$ .** Note that our proposed algorithms use  $a$  and  $b$  as input parameters. When applying these algorithms to real-world datasets that do not conform to the StSSBM model, we must approximate these parameters. Indeed in Section 5 we used empirical estimates of  $a$  and  $b$ .

In this section we provide some evidence that our algorithms’ behavior is robust to the choice of  $a$  and  $b$ . In Figures 11 and 12 we present some results of experiments on synthetic data, where the algorithm and the model generating the input graph are given different  $a$  or  $b$  parameters. We observe that even with a relatively high discrepancy between the approximate and true parameters, STREAMBP\* still achieves results comparable to the optimal setting of the parameters. Similar observations can be made about STREAMBP.

Various settings of  $k$ ,  $a$ ,  $b$ , and  $\alpha$  are given as the caption to the individual plots. Here  $a$  and  $b$  signify the true parameters, according to which the input graph was generated. The performance of STREAMBP\* is then plotted with various input parameters. For instance, the label “a+200%” in Figure 11 indicates that the algorithm receives a parameter  $a$  that is three times greater than the true value. Similarly “b-67%” in Figure 12 indicates that the algorithm receives a parameter  $b$  that is 67% less than the true value.

## B Experiments without side information

In this section, we provide details of our experiments on synthetic data, when no side information is provided to the algorithms. Here  $\alpha$  is set to  $1 - 1/k$ , that is,  $\tilde{\tau}$  is completely independent from  $\tau$ . Note that in this setting we cannot hope to have large overlap between the output labels  $\hat{\tau}$  and the true labels  $\tau$ . Therefore, we must evaluate the algorithms based on the best possible permutation of the output labels, as described in Section 2.1 in the definition of  $Q_n(\mathcal{A})$ :

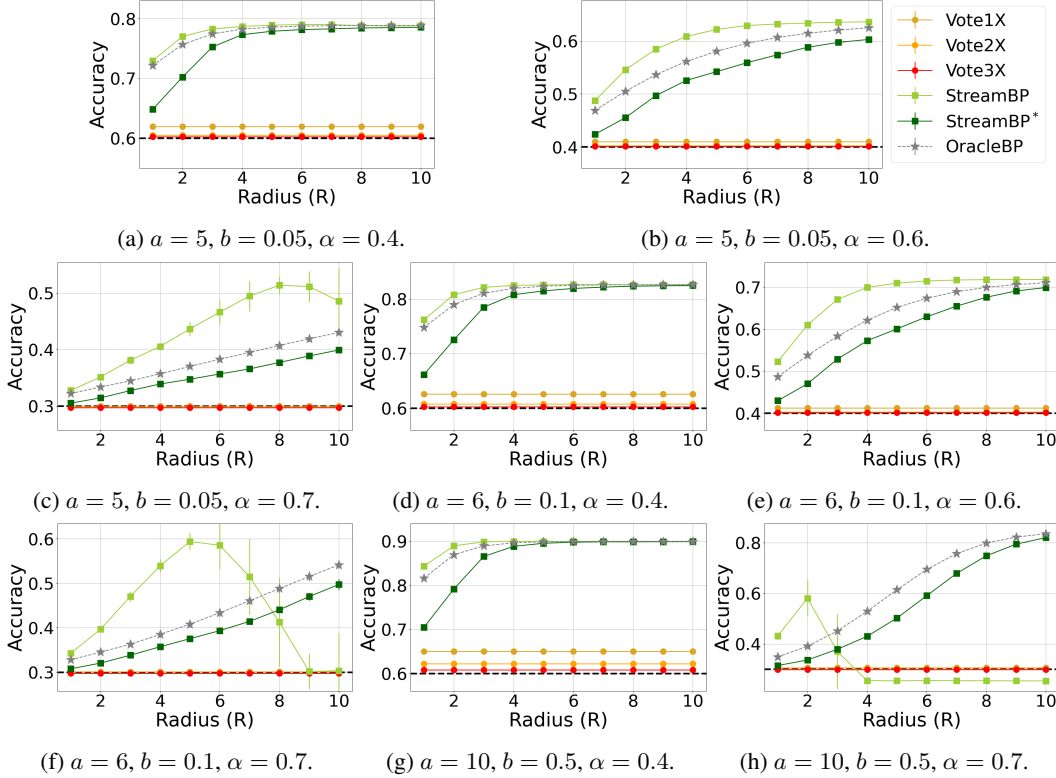


Figure 8: Results of the experiments on synthetic datasets with 50,000 vertices, and  $k = 4$ . The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely  $1 - \alpha$ .

$$\max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{v \in V(n)} \mathbb{1}(\hat{\tau}(v) = \pi \circ \tau(v)).$$

Our observations confirm the theoretical result from Corollary 1. As predicted, when  $r$  is small, no algorithm—neither the baselines nor our algorithms—can achieve any meaningful improvement over the trivial  $1/k$  accuracy. We further observe that for sufficiently large  $r$ , both the offline algorithm ORACLEBP and our algorithm STREAMBP\* perform significantly better than  $1/k$ . However, this occurs when  $r$  is comparable to the diameter of the graph, when an online algorithm is no longer efficient. STREAMBP is unable to get any non-trivial result due to issues described in Section 5.1 which become especially problematic for large values of  $a$ ,  $b$ , and  $r$ .

We present the results of our experiments for various settings of  $k$ ,  $a$ , and  $b$  in Figure 13. Note that the settings always satisfy the Kesten-Stigum condition from Equation (4). Without side information, this is necessary to see any non-trivial behavior, even for large radii.

## C Additional results on real-world datasets

Figure 14 summarizes the results obtained by running the different algorithms on the datasets Cora, Citeseer, and Polblogs. For each dataset and choice of the radius (used by the algorithms STREAMBP, STREAMBP\*, and ORACLEBP) and of the parameter  $\alpha$ , we run each algorithm 9 times; each run independently chooses an arrival order of the vertices (uniformly at random among all permutations of the vertices) and the side information (as described in Section 2). We note that the accuracy of our streaming algorithm STREAMBP\* is comparable to that of the *offline* algorithm ORACLEBP, and significantly superior to the accuracy of the voting algorithms. STREAMBP produces high-quality results for the datasets Cora and Citeseer, but behaves erratically in the Polblogs dataset, generally worse than the voting algorithms; that is likely due to the issues discussed in Section 5.1.

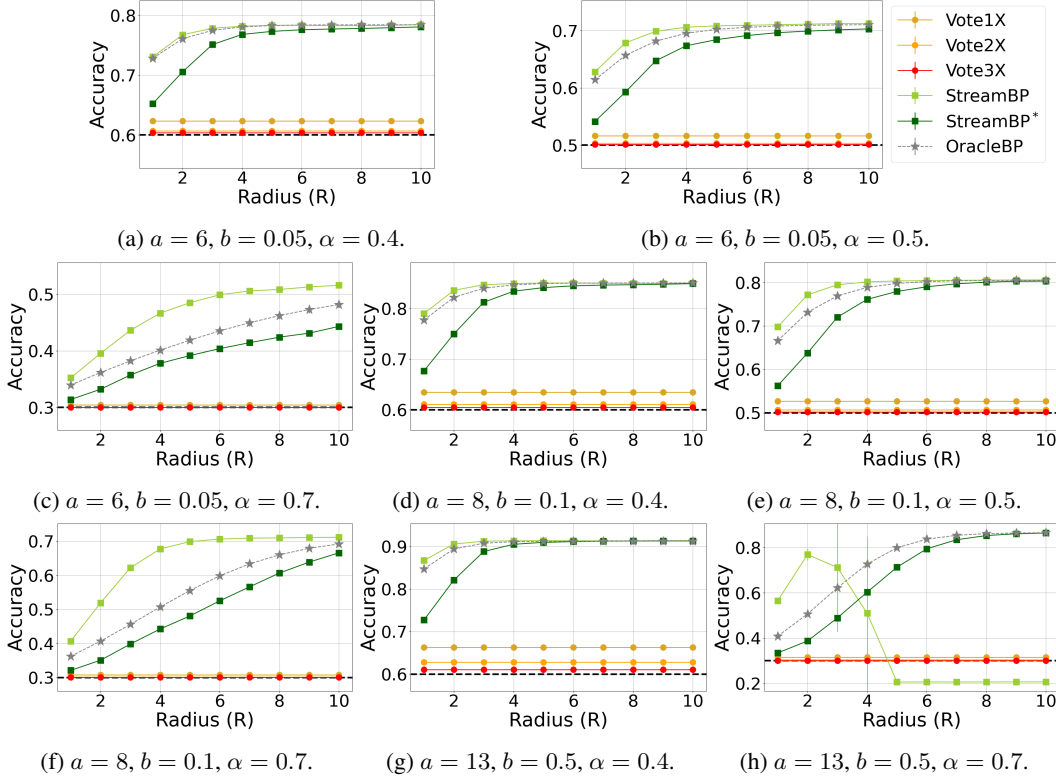


Figure 9: Results of the experiments on synthetic datasets with 50,000 vertices, and  $k = 5$ . The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely  $1 - \alpha$ .

## D Definitions and technical lemmas

For completeness, we reproduce a standard lemma establishing that sparse graphs from SBM are locally tree-like.

**Lemma 1** ([MNS15]). *Let  $(X, G) \sim \text{SSBM}(n, 2, a/n, b/n)$  and  $R = R(n) = \lfloor \frac{1}{10} \log(n) / \log(2(a+b)) \rfloor$ . Let  $B_R := \{v \in [n] : d_G(1, v) \leq R\}$  be the set of vertices at graph distance at most  $R$  from vertex 1,  $G_R$  be the restriction of  $G$  on  $B_R$ , and let  $X_R = \{X_u : u \in B_R\}$ . Let  $T_R$  be a Galton-Watson tree with offspring  $\text{Poisson}(a+b)/2$  and  $R$  generations, and let  $\tilde{X}^{(t)}$  be the labelling on the vertices at generation  $t$  obtained by broadcasting the bit  $\tilde{X}^{(0)} := X_1$  from the root with flip probability  $b/(a+b)$ . Let  $\tilde{X}_R = \{\tilde{X}_u^{(t)} : t \leq R\}$ . Then, there exists a coupling between  $(G_R, X_R)$  and  $(T_R, \tilde{X}_R)$  such that*

$$\lim_{n \rightarrow \infty} \mathbb{P} \left( (G_R, X_R) = (T_R, \tilde{X}_R) \right) = 1.$$

For  $A \subseteq V$ , let  $\tau(A)$  be the vector containing all true labels of vertices in  $A$ , and  $\tilde{\tau}(A)$  be the vector containing all noisy labels of vertices in  $A$ .

**Definition 3** (Labeled branching tree). *For  $r \in \mathbb{N}^+$ ,  $d > 0$ ,  $p \in (0, 1)$ , let  $\mathbb{P}_{r,d,p}^T$  denote the law of a labeled Galton Watson branching tree: this tree has  $r$  generations, with the offspring distribution being Poisson with expectation  $d$ . Each vertex in the tree is associated with a label taking values in  $\{1, 2\}$ . The label at the root node is uniformly distributed over  $\{1, 2\}$ , and labels on the rest of the vertices are obtained by broadcasting the label from the root node with flip probability  $p$ .*

Furthermore, we denote the optimal estimation accuracy by

$$Q_n^* \triangleq \sup_A \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{v \in V(n)} \mathbf{1}(\mathcal{A}(v; G(n), \tilde{\tau}) = \pi \circ \tau(v)) \right]. \quad (7)$$

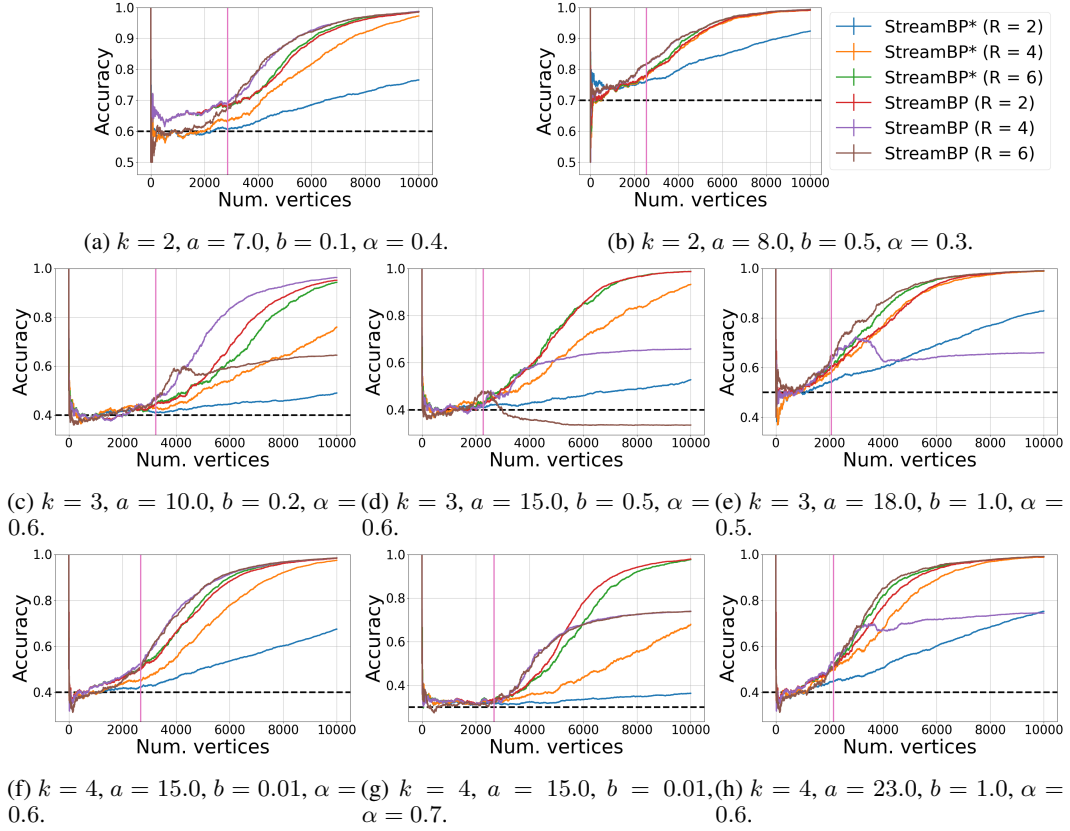


Figure 10: Accuracy per iteration obtained by running STREAMBP and STREAMBP\* on a graph sampled from  $\text{StSSBM}(n, k, a, b, \alpha)$  with  $n = 10,000$ . The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely  $1 - \alpha$ . The pink vertical line indicates the point where the signal-to-noise ratio crosses the threshold 1.0.

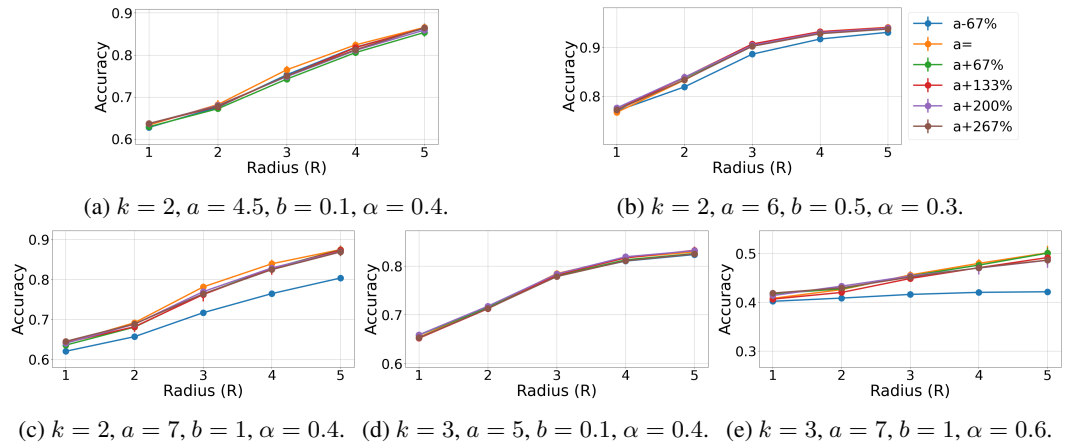


Figure 11: Robustness of STREAMBP\* w.r.t. to the choice of  $a$ .

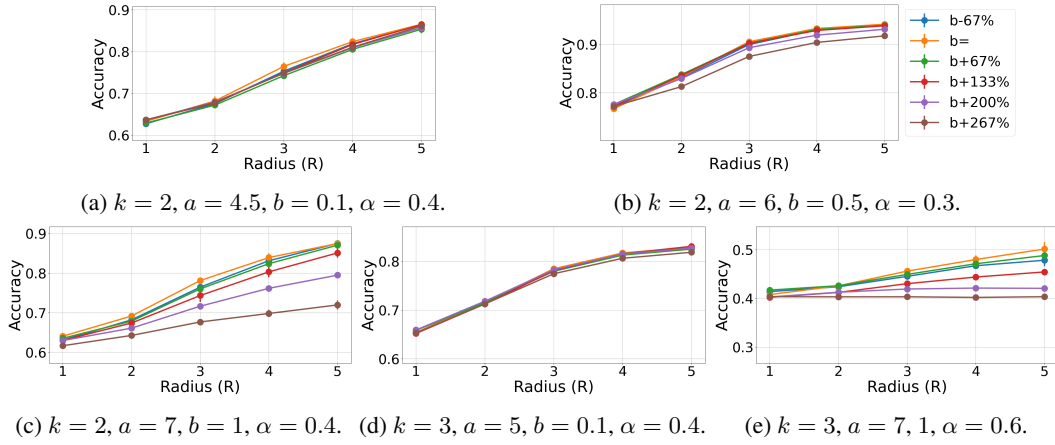


Figure 12: Robustness of STREAMBP\* w.r.t. to the choice of  $b$ .

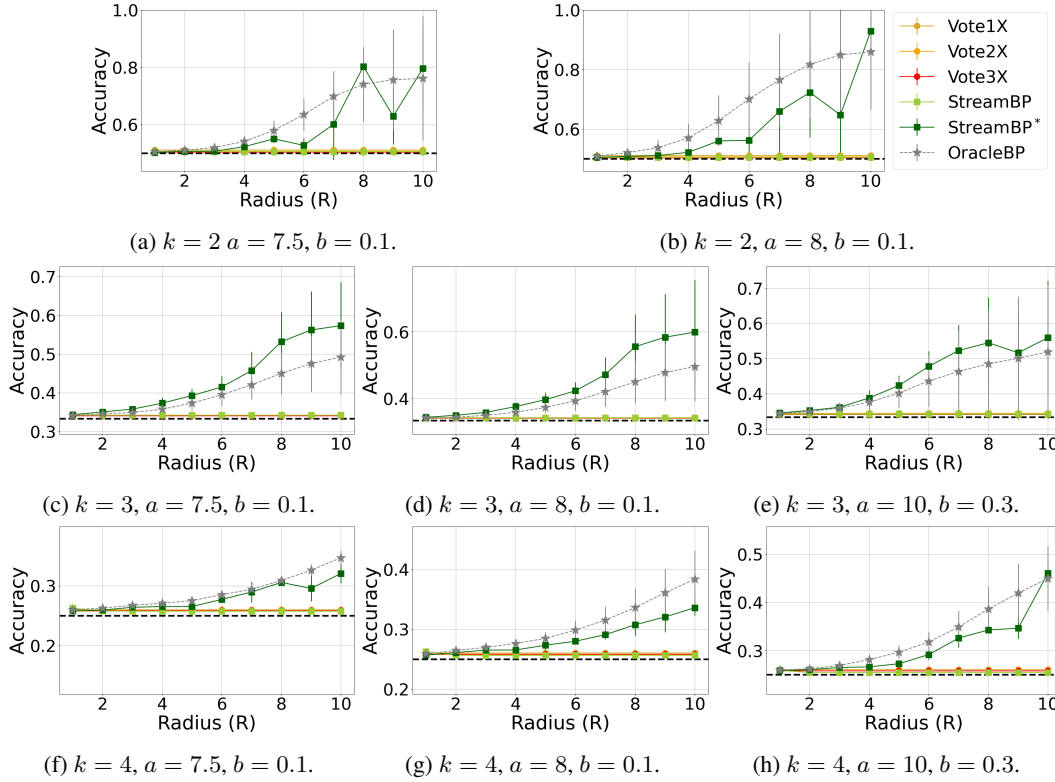


Figure 13: Results of the experiments on synthetic datasets with 10,000 vertices, and no side information (i.e.  $\alpha = 1 - 1/k$ ).

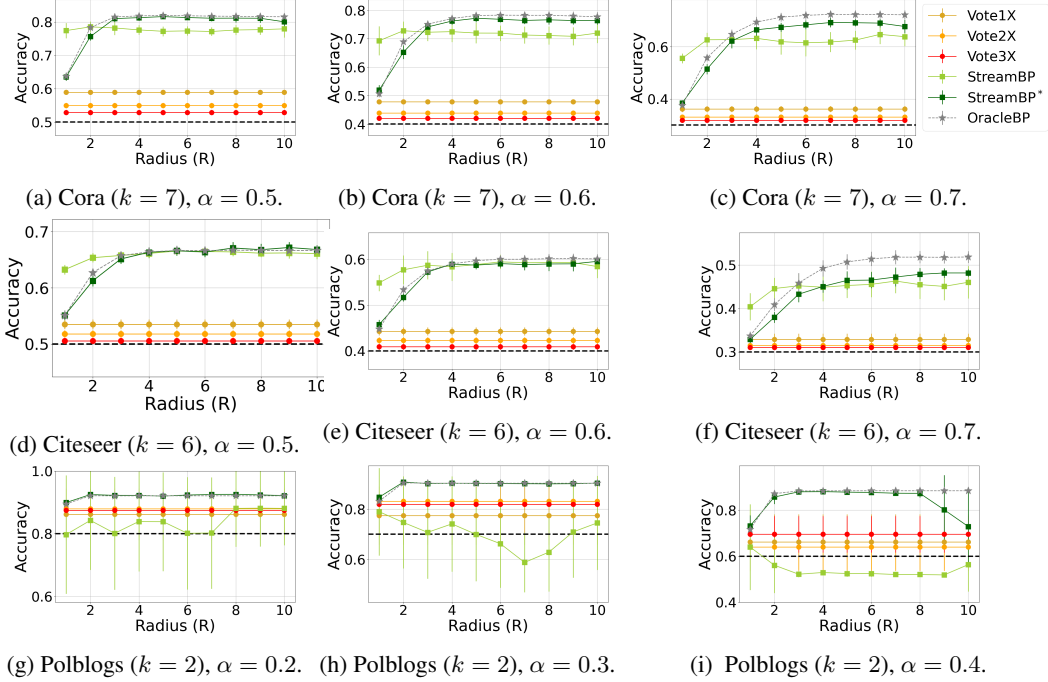


Figure 14: Results of the experiments on real-world datasets. The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely  $1 - \alpha$ .

## E Analysis of local streaming algorithms

### E.1 Proof of Theorem 1

With a slight abuse of notations, in this part of the proof, when we refer to  $V_r^t(v)$ ,  $E_r^t(v)$ ,  $B_r^t(v)$  and  $\mathcal{G}_r^t(v)$ , we do not assume we know the revealing orderings of vertices within. For the sake of simplicity, we consider  $k = 2$ , cases with  $k > 2$  can be proven similarly. Let  $d = (a + b)/2$  be the average degree,  $r \in \mathbb{N}^+$  satisfying

$$r \geq 2^{2R+3} R \left( 1 + (1 - e^{-d})^{-1} \sum_{i=0}^R d^i \right) d^{2R} e.$$

Let  $(\lambda_{r+R}, T_{r+R}) \sim \mathbb{P}_{r+R, d, \frac{b}{a+b}}^T$  as in Definition 3, with  $T_{r+R}$  being the graph and  $\lambda_{r+R}$  being the set of labels. By Lemma 1, for any  $\epsilon > 0$ , there exists  $n_\epsilon \in \mathbb{N}^+$  which is a function of  $r, R$  and  $\epsilon$ , such that for  $n \geq n_\epsilon$ , there exists a coupling of  $(\lambda_{r+R}, T_{r+R})$  and  $(\tau(V_{r+R}^n(v_0)), B_{r+R}^n(v_0))$  preserving  $u_0$  paired up with  $v_0$ , and satisfies

$$\mathbb{P}((\lambda_{r+R}, T_{r+R}) \neq (\tau(V_{r+R}^n(v_0)), B_{r+R}^n(v_0))) \leq \epsilon/2.$$

If  $\mathcal{G}_{v_0}^n$  is not a subgraph of  $B_r^n(v_0)$ , then there must exist  $v_b \in D_r^n(v_0)$  such that  $v_b$  belongs to  $\mathcal{G}_{v_0}^n$ . This is equivalently saying that there exists an ‘‘information flow’’ starting at  $v_b$ , proceeds as vertices are gradually revealed, and finally could reach  $v_0$  by the end.

**Definition 4** (Information flow). *Given the graph  $G(n) = (V(n), E(n))$ , an information flow with origin at  $v_b$  and end at  $v_0$  is defined as a sequence of vertices  $p_1, p_2, \dots, p_l \in V(n)$ , such that*

1.  $t^*(p_i) = t_i$ , and  $t_i < t_{i+1}$  for all  $i \in [l - 1]$ .
2.  $V_R^n(p_i) \cap V_R^n(p_{i+1}) \neq \emptyset$ . Furthermore,  $\min\{d(v_0, v) : v \in V_R^n(p_{i+1})\} < \min\{d(v_0, v) : v \in V_R^n(p_i)\}$ , for all  $i \in [l - 1]$ .
3.  $v_b \in V_R^n(p_1)$ ,  $v_0 \in V_R^n(p_l)$ ,  $v_0 \notin V_R^n(p_i)$  for all  $i \in [l - 1]$ .



Notice that on the event  $\mathcal{B}_{r+R}^n(v_0)$  is a tree, a necessary condition for  $v_b \in \mathcal{G}_{v_0}^n$  is that there exists an information flow  $u_1, u_2, \dots, u_l$  with origin at  $v_b$  and end at  $v_0$ . For  $i \in [l]$ , among all vertices on the shortest path connecting  $v_0$  and  $v_b$ , let  $v_i$  be a vertex with the smallest graph distance to  $u_i$ . A moment of thought reveals that we can find an eligible information flow such that  $v_1, \dots, v_l$  are distinct.

Denote the set of vertices on the path connecting  $v_0$  and  $v_b$  by  $V_p$ . Given  $l$  and the graph, the number of (unordered) vertex combinations  $\{v_1, v_2, \dots, v_l\}$  is upper bounded by  $\binom{r}{l}$ . For each  $x \in \{v_1, v_2, \dots, v_l\}$ , we define the following set:

$$V_p^x := \{v \in V_R^n(x) : \nexists v' \in V_p, v' \neq x, d(v, v') < d(v, x)\}.$$

Given  $\{v_1, v_2, \dots, v_l\}$ , the total number of possible unordered vertex combinations  $\{u_1, u_2, \dots, u_l\}$  is upper bounded by  $\prod_{i=1}^l |V_p^{v_i}|$ . Furthermore, given unordered set  $\{u_1, u_2, \dots, u_l\}$ , by definition one can specify their relative ordering  $u_1, u_2, \dots, u_l$  by sorting the following distances:  $\min\{d(v_0, v) : v \in V_R^n(u_i)\}$ . Finally, for such combinations to exist, one must have  $\lfloor r/2R \rfloor \leq l \leq r$ .

As a result, given  $\mathcal{B}_{r+R}^n(v_0)$ , if it is a tree, then the conditional probability of  $\mathcal{G}_{v_0}^n$  not being a subgraph of  $\mathcal{B}_r^n(v_0)$  is upper bounded by

$$\sum_{v_b \in V(n)} \mathbb{1}\{v_b \in D_r^n(v_0)\} \sum_{l=\lfloor r/2R \rfloor}^r \sum_{\{v_1, v_2, \dots, v_l\}} \prod_{i=1}^l |V_p^{v_i}| \times \frac{1}{l!}.$$

Let  $\mathbb{P}_{bt}(\cdot)$  denote the probability distribution over  $(\lambda_{r+R}, T_{r+R})$ , and let  $\mathbb{E}_{bt}(\cdot)$  be the corresponding notation for taking expectation under that distribution. Then we have

$$\begin{aligned} & \mathbb{P}(\mathcal{G}_{v_0}^n \text{ is not a subgraph of } \mathcal{B}_r^n(v_0)) \\ & \leq \mathbb{P}((\lambda_{r+R}, T_{r+R}) \neq (\tau(V_{r+R}^n(v_0)), \mathcal{B}_{r+R}^n(v_0))) + \\ & \quad \mathbb{P}(\mathcal{G}_{v_0}^n \text{ is not a subgraph of } \mathcal{B}_r^n(v_0), (\lambda_{r+R}, T_{r+R}) = (\tau(V_{r+R}^n(v_0)), \mathcal{B}_{r+R}^n(v_0))) \\ & \leq \epsilon/2 + \sum_{v_b \in V(n)} \mathbb{E}_{bt} \left[ \mathbb{1}\{v_b \in D_r^n(v_0)\} \sum_{l=\lfloor r/2R \rfloor}^r \sum_{\{v_1, v_2, \dots, v_l\}} \prod_{i=1}^l |V_p^{v_i}| \times \frac{1}{l!} \right] \\ & \stackrel{(i)}{\leq} \epsilon/2 + \sum_{v_b \in V(n)} \mathbb{E}_{bt} \left[ \mathbb{1}\{v_b \in D_r^n(v_0)\} \sum_{l=\lfloor r/2R \rfloor}^r \sum_{\{v_1, v_2, \dots, v_l\}} \frac{1}{l!} \times \underbrace{\left(1 + (1 - e^{-d})^{-1} \sum_{i=1}^R d^i\right)^l}_{C(d, R)^l} \right] \\ & \leq \epsilon/2 + \sum_{l=\lfloor r/2R \rfloor}^r \binom{r}{l} \frac{1}{l!} C(d, R)^l d^r, \end{aligned} \tag{8}$$

where (i) uses the fact that conditioning on  $v_b \in D_r^n(v_0)$ , the path connecting  $v_0$  and  $v_b$ , and the choice of  $v_1, \dots, v_l$ , the conditional expectation of  $\prod_{i=1}^l |V_p^{v_i}|$  is upper bounded by  $C(d, R)^l$ . By Stirling's formula, there exists numerical constants  $C_1$  and  $C_2$ , such that for any positive integer  $m$ ,

$$C_2 \sqrt{m} \left(\frac{m}{e}\right)^m \leq m! \leq C_1 \sqrt{m} \left(\frac{m}{e}\right)^m.$$

Then application of Stirling's formula gives us

$$\begin{aligned} & \sum_{l=\lfloor r/2R \rfloor}^r \binom{r}{l} \frac{1}{l!} C(d, R)^l d^r \\ & \leq \sum_{l=\lfloor r/2R \rfloor}^r \frac{2^r d^r}{C_2} \left(\frac{e}{l}\right)^l C(d, R)^l \\ & \leq \frac{2^{2R} d^{2R}}{C_2} \sum_{l=\lfloor r/2R \rfloor}^{\infty} \left(\frac{2^{2R+2} R C(d, R) d^{2R} e}{r}\right)^l. \end{aligned} \tag{9}$$

where we use the fact that  $r/2R - 1 \leq l \leq r/2R$  and  $2R/(r - 2R) \leq 4R/r$  for all  $r \geq 2^{2R+3}RC(d, R)d^{2R}e$ . Furthermore, with  $r$  having value exceeding this threshold,  $\left(\frac{2^{2R+2}RC(d, R)d^{2R}e}{r}\right) \leq \frac{1}{2}$ , thus

$$\begin{aligned} \frac{2^{2R}d^{2R}}{C_2} \sum_{l=\lfloor r/2R \rfloor}^{\infty} \left(\frac{2^{2R+2}RC(d, R)d^{2R}e}{r}\right)^l &\leq \frac{2^{2R+1}d^{2R}}{C_2} \left(\frac{2^{2R+2}RC(d, R)d^{2R}e}{r}\right)^{\lfloor r/2R \rfloor} \\ &= C_R \left(\frac{2^{2R+2}RC(d, R)d^{2R}e}{r}\right)^{\lfloor r/2R \rfloor}, \end{aligned} \quad (10)$$

where  $C_R$  is a constant that depends only on  $d, R$ . Then there exists an  $r_\epsilon \in \mathbb{N}^+$ , such that for  $r \geq r_\epsilon$ ,

$$C_R \left(\frac{2^{2R+2}RC(d, R)d^{2R}e}{r}\right)^{\lfloor r/2R \rfloor} \leq \epsilon/2. \quad (11)$$

Combining equations (8), (9), (10) and (11) gives us  $\mathbb{P}(\mathcal{G}_{v_0}^n \text{ is a subgraph of } B_r^n(v_0)) \leq \epsilon$  for large enough  $n$  and  $r$ . Note that the choice of  $r_\epsilon$  indeed only depends on  $R, d$  and  $\epsilon$ . Having decided the value of  $r_\epsilon$ , what remains to be done is to select  $n_\epsilon$  large enough to accommodate with this choice of  $r_\epsilon$  such that for  $n \geq n_\epsilon$ , the  $r_\epsilon + R$  neighborhood of  $v_0$  is locally tree-like with high probability as in Lemma 1. Note that since the choice of  $r_\epsilon$  depends only on  $R, d$  and  $\epsilon$ , the choice of  $n_\epsilon$  essentially depends only on  $R, d$  and  $\epsilon$ . This finishes the proof of the first part of Theorem 1.

As for the second part of the theorem, we simply reverse the direction of information flow analysis and everything else remains the same.

## E.2 Proof of Corollary 1

By Theorem 1, for any  $\epsilon > 0$ , there exists  $n_\epsilon, r_\epsilon \in \mathbb{N}^+$ , such that for all  $n \geq n_\epsilon$ ,

$$\begin{aligned} &\mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\mathcal{A}(i; G(n)) = \pi \circ \tau(i)\} \right] \\ &\leq \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n (\mathbb{1}\{\mathcal{A}(i; G(n)) = \pi \circ \tau(i), \mathcal{G}_i^n \text{ is a subgraph of } B_{r_\epsilon}^n(i)\} \right. \\ &\quad \left. + \mathbb{1}\{\mathcal{G}_i^n \text{ is not a subgraph of } B_{r_\epsilon}^n(i)\}) \right] \\ &\leq \epsilon + \sup_{\mathcal{A}'} \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\mathcal{A}'(i; G(n)) = \pi \circ \tau(i)\} \right], \end{aligned}$$

where in the last line  $\mathcal{A}'$  is taken over the family of  $r_\epsilon$ -local algorithms. Lemma 1 implies that the last line above has limiting supremum no larger than  $1/k + \epsilon$  as  $n \rightarrow \infty$  (for detailed arguments, see [KMS16]). Since  $\epsilon$  is arbitrary, then the corollary directly follows.

## F Analysis of streaming belief propagation with side information

In this section we prove Theorem 2. Let  $F(x) = \frac{1}{2} \log \left( \frac{e^{2x}a+b}{e^{2x}b+a} \right)$ ,  $h_1 = \frac{1}{2} \log \frac{1-\alpha}{\alpha}$ , and  $h_2 = \frac{1}{2} \log \frac{\alpha}{1-\alpha}$ . With  $k = 2$ , notice that Algorithm 1 can be equivalently reduced to the following form, with which we will continue our proof:

---

**Algorithm 3**  $R$ -local streaming belief propagation with  $k = 2$

---

- 1: Initialization:  $V(0) = E(0) = G(0) = \emptyset$ .
- 2: **for**  $t = 1, 2, \dots, n$  **do**
- 3:      $V(t) \leftarrow V(t-1) \cup \{v(t)\}$
- 4:      $E(t) \leftarrow E(t-1) \cup \{(v(t), v) : v \in V(t-1), (v(t), t) \in E\}$
- 5:      $G(t) \leftarrow (V(t), E(t))$
- 6:     **for**  $v \in D_1^t(v(t))$  **do**

```

7:       $M_{v \rightarrow v(t)} \leftarrow h_{\tilde{\tau}(v)} + \sum_{v' \in D_1^{t-1}(v)} F(M_{v' \rightarrow v})$ 
8:    end for
9:    for  $r = 1, 2, \dots, R$  do
10:     for  $v \in D_r^t(v(t))$  do
11:       Let  $v' \in D_1^t(v)$  be a vertex which is on a shortest path connecting  $v$  and  $v(t)$ .
12:        $M_{v' \rightarrow v} \leftarrow h_{\tilde{\tau}(v')} + \sum_{v'' \in D_1^t(v') \setminus \{v\}} F(M_{v'' \rightarrow v'})$ 
13:     end for
14:   end for
15: end for
16: for  $u \in V(n)$  do
17:    $M_u \leftarrow h_{\tilde{\tau}(u)} + \sum_{u' \in D_1^n(u)} F(M_{u' \rightarrow u})$ 
18:   Output  $-\mathbb{1}\{M_u \geq 0\} + 2$  as an estimate for  $\tau(u)$ .
19: end for

```

---

We start the proof by introducing the following definition and lemmas:

**Definition 5** (Output of belief propagation). *Let  $T = (V(T), E(T))$  be a tree rooted at  $u$ . Let  $L_T$  be the set of leaves of  $T$ :  $L_T = \{v : v \text{ has degree } 1 \text{ in } T\}$ . For each  $v \in \partial T$ , assume we are given  $M_v^{input} \in \mathbb{R}$  which we refer to as the input belief into  $T$  at  $v$ . For each  $v \in V(T)$ , suppose we observe a noisy label  $\tilde{\tau}(v) \in \{1, 2\}$ , and denote the set of children of  $v$  in  $T$  by  $\mathcal{C}(v)$ . Given the model parameters  $a, b$  and  $\alpha$ ,*

1. For  $v \in \partial T$ , set  $\tilde{M}_{v \rightarrow pa(v)} = M_v^{input}$ , where  $pa(v)$  is the parent vertex of  $v$  in  $T$ .
2. Denote the depth of tree  $T$  by  $R$ . For  $r = R - 1, R - 2, \dots, 1$ , sequentially conduct the following updates: for any  $v \in D_r^n(v_0)$ , let

$$\tilde{M}_{v \rightarrow pa(v)} = h_{\tilde{\tau}(v)} + \sum_{v' \in \mathcal{C}(v)} F(\tilde{M}_{v' \rightarrow v}).$$

3. We define the output of belief propagation  $\text{BPROP}(u; T, \{M_v^{input} : v \in \partial T\}, \tilde{\tau}, \alpha, a, b)$  on the tree  $T$  as follows:

$$\text{BPROP}(u; T, \{M_v^{input} : v \in \partial T\}, \tilde{\tau}, \alpha, a, b) = h_{\tilde{\tau}(u)} + \sum_{u' \in \mathcal{C}(u)} F(\tilde{M}_{u' \rightarrow u}).$$

**Lemma 2.** *Let  $(M_u)_{u \in V(n)}$  be the output of Algorithm 3, under  $\text{StSSBM}(n, 2, a, b, \alpha)$ , for any  $\epsilon > 0$ , there exists  $r_\epsilon, n_\epsilon \in \mathbb{N}^+$ , such that for any  $u \in V(n)$ ,  $n \geq n_\epsilon$ , with probability at least  $1 - \epsilon$ , the following holds:*

$$M_u = \text{BPROP}(u; T, \{M_v^{input} = h_{\tilde{\tau}(v)} : v \in \partial T\}, \tilde{\tau}, \alpha, a, b)$$

for some (random) tree  $T$  rooted at  $u$ , with the depth of  $T$  no larger than  $r_\epsilon$ .

*Proof.* By Lemma 1 and Theorem 1, for any  $\epsilon > 0$ , there exists  $r_\epsilon, n_\epsilon \in \mathbb{N}^+$ , such that with probability no less than  $1 - \epsilon$ , (1)  $\mathcal{G}_u^n$  is a subgraph of  $B_{r_\epsilon}^n(u)$ , so  $M_u$  is a function of  $\tilde{\mathcal{G}}_u^n$  and (2)  $B_{r_\epsilon}^n(u)$  is a tree. The result then follows by observing the iterating formulas of Algorithm 3. Furthermore, if the event just described occurs, then the depth of  $T$  is no larger than  $r_\epsilon$ .  $\square$

**Lemma 3.** *Consider Algorithm 3, denote the value of  $M_{v \rightarrow v'}$  before vertex  $v(t + 1)$  arrives by  $M_{v \rightarrow v'}^t$ . Then the following equation holds for some random time indices  $\{t(v) \in [n] : v \in D_R^n(u)\}$ :*

$$M_u = \text{BPROP}(u; B_R^n(u), \{M_v^{input} = M_{v \rightarrow pa(v)}^{t(v)} : v \in D_R^n(u)\}, \tilde{\tau}, \alpha, a, b).$$

*Proof.* We conduct this proof by induction. Actually, we will show a stronger result. Instead of focusing only on  $R$ -local streaming belief propagation, we will show a more general result for  $N$ -local streaming belief propagation with any  $N \geq R$ . Specifically, we will consider the following algorithm:

1. At each time  $t$ , a new vertex  $v(t)$  is revealed. For  $v \in D_1^t(v(t))$ , define  $M_{v \rightarrow v(t)}(N) = h_{\tilde{\tau}(v)} + \sum_{v' \in D_1^t(v) \setminus \{v(t)\}} F(M_{v' \rightarrow v}(N))$ .
2. For  $r = 1, 2, \dots, N$ , sequentially conduct the following updates: for each  $v \in D_r^t(v)$ , let  $v' \in D_1^t(v)$  be a vertex which is on one of the shortest path connecting  $v(t)$  and  $v$ , update or initialize the value of  $M_{v' \rightarrow v}(N) = h_{\tilde{\tau}(v')} + \sum_{v'' \in D_1^t(v') \setminus \{v\}} F(M_{v'' \rightarrow v'}(N))$ . Since the graph is with high probability locally tree-like, such  $v'$  is with high probability unique.
3. Repeat step 1 and step 2 for all  $1 \leq t \leq n$ .

For  $(v, v') \in E(t)$ , let  $M_{v \rightarrow v'}^t(N)$  denote the value of  $M_{v \rightarrow v'}(N)$  after  $t$ -th iteration obtained by the  $N$ -local streaming belief propagation algorithm. Let  $t_R(u)$  be the first time such that all vertices in  $V_R^n(u)$  have been revealed:

$$t_R(u) := \inf\{t \in [n] : V_R^n(u) = V_R^t(u)\}.$$

Let  $M_u^t(N) = h_{\tilde{\tau}(u)} + \sum_{u' \in D_1^t(u)} F(M_{u' \rightarrow u}^t(N))$ . Now instead of proving Lemma 3, we show that for any  $N \geq R$ , and any  $t \geq t_R(u)$ , if  $B_R^n(u)$  is a tree, then for some random time indices  $\{t(v) \in [n] : v \in D_R^t(u)\}$ , we have

$$M_u^t(N) = \text{BPROP}(u; B_R^n(u), \{M_v^{\text{input}} = M_{v \rightarrow \text{pa}(v)}^{t(v)}(N) : v \in D_R^n(u)\}, \tilde{\tau}, \alpha, a, b). \quad (12)$$

Note that for all  $t \geq t_R(u)$ ,  $M_u^t(R) = M_u^n(R) = M_u$ , therefore the result we have just described is indeed a stronger version of Lemma 3. Then we will prove this stronger result by performing induction on  $R$ . Equation (12) obviously holds for  $R = 1$ . Now suppose equation (12) holds for  $R = r$ , we will show it holds for  $R = r + 1$  by induction. If  $B_{r+1}^n(u)$  is a tree rooted at  $u$ , for any  $N \geq r + 1$ ,  $u' \in D_1^n(u)$ , let  $\bar{T}_{u'}^r$  be the depth  $r$  subtree consisting of  $u'$  and its descendants in  $B_{r+1}^n(u)$ . Let

$$\bar{t}_r(u') = \inf\{t : \bar{T}_{u'}^r \text{ is a subgraph of } B_{r+1}^t(u)\}.$$

For  $u_1, u_2 \in D_1^n(u)$ ,  $u_1 \neq u_2$ , we have  $\bar{t}_r(u_1) \neq \bar{t}_r(u_2)$ , and  $t_*(u) \neq \bar{t}_r(u_1)$ , where  $t_*$  is defined in Section G. For  $u' \in D_1^n(u)$ , by the induction hypothesis, for  $t \geq t_*(u) \vee \bar{t}_r(u')$ , there exists random time indices  $\{t(v) \in [n] : v \text{ being a leaf vertex in } \bar{T}_{u'}^r\}$ , such that

$$M_{u' \rightarrow u}^t(N) = \text{BPROP}(u'; \bar{T}_{u'}^r, \{M_v^{\text{input}} = M_{v \rightarrow \text{pa}(v)}^{t(v)}(N) : v \text{ being a leaf vertex in } \bar{T}_{u'}^r\}, \tilde{\tau}, \alpha, a, b).$$

Since  $t_{r+1}(u) \geq \bar{t}_r(u') \vee t_*(u)$  for all  $u' \in D_1^n(u)$ , then for all  $t \geq t_{r+1}(u)$ , there exists random time indices  $\{t(v) \in [n] : v \in D_{r+1}^t(u)\}$ , such that

$$\begin{aligned} M_u^t(N) &= h_{\tilde{\tau}(u)} + \sum_{u' \in D_1^t(u)} F(M_{u' \rightarrow u}^t(N)) \\ &= \text{BPROP}(u; B_{r+1}^n(u), \{M_v^{\text{input}} = M_{v \rightarrow \text{pa}(v)}^{t(v)}(N) : v \in D_{r+1}^t(u)\}, \tilde{\tau}, \alpha, a, b), \end{aligned}$$

which finishes the proof of this lemma.

## F.1 Proof of Theorem 2

For  $u \in V$ , let  $(\lambda_{r_\epsilon}, T_{u^\epsilon}^{r_\epsilon}) \sim \mathbb{P}_{r_\epsilon, (a+b)/2, b/(a+b)}^T$  where  $T_{u^\epsilon}^{r_\epsilon}$  is the tree and  $\lambda_{r_\epsilon}$  is the set of labels. For vertices in  $T_{u^\epsilon}^{r_\epsilon}$ , we denote the set of noisy labels generated independently with incorrect probability  $\alpha$  by  $\tilde{\lambda}_{r_\epsilon}$ . For any  $\epsilon > 0$ , by Theorem 1 and Lemma 1, there exists  $r_\epsilon, n_\epsilon \in \mathbb{N}^+$ , such that for all  $n \geq n_\epsilon$ , with probability at least  $1 - \epsilon$ : (1)  $\mathcal{G}_u^n$  is a subgraph of  $B_{r_\epsilon}^n(u)$ , (2)  $(\lambda_{r_\epsilon}, \tilde{\lambda}_{r_\epsilon}, T_{u^\epsilon}^{r_\epsilon}) = (\tau(V_{r_\epsilon}^n(u)), \tilde{\tau}(V_{r_\epsilon}^n(u)), B_{r_\epsilon}^n(u))$ . Further there exists a coupling of  $(\lambda_{r_\epsilon}, \tilde{\lambda}_{r_\epsilon}, T_{u^\epsilon}^{r_\epsilon})$  and  $(\tau(V_{r_\epsilon}^n(u)), \tilde{\tau}(V_{r_\epsilon}^n(u)), B_{r_\epsilon}^n(u))$  such that defining

$$S_\epsilon := \left\{ (\lambda_{r_\epsilon}, \tilde{\lambda}_{r_\epsilon}, T_{u^\epsilon}^{r_\epsilon}) = (\tau(V_{r_\epsilon}^n(u)), \tilde{\tau}(V_{r_\epsilon}^n(u)), B_{r_\epsilon}^n(u)), \mathcal{G}_u^n \text{ is a subgraph of } B_{r_\epsilon}^n(u) \right\},$$

then  $\mathbb{P}(S_\epsilon) \geq 1 - \epsilon$  for all  $n \geq n_\epsilon$ . In the following parts of the analysis, we always assume  $S_\epsilon$  occurs. Let  $\mathbb{P}_{bt}$  denote the probability distribution of  $(\lambda_{r_\epsilon}, \tilde{\lambda}_{r_\epsilon}, T_u^{r_\epsilon})$ . According to Lemma 2 and 3, conditioning on  $S_\epsilon$ , there exists  $T_u$  being a tree rooted at  $u$ , such that  $\mathcal{B}_R^n(u)$  is a subgraph of  $T_u$  and  $T_u$  is a subgraph of  $\mathcal{B}_{r_\epsilon}^n(u)$ . Furthermore,  $M_u$  can be expressed as:

$$M_u = \text{BPROP}(u; T_u, \{M_v^{input} = h_{\tilde{\tau}(v)} : v \in \partial T_u\}, \tilde{\tau}, \alpha, a, b) = \frac{1}{2} \log \frac{\mathbb{P}_{bt}(\tau(u) = 1 | T_u, \tilde{\tau}(T_u))}{\mathbb{P}_{bt}(\tau(u) = 2 | T_u, \tilde{\tau}(T_u))},$$

where  $\partial T_u$  is the set of leaf vertices in  $T_u$ ,  $\tilde{\tau}(T_u)$  refers to the set of noisy labels of vertices in  $T_u$ . Then for all  $n \geq n_\epsilon$ , we have

$$\begin{aligned} Q_n(\mathcal{A}_R) &\geq \mathbb{P}(\{\mathcal{A}_R(u; G(n), \tilde{\tau}) = 1, \tau(u) = 1\} \cap S_\epsilon) + \mathbb{P}(\{\mathcal{A}_R(u; G(n), \tilde{\tau}) = 2, \tau(u) = 2\} \cap S_\epsilon) \\ &\geq \frac{1}{2} \mathbb{P}_{bt}(M_u \geq 0 | \tau(u) = 1) + \frac{1}{2} \mathbb{P}_{bt}(M_u < 0 | \tau(u) = 2) - \epsilon \\ &= \frac{1}{2} + \mathbb{E}_{bt} \left[ \left| \mathbb{P}_{bt}(\tau(u) = 1 | T_u, \tilde{\tau}(T_u)) - \frac{1}{2} \right| \right] - \epsilon. \end{aligned}$$

We further have

$$\mathbb{E}_{bt} [\mathbb{P}_{bt}(\tau(u) = 1 | T_u, \tilde{\tau}(T_u)) | \mathcal{B}_R^n(u), \tilde{\tau}(V_R^n(u))] = \mathbb{P}_{bt}(\tau(u) = 1 | \mathcal{B}_R^n(u), \tilde{\tau}(V_R^n(u))).$$

Since  $x \mapsto |x - \frac{1}{2}|$  is convex, therefore, by Jensen's inequality, for all  $n \geq n_\epsilon$ ,

$$Q_n(\mathcal{A}_R) \geq \frac{1}{2} + \mathbb{E}_{bt} \left[ \left| \mathbb{P}_{bt}(\tau(u) = 1 | \mathcal{B}_R^n(u), \tilde{\tau}(V_R^n(u))) - \frac{1}{2} \right| \right] - \epsilon.$$

Since  $\epsilon$  is arbitrary, we have

$$\liminf_{n \rightarrow \infty} \left( Q_n(\mathcal{A}_R) - \frac{1}{2} - \mathbb{E}_{bt} \left[ \left| \mathbb{P}_{bt}(\tau(u) = 1 | \mathcal{B}_R^n(u), \tilde{\tau}(V_R^n(u))) - \frac{1}{2} \right| \right] \right) \geq 0. \quad (13)$$

According to Lemma 3.7 in [MX16], we have

$$\lim_{n \rightarrow \infty} \left| Q_n(\mathcal{A}_R^{\text{off}}) - \frac{1}{2} - \mathbb{E}_{bt} \left[ \left| \mathbb{P}_{bt}(\tau(u) = 1 | \mathcal{B}_R^n(u), \tilde{\tau}(V_R^n(u))) - \frac{1}{2} \right| \right] \right| = 0. \quad (14)$$

Combining equations (13) and (14), we have

$$\liminf_{n \rightarrow \infty} (Q_n(\mathcal{A}_R) - Q_n(\mathcal{A}_R^{\text{off}})) \geq 0,$$

Thus finishes the proof of Theorem 2.

## F.2 Proof of Corollary 2

According to Theorem 2.3 in [MX16], under the three regimes listed in this theorem, we have

$$\lim_{R \rightarrow \infty} \limsup_{n \rightarrow \infty} (Q_n^* - Q_n(\mathcal{A}_R^{\text{off}})) = 0. \quad (15)$$

Combining (15) and Theorem 2, we have

$$\begin{aligned} \limsup_{n \rightarrow \infty} (Q_n^* - Q_n(\mathcal{A}_R)) &\leq \limsup_{n \rightarrow \infty} (Q_n^* - Q_n(\mathcal{A}_R^{\text{off}})) + \limsup_{n \rightarrow \infty} (Q_n(\mathcal{A}_R^{\text{off}}) - Q_n(\mathcal{A}_R)) \\ &\leq \limsup_{n \rightarrow \infty} (Q_n^* - Q_n(\mathcal{A}_R^{\text{off}})), \end{aligned}$$

thus

$$\limsup_{R \rightarrow \infty} \limsup_{n \rightarrow \infty} (Q_n^* - Q_n(\mathcal{A}_R)) \leq \lim_{R \rightarrow \infty} \limsup_{n \rightarrow \infty} (Q_n^* - Q_n(\mathcal{A}_R^{\text{off}})) = 0.$$

Since  $Q_n^* \geq Q_n(\mathcal{A}_R)$ , the other direction naturally holds, thus finishes the proof of Corollary 2.  $\square$

## G Local streaming algorithms with summary statistics

The class of local algorithms is somewhat restrictive. In practice we can imagine keeping a small memory containing global information and updating it each time a new vertex joins. We will not consider general streaming algorithms under a memory constraint; we instead consider a subclass that we name ‘local streaming algorithms with summary statistics’. For instance, in social network setting, such algorithm has access not only to local information as in Definition 1, but also global statistics like the average number of friends.

Formally speaking, the state of the algorithm at time  $t$  is encoded in two vectors  $\mathbf{w}^t = (w_i^t)_{i \in V(t)} \in (\mathbb{R}^m)^{V(t)}$ ,  $\mathbf{e}^t = (e_{ij}^t)_{(i,j) \in E(t)} \in (\mathbb{R}^m)^{E(t)}$ , indexed respectively by the vertices and edges of  $G(t)$ . Here  $m$  is a fixed integer independent of  $n$ . These are initialized to independent random variables  $w_i^{t_*(i)-1} \stackrel{iid}{\sim} P_w$ ,  $e_{ij}^{t_*(i) \vee t_*(j)-1} \stackrel{iid}{\sim} P_e$ , where  $t_*(i)$  is the time at which vertex  $i$  joins the graph ( $v(t_*(i)) = i$ ),  $t_*(i) \vee t_*(j) \triangleq \max\{t_*(i), t_*(j)\}$ , and  $P_w, P_e$  are probability distributions over  $\mathbb{R}^m$ .

At each  $t \in [n]$ , a new vertex  $v(t)$  joins the graph, and a ‘range of action’  $(V_t^{\text{act}}, E_t^{\text{act}})$  is decided, with  $V_t^{\text{act}} \subseteq V(t)$  a vertex set and  $E_t^{\text{act}} \subseteq E(t)$  an edge set. We assume  $(V_t^{\text{act}}, E_t^{\text{act}})$  to depend uniquely on the  $R$ -neighborhood of  $v(t)$ ,  $B_R^t(v(t))$ , and to be such that: (i) the range of action is a subset of the neighborhood:  $V_t^{\text{act}} \subseteq V_R^t(v(t))$ ,  $E_t^{\text{act}} \subseteq E_R^t(v(t))$ ; and (ii) the range of action has bounded size:  $|V_t^{\text{act}}| + |E_t^{\text{act}}| \leq C_{\text{act}} = C_{\text{act}}(R)$  which does not scale with  $n$ . Notice that the second condition is only required because the maximum degree in  $G(n)$  is  $\log n$ , and it is to avoid pathological behavior due to high-degree vertices; we believe it should be possible to avoid it at the cost of extra technical work.

At each time  $t$ , the algorithm updates the quantities  $w_i^t, e_{ij}^t$  in the range of action:

$$\begin{aligned} w_i^t &= F_w^t(\mathbf{w}^{t-1}(V_t^{\text{act}}), \mathbf{e}^{t-1}(E_t^{\text{act}}), \bar{w}^{t-1}, \bar{e}^{t-1}|i), & \forall i \in V_t^{\text{act}}, \\ e_{ij}^t &= F_e^t(\mathbf{w}^{t-1}(V_t^{\text{act}}), \mathbf{e}^{t-1}(E_t^{\text{act}}), \bar{w}^{t-1}, \bar{e}^{t-1}|i, j), & \forall (i, j) \in E_t^{\text{act}}. \end{aligned}$$

Here  $\mathbf{w}^{t-1}(V_t^{\text{act}}), \mathbf{e}^{t-1}(E_t^{\text{act}})$  are the restrictions of  $\mathbf{w}^{t-1}, \mathbf{e}^{t-1}$  to the range of action sets, and  $\bar{w}^{t-1}, \bar{e}^{t-1}$  are summary statistics, updated according to:

$$\bar{w}^t = \frac{1}{|V(t)|} \sum_{v \in V(t)} w_v^t, \quad \bar{e}^t = \frac{1}{|E(t)|} \sum_{(i,j) \in E(t)} e_{ij}^t.$$

Finally, vertex labels are estimated using a function  $\hat{\tau} : \mathbb{R}^m \rightarrow [k]$ . Namely, label at vertex  $v$  is estimated at time  $t$  as  $\hat{\tau}(w_v^t)$ . If  $w_i^t =$  number of friends user  $i$  has at time  $t$ , then  $m = 1$ , and the corresponding summary statistics is the average number of friends per user in the current network.

We next establish that, under mild assumptions, local streaming algorithms with summary statistics cannot achieve non-trivial reconstruction in the symmetric model  $\text{StSSBM}(n, k, a, b)$ . Notice that this claim cannot hold for a general algorithm in this class. Indeed, each node  $i$  could encode the structure of (a bounded-size subgraph)  $B_R^t(i)$  in the decimal expansion of  $w_i^t$ , in such a way that distinct vertices use non-overlapping sets of digits. Then the summary statistics  $\bar{w}^t$  would contain the structure of the whole graph. We avoid this by requiring the update functions to be bounded Lipschitz, and adding a small amount  $\varepsilon$  of noise to  $w_i^t$ , before taking a decision. Informally speaking, this means that we assume small change in the quantitative result can not lead to a big difference in the qualitative outcome, thus adding small perturbation will not have huge affect on the final output.

**Theorem 3.** *Assume that there exist numerical constant  $L_F$ , independent of  $n$ , such that for all  $t \in [n]$ , all  $i \in [t]$  and all  $1 \leq j < l \leq t$ , we have  $\|F_w^t(\cdot | i)\|_\infty, \|F_e^t(\cdot | j, l)\|_\infty \leq L_F$  and  $\|F_w^t(\cdot | i)\|_{\text{Lip}}, \|F_e^t(\cdot | j, l)\|_{\text{Lip}} \leq L_F$ . (Here  $\|f\|_{\text{Lip}}$  denotes the Lipschitz modulus of function  $f$ .) Let  $\{w_i^t : t \leq n, i \in V(t)\}$  be the vertex variables generated by the local streaming algorithm with summary statistics defined by functions  $F_w, F_e$ . Let  $\hat{\tau} : \mathbb{R}^m \rightarrow [k]$  and  $(U_{ij})_{i \leq n, j \leq m} \stackrel{iid}{\sim} \text{Unif}([-1, 1])$  independent of the other randomness. Let  $U_i = (U_{ij})_{j \leq m} \in \mathbb{R}^m$ . Then under  $\text{StSSBM}(n, k, a, b)$ , for any  $\varepsilon > 0$ ,*

$$\limsup_{n \rightarrow \infty} \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}(\hat{\tau}(w_i^n + \varepsilon U_i) = \pi \circ \tau(i)) \right] = \frac{1}{k}.$$



## H Analysis of local streaming algorithms with summary statistics

For the sake of simplicity, we assume  $m = 1$ . We point out that extension to  $m \geq 2$  is straightforward.

### H.1 An auxiliary algorithm

To prove Theorem 3, we first introduce an auxiliary algorithm (Algorithm 4) which is almost a local algorithm. Then we show that the original algorithm can be well approximated by the proposed auxiliary algorithm (Lemma 4). Finally, we show that the auxiliary algorithm can not achieve non-trivial estimation accuracy (Lemma 5).

Let  $\delta \in (0, 1)$  be a small constant independent of  $n$ , and let  $n_h = |\{v \in V(n) : \tau(v) = h\}|$ ,  $h \in [k]$ . We shall perform a global algorithm up to time  $\lceil \delta n \rceil$ , followed by a local algorithm. To represent the information up to time  $\lceil \delta n \rceil$  and the size of communities at time  $n$ , we introduce the following sigma-algebra:

$$\mathcal{F}_a := \sigma\{G(\lceil \delta n \rceil), (v(i), \xi_{v(i)}, \tau(v(i))), w_{v(i)}^{i-1}, e_{v(j)v(s)}^{j \vee s - 1}, n_h : \\ 1 \leq i, j, s \leq \lceil \delta n \rceil, (v(j), v(s)) \in E(\lceil \delta n \rceil), h \in [k]\}.$$

In the auxiliary algorithm, we attach a number  $b_i^t$  to node  $i$  at time  $t$ , with the same initialization as  $w$ :  $b_i^{t_*(i)-1} = w_i^{t_*(i)-1}$ . Similarly, we attach to edge  $(v(i), v(j))$  at time  $t$  a number  $c_{ij}^t = c_{ji}^t$  with initialization  $c_{ij}^{t_*(i) \vee t_*(j) - 1} = e_{ij}^{t_*(i) \vee t_*(j) - 1}$ . Note that  $w_i^{t_*(i)-1}$  and  $e_{ij}^{t_*(i) \vee t_*(j) - 1}$  are as defined in the original algorithm. Denote the vector containing all  $b$ 's attached to vertices in  $V_s^{\text{act}}$  at time  $t$  by  $\mathbf{b}_s^t$ , and the vector containing all  $c$ 's attached to edges in  $E_s^{\text{act}}$  at time  $t$  by  $\mathbf{c}_s^t$ . Similarly, let  $\mathbf{w}_s^t$  and  $\mathbf{e}_s^t$  denote the restrictions of  $\mathbf{w}^t$  and  $\mathbf{e}^t$  to  $V_s^{\text{act}}$  and  $E_s^{\text{act}}$ , respectively. For  $\lceil \delta n \rceil \leq t \leq [n]$ , let

$$\text{average}_b(t) = \frac{1}{t} \sum_{i=1}^t b_{v(i)}^t, \quad \text{average}_c(t) = \frac{1}{|E(t)|} \sum_{(j,k) \in E(t)} c_{jk}^t.$$

Then we introduce the following auxiliary algorithm:

---

#### Algorithm 4 Auxiliary algorithm with summary statistics

---

```

1: for  $i \in V(\lceil \delta n \rceil)$  do
2:    $b_i^{\lceil \delta n \rceil} = w_i^{\lceil \delta n \rceil}$ 
3: end for
4: for  $(i, j) \in E(\lceil \delta n \rceil)$  do
5:    $c_{ij}^{\lceil \delta n \rceil} = c_{ji}^{\lceil \delta n \rceil} = e_{ij}^{\lceil \delta n \rceil}$ 
6: end for
7:  $\bar{b}^{\lceil \delta n \rceil} \leftarrow \text{average}_b(\lceil \delta n \rceil)$ 
8:  $\bar{c}^{\lceil \delta n \rceil} \leftarrow \text{average}_c(\lceil \delta n \rceil)$ 
9: for  $t = \lceil \delta n \rceil + 1, \lceil \delta n \rceil + 2, \dots, n$  do
10:  for  $i \in V_t^{\text{act}}$  do
11:     $b_i^t \leftarrow F_w^t(\mathbf{b}_t^{t-1}, \mathbf{c}_t^{t-1}, \mathbb{E}[\bar{b}^{t-1} | \mathcal{F}_a], \mathbb{E}[\bar{c}^{t-1} | \mathcal{F}_a] | i)$ 
12:  end for
13:  for  $(i, j) \in E_t^{\text{act}}$  do
14:     $c_{ij}^t \leftarrow F_e^t(\mathbf{b}_t^{t-1}, \mathbf{c}_t^{t-1}, \mathbb{E}[\bar{b}^{t-1} | \mathcal{F}_a], \mathbb{E}[\bar{c}^{t-1} | \mathcal{F}_a] | i, j)$ 
15:  end for
16:   $\bar{b}^t \leftarrow \text{average}_b(t)$ 
17:   $\bar{c}^t \leftarrow \text{average}_c(t)$ 
18: end for
19: for  $t = 1, 2, \dots, n$  do
20:  Output  $\hat{\tau}(b_t^n)$  as an estimate for  $\tau(t)$ .
21: end for

```

---

We can prove the following lemmas regarding Algorithm 4:

**Lemma 4.** *Under the conditions stated in Theorem 3, Algorithm 4 and the original algorithm proposed in Section G are asymptotically equivalent, in the sense that as  $n \rightarrow \infty$ ,*

$$\frac{1}{n} \sum_{i=1}^n |w_i^n - b_i^n| + \frac{1}{|E(n)|} \sum_{(i,j) \in E(n)} |e_{ij}^n - c_{ij}^n| \xrightarrow{P} 0.$$

**Lemma 5.** *Under the conditions stated in Theorem 3, for any  $\epsilon > 0$ , the following holds:*

$$\limsup_{\delta \rightarrow 0^+} \limsup_{n \rightarrow \infty} \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\hat{\tau}(b_i^n + \epsilon U_i) = \pi \circ \tau(i)\} \right] = \frac{1}{k}.$$

We defer the proofs of Lemma 4 and 5 to later parts of the appendix. With these two lemmas, we are able to prove Theorem 3.

## H.2 Proof of Theorem 3

For any  $\delta \in (0, 1)$ , using Lemma 4, we conclude that for any  $\zeta \in (0, 1)$ ,  $n$  large enough, with probability at least  $1 - \zeta$ ,  $\frac{1}{n} \sum_{i=1}^n |w_i^n - b_i^n| \leq \zeta^2$ . If this happens, then  $\#\{i : |w_i^n - b_i^n| \geq \zeta\} \leq \zeta n$ . For  $\epsilon$  given in the theorem, let  $\psi_\epsilon(\zeta) := \text{TV}(\epsilon Z_1, \zeta + \epsilon Z_2)$  where  $Z_1, Z_2 \sim \text{Unif}[-1, 1]$  and TV stands for the total variation distance between probability distributions. One can verify that  $\psi_\epsilon(\zeta) \rightarrow 0$  as  $\zeta \rightarrow 0$ . Then we have

$$\begin{aligned} & \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\hat{\tau}(w_i^n + \epsilon U_i) = \pi \circ \tau(i)\} \right] \\ & \leq \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\hat{\tau}(w_i^n + \epsilon U_i) = \pi \circ \tau(i)\} \mathbb{1}\left\{\frac{1}{n} \sum_{i=1}^n |w_i^n - b_i^n| \leq \zeta^2\right\} \right] + \zeta \\ & \leq \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\hat{\tau}(w_i^n + \epsilon U_i) = \pi \circ \tau(i), |w_i^n - b_i^n| \leq \zeta\} \mathbb{1}\left\{\frac{1}{n} \sum_{i=1}^n |w_i^n - b_i^n| \leq \zeta^2\right\} \right] + 2\zeta \end{aligned} \tag{16}$$

$$:= \Delta. \tag{17}$$

Conditioning on given values of  $w_i^n$  and  $b_i^n$ , we may bound the total variation distance between  $w_i^n + \epsilon U_i$  and  $b_i^n + \epsilon U_i$ . Specifically, conditional on all  $w_i^n$  and  $b_i^n$ , there exists  $U_i'^{iid} \sim \text{Unif}[-1, 1]$  for  $i \in [n]$ , independent of  $w_i^n$  and  $b_i^n$ , such that with probability at least  $1 - \psi_\epsilon(|w_i^n - b_i^n|)$ , we have  $w_i^n + \epsilon U_i = b_i^n + \epsilon U_i'$ . Note that  $U_i'$  is independent of  $w_i^n, b_i^n$ , but dependent on  $U_i$ . Then we have

$$\begin{aligned} \Delta & \leq \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\hat{\tau}(b_i^n + \epsilon U_i') = \pi \circ \tau(i), |w_i^n - b_i^n| \leq \zeta\} \mathbb{1}\left\{\frac{1}{n} \sum_{i=1}^n |w_i^n - b_i^n| \leq \zeta^2\right\} \right] \\ & \quad + 2\zeta + \psi_\epsilon(\zeta) \\ & \leq \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\hat{\tau}(b_i^n + \epsilon U_i') = \pi \circ \tau(i)\} \right] + 2\zeta + \psi_\epsilon(\zeta). \end{aligned}$$

Since  $\zeta$  can be arbitrarily small,

$$\begin{aligned} \limsup_{n \rightarrow \infty} \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\hat{\tau}(w_i^n + \epsilon U_i) = \pi \circ \tau(i)\} \right] & \leq \\ \limsup_{n \rightarrow \infty} \mathbb{E} \left[ \max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\hat{\tau}(b_i^n + \epsilon U_i') = \pi \circ \tau(i)\} \right]. & \end{aligned}$$

This holds for any value of  $\delta$ . Taking  $\delta \rightarrow 0^+$  then using Lemma 5 finishes the proof of Theorem 3.

### H.3 Proof of Lemma 4

For simplicity of presentation, in the proof of this lemma we drop the edge attachments (i.e., setting them to zero), and consider only the vertex attachments. We point out that proof involving edge attachments can be conducted almost identically. By the Lipschitz continuity assumption, as the  $(t+1)$ -th vertex joins we have

$$\text{for all } l \in \mathbf{V}_{t+1}^{\text{act}}, \quad |w_l^{t+1} - b_l^{t+1}| \leq L_F \sum_{i \in \mathbf{V}_{t+1}^{\text{act}}} |w_i^t - b_i^t| + L_F |\bar{w}^t - \bar{b}^t| + L_F |\bar{b}^t - \mathbb{E}[\bar{b}^t | \mathcal{F}_a]|, \quad (18)$$

$$\text{for all } l \notin \mathbf{V}_{t+1}^{\text{act}}, \quad |w_l^{t+1} - b_l^{t+1}| = |w_l^t - b_l^t|. \quad (19)$$

Using equations (18) and (19), we have

$$\begin{aligned} |\bar{w}^{t+1} - \bar{b}^{t+1}| &\leq |\bar{w}^t - \bar{b}^t| + \frac{1}{t+1} \sum_{i \in \mathbf{V}_{t+1}^{\text{act}}} |w_i^{t+1} - b_i^{t+1}| + \frac{1}{t+1} \sum_{i \in \mathbf{V}_{t+1}^{\text{act}}} |w_i^t - b_i^t| \\ &\leq \left(1 + \frac{L_F |\mathbf{V}_{t+1}^{\text{act}}|}{t+1}\right) |\bar{w}^t - \bar{b}^t| + \frac{L_F |\mathbf{V}_{t+1}^{\text{act}}| + 1}{t+1} \sum_{i \in \mathbf{V}_{t+1}^{\text{act}}} |w_i^t - b_i^t| + \frac{L_F |\mathbf{V}_{t+1}^{\text{act}}|}{t+1} |\bar{b}^t - \mathbb{E}[\bar{b}^t | \mathcal{F}_a]|. \end{aligned} \quad (20)$$

For  $t \in [n]$ , let  $\mathbf{d}_t \in \mathbb{R}^{n+1}$  with entries indexed from 0 to  $n$ . The first entry is set to  $|\bar{w}^t - \bar{b}^t|$ , and for  $i \in [n]$ , the entry with index  $i$  is set to  $|w_i^t - b_i^t|$ . By definition for all  $1 \leq s \leq \lceil \delta n \rceil$ ,  $\mathbf{d}_s$  has only zero entries.

For  $S_1, S_2, S \subseteq \{0\} \cup [n]$ , let  $\mathbf{1}_S \in \mathbb{R}^{n+1}$  be a vector with entries indexed by 0 to  $n$ , and the entry with index  $\alpha$  is 1 if and only if  $\alpha \in S$  otherwise it is zero. For simplicity, let  $\mathbf{1}_t = \mathbf{1}_{\{t\}}$ , and let  $Q(S_1, S_2) := \mathbf{1}_{S_1} \mathbf{1}_{S_2}^T \in \mathbb{R}^{(n+1) \times (n+1)}$ . Then define the following matrices:

$$\begin{aligned} A_t &:= Q(\mathbf{V}_t^{\text{act}}, \{0\} \cup \mathbf{V}_t^{\text{act}}) + \frac{|\mathbf{V}_t^{\text{act}}| + 1}{t} Q(\{0\}, \{0\} \cup \mathbf{V}_t^{\text{act}}), \\ A_{t_1, t_2} &:= Q(\mathbf{V}_{t_1}^{\text{act}}, \{0\} \cup \mathbf{V}_{t_2}^{\text{act}}) + \frac{|\mathbf{V}_{t_1}^{\text{act}}| + 1}{t_1} Q(\{0\}, \{0\} \cup \mathbf{V}_{t_2}^{\text{act}}). \end{aligned}$$

Without loss of generality we may assume  $L_F \geq 1$ , then combining equations (18), (19) and (20) gives

$$\mathbf{d}_{t+1} \leq (I + L_F A_{t+1}) \mathbf{d}_t + L_F (|\bar{b}^t - \mathbb{E}[\bar{b}^t | \mathcal{F}_a]|) A_{t+1} \mathbf{1}_{v(t+1)},$$

here “ $\leq$ ” refers to element-wise comparison. Let

$$\tilde{\mathbf{d}}_{\lceil \delta n \rceil} = \mathbf{d}_{\lceil \delta n \rceil} = \vec{0}, \quad \tilde{\mathbf{d}}_{t+1} = (I + L_F A_{t+1}) \tilde{\mathbf{d}}_t + (|\bar{b}^t - \mathbb{E}[\bar{b}^t | \mathcal{F}_a]|) L_F A_{t+1} \mathbf{1}_{v(t+1)},$$

then we have  $\mathbf{d}_t \leq \tilde{\mathbf{d}}_t$  for all  $t \geq \lceil \delta n \rceil$ . Furthermore, we have the following decomposition with  $h_t$  defined in equation (22):

$$\langle \mathbf{d}_n, \vec{1} \rangle \leq \langle \tilde{\mathbf{d}}_n, \vec{1} \rangle = \sum_{t=\lceil \delta n \rceil+1}^{n-1} h_t |\bar{b}^t - \mathbb{E}[\bar{b}^t | \mathcal{F}_a]|. \quad (21)$$

Before elaborating on the definition of  $h_t$ , we state the following lemma without proof. Notice that the proof is nothing but basic linear algebra.

**Lemma 6.** For  $n \geq t_m > t_{m-1} > \dots > t_1 \geq \lceil \delta n \rceil + 1$ , we have

$$A(t_m, t_{m-1}, \dots, t_1) := A_{t_m} A_{t_{m-1}} \dots A_{t_1} = \prod_{k=1}^{m-1} \left( \frac{|\mathbf{V}_{t_k}^{\text{act}}| + 1}{t_k} + |\mathbf{V}_{t_k}^{\text{act}} \cap \mathbf{V}_{t_{k+1}}^{\text{act}}| \right) A_{t_m, t_1}.$$

Applying Lemma 6, we have for all  $\lceil \delta n \rceil + 1 \leq t \leq n-1$ ,

$$h_t = \sum_{n \geq t_m > \dots > t_1 = t+1} \langle L_F^m A(t_m, t_{m-1}, \dots, t_1) \mathbf{1}_{v(t+1)}, \vec{1} \rangle$$

$$= \sum_{n \geq t_m > \dots > t_1 = t+1} L_F^m \left( |\mathbf{V}_{t_m}^{\text{act}}| + \frac{|\mathbf{V}_{t_m}^{\text{act}}| + 1}{t_m} \right) \times \prod_{k=1}^{m-1} \left( |\mathbf{V}_{t_k}^{\text{act}} \cap \mathbf{V}_{t_{k+1}}^{\text{act}}| + \frac{|\mathbf{V}_{t_k}^{\text{act}}| + 1}{t_k} \right). \quad (22)$$

For  $\lceil \delta n \rceil + 1 \leq s \leq n-1$ , let  $H(s) = \sum_{t=s}^{n-1} h_t$ , and let

$$C(s) = \sum_{n \geq t_m > \dots > t_1 = s+1} L_F^m \left( |\mathbf{V}_{t_m}^{\text{act}}| + \frac{|\mathbf{V}_{t_m}^{\text{act}}| + 1}{t_m} \right) \times \prod_{k=1}^{m-1} \left( |\mathbf{V}_{t_k}^{\text{act}} \cap \mathbf{V}_{t_{k+1}}^{\text{act}}| + \frac{|\mathbf{V}_{t_k}^{\text{act}}| + 1}{t_k} \right) \mathbb{1} \left\{ |\mathbf{V}_{t_k}^{\text{act}} \cap \mathbf{V}_{t_{k+1}}^{\text{act}}| > 0 \right\}.$$

Then we can provide an upper bound for  $H(s)$  using  $C(s)$  and  $H(s+1)$ :

$$H(s) \leq \left( 1 + \frac{1}{\lceil \delta n \rceil} C(s) \right) H(s+1) + C(s).$$

By induction, and applying the fact that  $\log(1+x) \leq x$  for all  $x > -1$ , we have

$$H_{\lceil \delta n \rceil + 1} \leq \sum_{s=\lceil \delta n \rceil + 1}^{n-1} C(s) \prod_{s=\lceil \delta n \rceil + 1}^{n-1} \left( 1 + \frac{C(s)}{\lceil \delta n \rceil} \right) \leq \sum_{s=\lceil \delta n \rceil + 1}^{n-1} C(s) \exp \left( \frac{1}{\lceil \delta n \rceil} \sum_{s=\lceil \delta n \rceil + 1}^{n-1} C(s) \right). \quad (23)$$

Using equation (21), we can prove Lemma 4 if we can prove the following two lemmas.

**Lemma 7.** *Under the assumptions stated in Theorem 3, for all  $\delta > 0$ , we have  $H_{\lceil \delta n \rceil + 1} = O_p(n)$ .*

**Lemma 8.** *Under the assumptions stated in Theorem 3, we have*

$$\sup_{\lceil \delta n \rceil + 1 \leq t \leq n} |\bar{b}^t - \mathbb{E}[\bar{b}^t | \mathcal{F}_a]| = o_p(1).$$

The proof of Lemma 7 and 8 will be deferred to later parts of the appendix. Using Lemma 7 and 8, then apply equation (21), we have

$$\frac{1}{n} \sum_{i=1}^n |w_i^t - b_i^t| \leq \frac{1}{n} \langle \mathbf{d}_n \bar{\mathbf{1}} \rangle \leq \frac{1}{n} H(\lceil \delta n \rceil) \sup_{\lceil \delta n \rceil + 1 \leq t \leq n-1} |\bar{b}^t - \mathbb{E}[\bar{b}^t | \mathcal{F}_a]| = o_p(1),$$

this finishes the proof of Lemma 4.

#### H.4 Proof of Lemma 7

To prove Lemma 7 we shall first provide a uniform upper bound for the expectation of  $C(s)$  which is independent of  $n, \delta$  and  $s$ . If this holds, then by Markov's inequality  $\sum_{s=\lceil \delta n \rceil + 1}^n C(s) = O_p(n)$ .

Plugging this into equation (23) gives  $H_{\lceil \delta n \rceil + 1} = O_p(n)$ , which finishes the proof of this lemma. Let

$\delta_1 := \frac{1}{2} \left( \frac{(a+(k-1)b)\delta}{k} \wedge 1 \right) \delta$ . For  $n$  large enough, we have  $\frac{|\mathbf{V}_{t_k}^{\text{act}}| + 1}{t_k} < 1$  for all  $\lceil \delta n \rceil + 1 \leq t_k \leq n$ , then we have

$$\begin{aligned} C(s) &\leq \sum_{n \geq t_m > \dots > t_1 = s+1} 2^m L_F^m C_{act}^m \mathbb{1} \left\{ |\mathbf{V}_{t_k}^{\text{act}} \cap \mathbf{V}_{t_{k+1}}^{\text{act}}| > 0 \quad k = 1, 2, \dots, m-1 \right\} \\ &\leq \sum_{n \geq t_m > \dots > t_1 = s+1} 2^m L_F^m C_{act}^m \mathbb{1} \left\{ d(v(t_k), v(t_{k+1})) \leq 2R, \quad k = 1, 2, \dots, m-1 \right\}. \end{aligned} \quad (24)$$

Note that in equation (24),  $m$  can be any positive integer, therefore, taking the expectation of  $C(s)$  gives

$$\begin{aligned} \mathbb{E}[C(s)] &\leq \sum_{m=1}^{\infty} \sum_{n \geq t_m > \dots > t_1 = s+1} 2^m L_F^m C_{act}^m \mathbb{P}(d(v(t_k), v(t_{k+1})) \leq 2R, \quad k = 1, 2, \dots, m-1) \\ &\leq \sum_{m=1}^{\infty} \sum_{n \geq t_m > \dots > t_1 = s+1} 2^m L_F^m C_{act}^m \mathbb{E} \left[ \prod_{k=1}^{m-1} \frac{|V_{2R}^n(v(t_k))|}{n-k} \right] \end{aligned}$$

$$\leq \sum_{m=1}^{\infty} \frac{2^m L^m C_{act}^m}{(m-1)!} \mathbb{E}[|V_{2R}^n(v)|^m],$$

where  $v$  is an arbitrary vertex in  $V(n)$ . The following lemma provides an upper bound on  $\mathbb{E}[|V_{2R}^n(v)|^m]$ .

**Lemma 9.** *Under the assumptions of Theorem 3, we have*

$$\limsup_{n \rightarrow \infty} \sum_{m=1}^{\infty} \frac{2^m L_F^m C_{act}^m}{(m-1)!} \mathbb{E}[|V_{2R}^n(v)|^m] < \infty.$$

The proof of Lemma 9 is deferred to next subsection. With Lemma 9, we can apply Markov's inequality to prove  $\sum_{s=\lceil \delta n \rceil + 1}^n C(s) = O_p(n)$ , then Lemma 7 follows from equation (23).

## H.5 Proof of Lemma 9

To prove lemma 9, we introduce the following branching process:

1.  $X_0 = 1$ .
2. Let  $\{Z(i, t) : i, t \in \mathbb{Z}^+\}$  be an array of i.i.d.  $\mathbb{Z}^+$ -valued random variables with distribution Binomial( $n, \frac{a \vee b}{n}$ ). For  $t \geq 1$ , define  $X_t = \sum_{1 \leq i \leq X_{t-1}} Z(i, t)$ .

Then we have

$$\mathbb{E}[|V_{2R}^n(v)|^m] \leq \mathbb{E}\left[\left(\sum_{k=0}^{2R} X_k\right)^m\right]. \quad (25)$$

Let  $f_j(t) := \mathbb{E}[\exp(tX_j)]$ ,  $\mu_m := a \vee b$ . By the proof of Theorem 2.3.1 in [Ver18], we have  $f_1(t) \leq \exp(\mu_m(e^t - 1))$ . For  $\gamma > 0$ , let  $t_1^\gamma := \log\left(\frac{\gamma}{\mu_m} + 1\right)$ , and  $t_{j+1}^\gamma := \log\left(\frac{t_j^\gamma}{\mu_m} + 1\right)$  for  $j \in \mathbb{N}^+$ . Then by Proposition 5.2 in [LP17], we have

$$f_{j+1}(t_{j+1}^\gamma) \leq f_j(t_j^\gamma) \leq \dots \leq f_1(t_1^\gamma) \leq \exp(\gamma).$$

Then for all  $1 \leq j \leq 2R$  and  $\gamma_0 > 0$ , we have

$$\begin{aligned} \mathbb{E}[|X_j^m|] &= \int_0^\infty m\gamma^{m-1} \mathbb{P}(X_j \geq \gamma) d\gamma \\ &\leq \int_0^\infty m\gamma^{m-1} \exp(-\gamma t_j^\gamma + \gamma) d\gamma \\ &= \int_0^{\gamma_0} m\gamma^{m-1} \exp(-\gamma t_j^\gamma + \gamma) d\gamma + \int_{\gamma_0}^\infty m\gamma^{m-1} \exp(-\gamma t_j^\gamma + \gamma) d\gamma \\ &\leq \gamma_0^m \exp(\gamma_0) + \frac{m!}{(t_j^{\gamma_0} - 1)^m}. \end{aligned} \quad (26)$$

By equation (26), for any  $\gamma_0 > 0$ , we have

$$\mathbb{E}\left[\left(\sum_{k=0}^{2R} X_k\right)^m\right] \leq (2R+1)^m \gamma_0^m \exp(\gamma_0) + (2R+1)^{m-1} \sum_{j=1}^{2R} \frac{m!}{(t_j^{\gamma_0} - 1)^m} + (2R+1)^{m-1}. \quad (27)$$

Notice that we can choose  $\gamma_0$  large enough, such that  $\min_{1 \leq j \leq 2R} t_j^{\gamma_0} - 1 \geq 4L_F C_{act}(2R+1)$ . Using equations (25) and (27), we have

$$\sum_{m=1}^{\infty} \frac{2^m L_F^m C_{act}^m}{(m-1)!} \mathbb{E}[|V_{2R}^n(v)|^m]$$

$$\leq \underbrace{\sum_{m=1}^{\infty} \frac{2^m L_F^m C_{act}^m}{(m-1)!} (2R+1)^m (\gamma_0^m \exp(\gamma_0) + 1)}_I + \underbrace{\sum_{m=1}^{\infty} \frac{2^m L_F^m C_{act}^m (2R+1)^m m}{(\min_{1 \leq j \leq 2R} t_j^{\gamma_0} - 1)^m}}_II.$$

The choice of  $\gamma_0$  gives us equation II  $\leq \sum_{m=1}^{\infty} \frac{m}{2^m} < \infty$ . One can also derive equation I  $< \infty$  for any value of  $\gamma_0$ . Note that the derived upper bounds for equations I and II are independent of  $n$ . Thus we have finished proving Lemma 9.

## H.6 Proof of Lemma 8

We first show a weaker result. As  $n \rightarrow \infty$ , we want to show  $\sup_{\lceil \delta n \rceil + 1 \leq t \leq n} \mathbb{E}[|\bar{b}^t - \mathbb{E}[\bar{b}^t | \mathcal{F}_a]|^2] \rightarrow 0$ . Since the  $|F_w|$  functions are uniformly bounded by  $L_F$ , we only need to show as  $n \rightarrow \infty$ ,

$$\sup_{\lceil \delta n \rceil + 1 \leq t \leq n} \sup_{1 \leq x < y \leq t} \left| \mathbb{E} \left[ (b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t | \mathcal{F}_a]) (b_{v(y)}^t - \mathbb{E}[b_{v(y)}^t | \mathcal{F}_a]) \right] \right| \rightarrow 0. \quad (28)$$

For  $1 \leq x < y \leq n$ ,  $r \in \mathbb{N}^+$ , we introduce the following sets in the probability space we consider:

$$A_x^r = \left\{ \mathcal{G}_{v(x)}^n \text{ is a subgraph of } B_r^n(v(x)) \right\}, \quad B_{x,y}^r = \left\{ V_{r+1}^n(v(x)) \cap V_{r+1}^n(v(y)) = \emptyset \right\}.$$

Then we have

$$|\mathbb{E}[(b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t | \mathcal{F}_a]) (b_{v(y)}^t - \mathbb{E}[b_{v(y)}^t | \mathcal{F}_a])]| \quad (29)$$

$$\leq 4L_F^2 \left( \mathbb{P}((B_{x,y}^r)^c) + \mathbb{P}((A_x^r)^c) + \mathbb{P}((A_y^r)^c) \right) \quad (30)$$

$$+ \underbrace{|\mathbb{E}[(b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t | \mathcal{F}_a]) (b_{v(y)}^t - \mathbb{E}[b_{v(y)}^t | \mathcal{F}_a]) \mathbb{1}_{A_x^r \cap A_y^r \cap \mathbb{1}_{B_{x,y}^r}}]|}_{C(x,y,t,n,r)}.$$

Then we provide an upper bound for  $C(x, y, t, n, r)$ . The following procedures are conducted **conditioning only on  $\mathcal{F}_a$** . We point out that  $B_{r+1}^n(v(x))$  can be generated via the following procedure, which is equivalent to our proposed model  $\text{StSSBM}(n, k, a, b)$ .

1. If  $x \geq \lceil \delta n \rceil$ , then arbitrarily pick  $v(x)$  in  $V \setminus V(\lceil \delta n \rceil)$ . Otherwise,  $v(x)$  is already specified by  $\mathcal{F}_a$ .
2. Conditioning on  $\mathcal{F}_a$  and  $v(x)$ , generate  $B_{r+1}^n(v(x))$  following conditional distribution of  $\text{StSSBM}(n, k, a, b)$ .
3. For  $v \in V_{r+1}^n(v(x))$ , if  $t_*(v)$  is not yet specified, then assign it a value based on the current conditional distribution.

Given  $B_{r+1}^n(v(x))$  and the orderings of vertices within, we are able to judge whether  $\mathcal{G}_{v(x)}^n$  is a subgraph of  $B_r^n(v(x))$ . If  $\mathcal{G}_{v(x)}^n$  is not a subgraph of  $B_r^n(v(x))$ , then formula inside the expectation of  $C(x, y, t, n, r)$  is zero. If  $\mathcal{G}_{v(x)}^n$  is a subgraph of  $B_r^n(v(x))$ , then we consider another graph  $G_{new}$  which also follows  $\text{StSSBM}(n, k, a, b)$ :

1. We assume that conditioning on  $\mathcal{F}_a$ ,  $G_{new}$  and  $G(n)$  are exactly the same.
2. Again conditioning on  $\mathcal{F}_a$ , we can find a bijection  $\phi$  mapping vertices in  $G(n)$  to vertices in  $G_{new}$ : if  $v \in V(\lceil \delta n \rceil)$ , then  $\phi$  maps  $v$  to the vertex in  $G_{new}$  with reveal ordering  $t_*(v)$ ; otherwise, map  $v$  to an arbitrary vertex in  $G_{new}$  with the same community label. For  $S \subseteq V(n)$ , denote the image set of  $S$  obtained via mapping  $\phi$  by  $\phi(S)$ .
3. Among all unordered vertices in  $G_{new}$ , we randomly pick  $\tilde{v}(y)$  and assign it the revealing ordering  $y$ .
4. Generate the  $r+1$  neighborhood of  $\tilde{v}(y)$  in  $G_{new}$ , following the conditional distribution of  $\text{StSSBM}(n, k, a, b)$  conditioning on  $\mathcal{F}_a$ , and denote it by  $B_{r+1}^n(\tilde{v}(y))$ . For vertices in  $B_{r+1}^n(\tilde{v}(y))$  which have not been assigned orderings, we randomly assign them one.

In both  $G(n)$  and  $G_{new}$ , we denote the  $r$  neighborhood of  $v$  with the information about revealing orderings of vertices within encoded by  $O_r^n(v)$ . Notice that conditioning on  $\mathcal{F}_a$ ,  $O_{r+1}^n(\tilde{v}(y))$  is conditionally independent of  $O_{r+1}^n(v(x))$ . Let

$$\text{NonOverlap}(x, y) =$$

$\{\text{revealing orderings in } O_{r+1}^n(\tilde{v}(y)) \text{ does not overlap with revealing orderings in } O_{r+1}^n(v(x))\}$ .

Then we have

$$\begin{aligned} & \mathcal{L}(O_{r+1}^n(v(y))|O_{r+1}^n(v(x)), V_{r+1}^n(v(x)) \cap V_{r+1}^n(v(y)) = \emptyset, \mathcal{F}_a) \\ &= \mathcal{L}(O_{r+1}^n(\tilde{v}(y))|O_{r+1}^n(v(x)), V_{r+1}^n(\tilde{v}(y)) \cap \phi(V_{r+1}^n(v(x))) = \emptyset, \text{NonOverlap}(x, y), \mathcal{F}_a). \end{aligned}$$

where  $\mathcal{L}(\cdot|\cdot)$  denotes the conditional probability law. Furthermore, the marginal probability distributions are equal:  $\mathcal{L}(O_{r+1}^n(v(y))|\mathcal{F}_a) = \mathcal{L}(O_{r+1}^n(\tilde{v}(y))|\mathcal{F}_a)$ . Define the following sets:

$$\begin{aligned} \tilde{A}_y^r &= \left\{ \mathcal{G}_{\tilde{v}(y)}^n \text{ is a subgraph of } \mathbf{B}_r^n(\tilde{v}(y)) \right\}, \\ \tilde{C}_{x,y}^r &= \left\{ V_{r+1}^n(\tilde{v}(y)) \cap \phi(V_{r+1}^n(v(x))) = \emptyset \right\} \cap \text{NonOverlap}(x, y). \end{aligned}$$

In  $G_{new}$  we can still run Algorithm 4, and denote the obtained quantity at time  $t$  for vertex  $\tilde{v}(y)$  by  $\tilde{b}_{\tilde{v}(y)}^t$ . Then we have

$$\begin{aligned} & \left| \mathbb{E}[(b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t|\mathcal{F}_a])\mathbb{1}_{A_x^r}\mathbb{1}_{B_{x,y}^r}(b_{v(y)}^t - \mathbb{E}[b_{v(y)}^t|\mathcal{F}_a])\mathbb{1}_{A_y^r}|\mathcal{F}_a] \right. \\ &= \left| \mathbb{E}[(b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t|\mathcal{F}_a])\mathbb{1}_{A_x^r}\mathbb{1}_{B_{x,y}^r}\mathbb{E}[(b_{v(y)}^t - \mathbb{E}[b_{v(y)}^t|\mathcal{F}_a])\mathbb{1}_{A_y^r}|O_{r+1}^n(v(x)), B_{x,y}^r, \mathcal{F}_a]|\mathcal{F}_a] \right. \\ &= \left| \mathbb{E}[(b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t|\mathcal{F}_a])\mathbb{1}_{A_x^r}\mathbb{1}_{B_{x,y}^r}\mathbb{E}[(\tilde{b}_{\tilde{v}(y)}^t - \mathbb{E}[\tilde{b}_{\tilde{v}(y)}^t|\mathcal{F}_a])\mathbb{1}_{\tilde{A}_y^r}|O_{r+1}^n(v(x)), \tilde{C}_{x,y}^r, \mathcal{F}_a]|\mathcal{F}_a] \right. \\ &= \left| \mathbb{E} \left[ (b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t|\mathcal{F}_a])\mathbb{1}_{A_x^r}\mathbb{1}_{B_{x,y}^r} \frac{\mathbb{E}[(\tilde{b}_{\tilde{v}(y)}^t - \mathbb{E}[\tilde{b}_{\tilde{v}(y)}^t|\mathcal{F}_a])\mathbb{1}_{\tilde{A}_y^r}\mathbb{1}_{\tilde{C}_{x,y}^r}|O_{r+1}^n(v(x)), \mathcal{F}_a]}{\mathbb{P}(\tilde{C}_{x,y}^r|O_{r+1}^n(v(x)), \mathcal{F}_a)} \Big| \mathcal{F}_a \right] \right| \end{aligned} \quad (31)$$

Obviously  $\mathbb{P}(\tilde{C}_{x,y}^r|O_{r+1}^n(v(x)), \mathcal{F}_a) \xrightarrow{P} 1$ , also notice that  $\|F_w\|_\infty$  has a uniform upper bound, then equation (31) is no larger than

$$\begin{aligned} & \left| \underbrace{\mathbb{E} \left[ (b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t|\mathcal{F}_a])\mathbb{1}_{A_x^r}\mathbb{1}_{B_{x,y}^r} (\tilde{b}_{\tilde{v}(y)}^t - \mathbb{E}[\tilde{b}_{\tilde{v}(y)}^t|\mathcal{F}_a])\mathbb{1}_{\tilde{A}_y^r}\mathbb{1}_{\tilde{C}_{x,y}^r} \Big| \mathcal{F}_a \right]}_{\Delta} \right| + \\ & 4L_F^2 \mathbb{E} \left[ \left| 1 - \mathbb{P}(\tilde{C}_{x,y}^r|O_{r+1}^n(v(x)), \mathcal{F}_a)^{-1} \right| \wedge 1 \mid \mathcal{F}_a \right]. \end{aligned} \quad (32)$$

Notice that

$$\begin{aligned} & \left| \Delta - \mathbb{E} \left[ (b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t|\mathcal{F}_a])\mathbb{1}_{A_x^r}\mathbb{1}_{B_{x,y}^r} (\tilde{b}_{\tilde{v}(y)}^t - \mathbb{E}[\tilde{b}_{\tilde{v}(y)}^t|\mathcal{F}_a])\mathbb{1}_{\tilde{A}_y^r}\mathbb{1}_{\tilde{C}_{x,y}^r} \Big| \mathcal{F}_a \right] \right| \leq \\ & 4L_F^2 \mathbb{P}((\tilde{A}_y^r)^c|\mathcal{F}_a) + 4L_F^2 \mathbb{P}((\tilde{C}_{x,y}^r)^c|\mathcal{F}_a) \end{aligned}$$

Conditioning on  $\mathcal{F}_a$ ,  $(\tilde{b}_{\tilde{v}(y)}^t - \mathbb{E}[\tilde{b}_{\tilde{v}(y)}^t|\mathcal{F}_a])$  is conditionally independent of  $(b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t|\mathcal{F}_a])\mathbb{1}_{A_x^r}\mathbb{1}_{B_{x,y}^r}$ . Combining this and equations (29), (32), we have

$$\begin{aligned} & \left| \mathbb{E}[(b_x^t - \mathbb{E}[\bar{b}^t|\mathcal{F}_a])(b_y^t - \mathbb{E}[\bar{b}^t|\mathcal{F}_a]) \right. \\ & \leq 4L_F^2 \mathbb{P}((B_{x,y}^r)^c) + 4L_F^2 \mathbb{P}((A_y^r)^c) + 4L_F^2 \mathbb{P}((A_x^r)^c) + 4L_F^2 \mathbb{P}((\tilde{A}_y^r)^c) + 4L_F^2 \mathbb{P}((\tilde{C}_{x,y}^r)^c) + \\ & 4L_F^2 \mathbb{E} \left[ \left| 1 - \mathbb{P}(\tilde{C}_{x,y}^r|O_{r+1}^n(v(x)), \mathcal{F}_a)^{-1} \right| \wedge 1 \right]. \end{aligned}$$

According to the proof of Theorem 1 we have

$$\limsup_{n \rightarrow \infty} \sup_{x,y \in [n]} \left\{ 4L_F^2 \mathbb{P}((A_x^r)^c) + 4L_F^2 \mathbb{P}((A_y^r)^c) + 4L_F^2 \mathbb{P}((\tilde{A}_y^r)^c) \right\} = 0.$$

Furthermore, we claim without proof that

$$\begin{aligned} & \limsup_{n \rightarrow \infty} \sup_{x,y \in [n]} \left\{ 4L_F^2 \mathbb{P}((B_{x,y}^r)^c) + 4L_F^2 \mathbb{P}((\tilde{C}_{x,y}^r)^c) \right\} = 0, \\ & \limsup_{n \rightarrow \infty} \sup_{x,y \in [n]} 4L_F^2 \mathbb{E} \left[ \left| 1 - \mathbb{P}(\tilde{C}_{x,y}^r|O_{r+1}^n(v(x)), \mathcal{F}_a)^{-1} \right| \wedge 1 \right] = 0 \end{aligned}$$

Notice that these asymptotic results do not depend on  $x, y$  or  $t$ , therefore,

$$\limsup_{n \rightarrow \infty} \sup_{\lceil \delta n \rceil + 1 \leq t \leq n} \sup_{1 \leq x < y \leq t} \left| \mathbb{E} \left[ (b_{v(x)}^t - \mathbb{E}[b_{v(x)}^t | \mathcal{F}_a]) (b_{v(y)}^t - \mathbb{E}[b_{v(y)}^t | \mathcal{F}_a]) \right] \right| = 0.$$

Then we conclude that equation (28) holds, and Lemma 8 follows from a discretization argument. Furthermore, from the proof of Lemma 8 we can deduce the following corollary:

**Corollary 3.** *Assume  $(\tau, \mathbf{G}) \sim \text{StSSBM}(n, k, a, b)$ . Let  $\mathcal{A}$  be any algorithm such that  $\mathcal{A}(i; G(n)) \in [k]$  is a function of  $\mathcal{G}_i^n$  and  $\mathcal{F}_a$ . Then we have as  $n \rightarrow \infty$ ,*

$$\sup_{\mathcal{A}} \text{Var} \left[ \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\mathcal{A}(i; G(n)) = \tau(i)\} \mid \mathcal{F}_a \right] \xrightarrow{P} 0.$$

*Proof.* To prove this corollary, we first show for all  $x, y \in [n]$ ,

$$C(x, y) := \sup_{\mathcal{A}} |\mathbb{P}(\mathcal{A}(v(x); G(n)) = \tau(v(x)), \mathcal{A}(v(y); G(n)) = \tau(v(y)) \mid \mathcal{F}_a) - \mathbb{P}(\mathcal{A}(v(x); G(n)) = \tau(v(x)) \mid \mathcal{F}_a) \mathbb{P}(\mathcal{A}(v(y); G(n)) = \tau(v(y)) \mid \mathcal{F}_a)| \xrightarrow{P} 0. \quad (33)$$

Consider the graph generating procedure described in the proof of Lemma 8 and we use the notations defined there. Furthermore, we define the following quantities (here we use tildes to represent objects in  $G_{new}$ ):

$$\Delta_1 := \mathbb{E} \left[ \mathbb{1}\{\mathcal{A}(v(x); G(n)) = \tau(v(x))\} \mathbb{1}_{A_x^r} \mathbb{1}_{B_{x,y}^r} \mathbb{1}\{\mathcal{A}(v(y); G(n)) = \tau(v(y))\} \mathbb{1}_{A_y^r} \mid \mathcal{F}_a \right],$$

$$\Delta_2 := \mathbb{E} \left[ \mathbb{1}\{\mathcal{A}(v(x); G(n)) = \tau(v(x))\} \mathbb{1}_{A_x^r} \mathbb{1}_{B_{x,y}^r} \mathbb{1}\{\mathcal{A}(\tilde{v}(y); \tilde{G}(n)) = \tau(\tilde{v}(y))\} \mathbb{1}_{\tilde{A}_y^r} \mathbb{1}_{\tilde{C}_{x,y}^r} \mid \mathcal{F}_a \right].$$

Similar to the proof of Lemma 8, we conclude that the following formulas hold:

$$\begin{aligned} & |\mathbb{P}(\mathcal{A}(v(x); G(n)) = \tau(v(x)), \mathcal{A}(v(y); G(n)) = \tau(v(y)) \mid \mathcal{F}_a) - \Delta_1| \\ & \leq \mathbb{P}((A_x^r)^c \mid \mathcal{F}_a) + \mathbb{P}((A_y^r)^c \mid \mathcal{F}_a) + \mathbb{P}((B_{x,y}^r)^c \mid \mathcal{F}_a), \end{aligned}$$

$$\begin{aligned} \Delta_1 &= \mathbb{E}[\mathbb{1}\{\mathcal{A}(v(x); G(n)) = \tau(v(x))\} \mathbb{1}_{A_x^r} \mathbb{1}_{B_{x,y}^r} \\ & \quad \mathbb{E}[\mathbb{1}\{\mathcal{A}(\tilde{v}(y); \tilde{G}(n)) = \tau(\tilde{v}(y))\} \mathbb{1}_{\tilde{A}_y^r} \mid O_{r+1}^n(v(x)), \tilde{C}_{x,y}^r, \mathcal{F}_a] \mid \mathcal{F}_a] \\ &= \mathbb{E}[\mathbb{1}\{\mathcal{A}(v(x); G(n)) = \tau(v(x))\} \mathbb{1}_{A_x^r} \mathbb{1}_{B_{x,y}^r} \\ & \quad \frac{\mathbb{E}[\mathbb{1}\{\mathcal{A}(\tilde{v}(y); \tilde{G}(n)) = \tau(\tilde{v}(y))\} \mathbb{1}_{\tilde{A}_y^r} \mathbb{1}_{\tilde{C}_{x,y}^r} \mid O_{r+1}^n(v(x)), \mathcal{F}_a]}{\mathbb{P}(\tilde{C}_{x,y}^r \mid O_{r+1}^n(v(x)), \mathcal{F}_a)} \mid \mathcal{F}_a], \\ |\Delta_1 - \Delta_2| &\leq \mathbb{E} \left[ \left| \mathbb{P}(\tilde{C}_{x,y}^r \mid O_{r+1}^n(v(x)), \mathcal{F}_a)^{-1} - 1 \right| \wedge 1 \mid \mathcal{F}_a \right], \\ |\Delta_2 - \mathbb{P}(\mathcal{A}(v(x); G(n)) = \tau(v(x)) \mid \mathcal{F}_a) \mathbb{P}(\mathcal{A}(v(y); G(n)) = \tau(v(y)) \mid \mathcal{F}_a)| \\ &\leq \mathbb{P}((A_x^r)^c \mid \mathcal{F}_a) + \mathbb{P}((B_{x,y}^r)^c \mid \mathcal{F}_a) + \mathbb{P}((\tilde{A}_y^r)^c \mid \mathcal{F}_a) + \mathbb{P}((\tilde{C}_{x,y}^r)^c \mid \mathcal{F}_a). \end{aligned}$$

Combining the above equations we have

$$\begin{aligned} C(x, y) &\leq 2\mathbb{P}((A_x^r)^c \mid \mathcal{F}_a) + 2\mathbb{P}((B_{x,y}^r)^c \mid \mathcal{F}_a) + 2\mathbb{P}((\tilde{A}_y^r)^c \mid \mathcal{F}_a) + \mathbb{P}((\tilde{C}_{x,y}^r)^c \mid \mathcal{F}_a) \\ &\quad + \mathbb{E} \left[ \left| \mathbb{P}(\tilde{C}_{x,y}^r \mid O_{r+1}^n(v(x)), \mathcal{F}_a)^{-1} - 1 \right| \wedge 1 \mid \mathcal{F}_a \right]. \quad (34) \end{aligned}$$

The upper bound on the right hand side of equation (34) is independent of  $\mathcal{A}$ , and converges in probability to zero by Theorem 1. Thus we have finished proving (33). Furthermore, similar to the proof of Lemma 8, from (34) we can conclude that as  $n \rightarrow \infty$ ,

$$\sup_{x, y \in [n]} \mathbb{E}[C(x, y)] \rightarrow 0,$$

thus we finishes the proof of this corollary by Markov's inequality.  $\square$



## H.7 Proof of Lemma 5

For any  $\pi \in \mathfrak{S}_k$  and  $\delta > 0$ , using Corollary 3, we have

$$\frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\hat{\tau}(b_i^n + \epsilon U_i) = \pi \circ \tau(i)\} = \frac{1}{n} \sum_{i=1}^n \mathbb{P}(\hat{\tau}(b_i^n + \epsilon U_i) = \pi \circ \tau(i) \mid \mathcal{F}_a) + o_p(1).$$

If we can show the following equation for all  $\pi \in \mathfrak{S}_k$ , then we finishes the proof of Lemma 5.

$$\limsup_{\delta \rightarrow 0^+} \limsup_{n \rightarrow \infty} \mathbb{E} \left[ \left| \frac{1}{n} \sum_{i=1}^n \mathbb{P}(\hat{\tau}(b_i^n + \epsilon U_i) = \pi \circ \tau(i) \mid \mathcal{F}_a) - \frac{1}{k} \right| \right] = 0.$$

Since

$$\mathbb{E} \left[ \left| \frac{1}{n} \sum_{i=1}^n \mathbb{P}(\hat{\tau}(b_i^n + \epsilon U_i) = \pi \circ \tau(i) \mid \mathcal{F}_a) - \frac{1}{k} \right| \right] \leq \frac{1}{n} \sum_{i=1}^n \sum_{h=1}^k \mathbb{E} \left[ \left| \mathbb{P}(\tau(i) = h \mid \mathcal{F}_a, b_i^n) - \frac{1}{k} \right| \right],$$

then we only need to show for all  $1 \leq i \leq n$  and all  $h \in [k]$ ,

$$\limsup_{\delta \rightarrow 0^+} \limsup_{n \rightarrow \infty} \mathbb{E} \left[ \left| \mathbb{P}(\tau(i) = h \mid \mathcal{F}_a, b_i^n) - \frac{1}{k} \right| \right] = 0. \quad (35)$$

By Theorem 1, for any  $\epsilon > 0$ , there exists  $r_\epsilon \in \mathbb{N}^+$ , such that  $\limsup_{n \rightarrow \infty} \mathbb{P}(\mathcal{G}_v^n \text{ is not a subgraph of } \mathbf{B}_{r_\epsilon}^n(v)) \leq \epsilon$  for any  $v \in [n]$ . Then for large enough  $n$ , after excluding irrelevant quantities then applying conditional Jensen's inequality, we have

$$\begin{aligned} & \mathbb{E} \left[ \left| \mathbb{P}(\tau(i) = h \mid b_i^n, \mathcal{F}_a) - \frac{1}{k} \right| \right] \\ & \leq \mathbb{P}(\mathcal{G}_i^n \text{ is not a subgraph of } \mathbf{B}_{r_\epsilon}^n(i)) \\ & \quad + \mathbb{E} \left[ \left| \mathbb{P}(\tau(i) = h \mid \mathcal{F}_a, \mathbf{B}_{r_\epsilon}^n(i)) - \frac{1}{k} \right| \mathbb{1}\{V(\lceil \delta n \rceil) \cap V_{r_\epsilon}^n(i) = \emptyset\} \right] \\ & \quad + \mathbb{P}(V(\lceil \delta n \rceil) \cap V_{r_\epsilon}^n(i) \neq \emptyset) \end{aligned} \quad (36)$$

On the event  $V(\lceil \delta n \rceil) \cap V_{r_\epsilon}^n(v) = \emptyset$ , for  $h_1, h_2 \in [k]$  with  $h_1 \neq h_2$ , we have

$$\frac{\mathbb{P}(\tau(i) = h_1 \mid \mathcal{F}_a, \mathbf{B}_{r_\epsilon}^n(i))}{\mathbb{P}(\tau(i) = h_2 \mid \mathcal{F}_a, \mathbf{B}_{r_\epsilon}^n(i))} = \frac{\mathbb{P}(\tau(i) = h_1 \mid \mathbf{B}_{r_\epsilon}^n(i), \{n_h : h \in [k]\}, \tau(V(\lceil \delta n \rceil)))}{\mathbb{P}(\tau(i) = h_2 \mid \mathbf{B}_{r_\epsilon}^n(i), \{n_h : h \in [k]\}, \tau(V(\lceil \delta n \rceil)))}$$

Therefore, the right hand side of equation (36) is equal to

$$\begin{aligned} & \mathbb{E} \left[ \left| \mathbb{P}(\tau(i) = h \mid \mathbf{B}_{r_\epsilon}^n(i), \{n_h : h \in [k]\}, \tau(V(\lceil \delta n \rceil))) - \frac{1}{k} \right| \mathbb{1}\{V(\lceil \delta n \rceil) \cap V_{r_\epsilon}^n(i) = \emptyset\} \right] \\ & \quad + \mathbb{P}(\mathcal{G}_i^n \text{ is not a subgraph of } \mathbf{B}_{r_\epsilon}^n(i)) + \mathbb{P}(V(\lceil \delta n \rceil) \cap V_{r_\epsilon}^n(i) \neq \emptyset). \end{aligned}$$

By Proposition 2 in [MNS15],

$$\lim_{\delta \rightarrow 0^+} \limsup_{n \rightarrow \infty} \mathbb{E} \left[ \left| \mathbb{P}(\tau(i) = h \mid \mathbf{B}_{r_\epsilon}^n(i), \{n_h : h \in [k]\}, \tau(V(\lceil \delta n \rceil))) - \frac{1}{k} \right| \right] = 0.$$

Furthermore, as  $\delta \rightarrow 0^+$ ,  $\limsup_{n \rightarrow \infty} \mathbb{P}(V(\lceil \delta n \rceil) \cap V_{r_\epsilon}^n(i) \neq \emptyset) \rightarrow 0$ , therefore,

$$\limsup_{\delta \rightarrow 0^+} \limsup_{n \rightarrow \infty} \mathbb{E} \left[ \left| \mathbb{P}(\tau(i) = h \mid \mathcal{F}_a, b_i^n) - \frac{1}{k} \right| \right] \leq \epsilon.$$

Since  $\epsilon$  is arbitrary, we conclude that equation (35) holds, this finishes the proof of this lemma.