

# On Local Distributed Sampling and Counting\*

Weiming Feng<sup>†</sup>      Yitong Yin<sup>‡</sup>

## Abstract

In classic distributed graph problems, each instance on a graph specifies a space of feasible solutions (e.g. all proper  $(\Delta + 1)$ -list-colorings of the graph), and the task of distributed algorithm is to construct a feasible solution using local information.

We study distributed sampling and counting problems, in which each instance specifies a joint distribution of feasible solutions. The task of distributed algorithm is to sample from this joint distribution, or to locally measure the volume of the probability space via the marginal probabilities. The latter task is also known as inference, which is a local counterpart of counting.

For self-reducible classes of instances, the following equivalences are established in the LOCAL model up to polylogarithmic factors:

- For all joint distributions, approximate inference and approximate sampling are computationally equivalent.
- For all joint distributions defined by local constraints, exact sampling is reducible to either one of the above tasks.
- If further, sequentially constructing a feasible solution is trivial locally, then all above tasks are easy if and only if the joint distribution exhibits strong spatial mixing.

Combining with the state of the arts of strong spatial mixing, we obtain efficient sampling algorithms in the LOCAL model for various important sampling problems, including: an  $O(\sqrt{\Delta} \log^3 n)$ -round algorithm for exact sampling matchings in graphs with maximum degree  $\Delta$ , and an  $O(\log^3 n)$ -round algorithm for sampling according to the hardcore model (weighted independent sets) in the uniqueness regime, which along with the  $\Omega(\text{diam})$  lower bound in [3] for sampling according to the hardcore model in the non-uniqueness regime, gives the first computational phase transition for distributed sampling.

---

\*This research is supported by the National Science Foundation of China under Grant No. 61672275 and No. 61722207.

<sup>†</sup>Department of Computer Science and Technology, Nanjing University. Email: fengwm@smail.nju.edu.cn.

<sup>‡</sup>State Key Laboratory for Novel Software Technology, Nanjing University. Email: yinyt@nju.edu.cn.

# 1 Introduction

In local computation, classic distributed graph problems are formulated in such a way that each instance  $\mathcal{I}$  on a graph  $G = (V, E)$  specifies a set  $\Omega_{\mathcal{I}}$  of feasible solutions  $\mathbf{y} = (y_v)_{v \in V}$ , and the goal of the distributed algorithm is to construct a feasible solution  $\mathbf{y} \in \Omega_{\mathcal{I}}$  by outputting  $y_v$  at each node  $v \in V$ . As a paradigm, we consider the list-coloring problem, where each instance  $\mathcal{I}$  gives a graph  $G = (V, E)$  with each node  $v \in V$  associated with a list  $L_v$  of available colors. Then the set  $\Omega_{\mathcal{I}}$  contains all proper list-colorings  $\mathbf{y}$  with  $y_v \in L_v$  for every  $v \in V$ , satisfying  $y_u \neq y_v$  for all edges  $uv \in E$ .

Alternatively, we can consider the set of feasible solutions  $\Omega_{\mathcal{I}}$  as the sample space of a probability distribution and imagine that each instance  $\mathcal{I}$  specifies a joint distribution  $\mu_{\mathcal{I}}$  of feasible solutions  $\mathbf{y} = (y_v)_{v \in V} \in \Omega_{\mathcal{I}}$ . A distributed graph problem is then given by a class of joint distributions  $\mu_{\mathcal{I}}$  indexed by instances  $\mathcal{I}$ . (In the paradigm of list-coloring problem, we may take each  $\mu_{\mathcal{I}}$  as the uniform distribution over all proper list-colorings of instance  $\mathcal{I}$ .) To each class of joint distributions of this form, there correspond a number of naturally defined problems.

- Construction: Exhibit a feasible solution  $\mathbf{y} = (y_v)_{v \in V}$  satisfying  $\mu_{\mathcal{I}}(\mathbf{y}) > 0$ , where each node  $v \in V$  outputs  $y_v$ . This is the task for classic distributed graph problems. (e.g. Construct a proper list-coloring of instance  $\mathcal{I}$ .)
- Sampling: Generate a random solution  $\mathbf{Y} = (Y_v)_{v \in V}$  distributed according to  $\mu_{\mathcal{I}}$ , where each node  $v \in V$  outputs  $Y_v$ . (e.g. Generate uniformly a proper list-coloring of instance  $\mathcal{I}$  at random.)
- Inference (Counting): Each node calculates the marginal probabilities of the random variable  $Y_v$  being specific values, where the random vector  $\mathbf{Y} = (Y_v)_{v \in V}$  is drawn from  $\mu_{\mathcal{I}}$ . (e.g. Each node  $v \in V$  calculates the probabilities of  $v$  being assigned specific colors in a uniform random proper list-coloring of instance  $\mathcal{I}$ .)

We choose the inference problem as a local version of counting, as the marginal probabilities are typical local knowledges regarding the volume of probability space. A more standard global definition of counting is to estimate the number of feasible solutions  $|\Omega_{\mathcal{I}}|$  (or the total weights if the joint distribution  $\mu_{\mathcal{I}}$  is non-uniform), which is unsuitable to study for local computation because it computes a global information. However, it is well known that for self-reducible problems, such global information can be decomposed via the chain rule into the marginal probabilities computed by inference problems [11]. Furthermore, the inference problem itself is especially well-motivated by distributed machine learning [17].

Previous studies in local computation were focused on the complexity of constructing a feasible solution. The studies of sampling problems in local computation were started very recently [3, 4]. Several fundamental questions regarding the local complexities of sampling and counting need to be answered.

**Question 1:** *What is the relation between sampling and counting in local computation?*

It is well known that for self-reducible problems, approximate counting and approximate sampling are inter-reducible on polynomial-time Turing machines [10]. A natural question is whether this is true for local computation. To see the nontriviality of the question, recall that the generic reduction from sampling to counting has to be sequential because the procedure is fully-adaptive: Each individual variable is sampled according to the marginal distribution conditioning on the outcomes of previous samplings. To understand the relation between counting and sampling in local computation, one has to answer the following question first:

“How much non-adaptively can we sample a random vector  $\mathbf{Y} = (Y_i)$  by accessing to marginal distributions of individual variables  $Y_i$ ?”

which by itself is a fundamental question with a broader background.

**Question 2:** *What are the roles of approximation in sampling and counting in local computation?*

Due to the pioneering works of Valiant [22] and Stockmeyer [21], on Turing machines exact counting can be much harder than approximate counting. Similar phenomena occur for local computation: Due to a straightforward information-theoretical argument, exact inference is impossible to compute locally unless for joint distributions with zero long-range correlation.

For sampling, a celebrated result of Jerrum, Valiant, and Vazirani [10] shows that on polynomial-time Turing machines, for self-reducible problems, approximate sampling can be boosted into exact sampling via a rejection sampling procedure (the JVV sampler), such that the algorithm succeeds with high probability (with certifiable failures) and conditioning on success the output is distributed precisely according to the joint distribution  $\mu_{\mathcal{I}}$  specified by the instance  $\mathcal{I}$ . Our question is whether there is a distributed variant of the JVV sampler. Answering this question also involves investigating a fundamental problem: “Can rejection sampling be made local?” Considering that the rejection sampling as a basic Monte Carlo method has been studied for more than half a century, this is certainly worth studying in local computation.

**Question 3:** *What makes a sampling or counting problem solvable by local computations?*

Finally, we want to characterize the easiness of sampling and counting in local computation by the properties of joint distributions. For sampling and counting on polynomial-time Turing machines, a phase transition of computational complexity is witnessed at the threshold for the decay of correlation (strong spatial mixing) [23, 19]. We wonder whether similar computational phase transitions exist for local computation.

## 1.1 Our results

We study the local complexities of self-reducible sampling and counting problems, and provide answers to above fundamental questions.

We formulate distributed sampling and inference (counting) problems by classes of joint distributions of solutions. Besides general joint distributions, we focus on the joint distributions with following properties naturally arising from local computation:

- ( $\star$ ) Joint distributions defined by local constraints. These are the counterparts of locally checkable labelings (LCL) in the world of sampling and counting. In our paper, these joint distributions are called local Gibbs distributions (Definition 2.3 and 2.4).
- ( $\star\star$ ) Joint distributions for which constructing a feasible solution is trivial for a sequential local oblivious procedure. For example, this includes distributions over  $(\Delta + 1)$ -list-colorings, but not the distributions over  $\Delta$ -list-colorings. The property is related to the ergodicity of the local dynamics on feasible solutions. In our paper, joint distributions of this property are called locally admissible (Definition 2.5).

**Main results:** For self-reducible classes of instances, we show the followings hold for sampling and inference (counting) in the LOCAL model:

- Approximate inference and approximate sampling are inter-reducible, in a sense that if one of the tasks is tractable in the LOCAL model so is the other one (Theorem 3.2 and Theorem 3.4), where an approximate problem is tractable in the LOCAL model if the problem is solvable for any  $n$  and  $\delta > 0$ , where  $n$  is the number of nodes and  $\delta$  is the approximation error, within time complexity  $\text{poly}(\log n, \log \frac{1}{\delta})$ .
- If property ( $\star$ ) is satisfied, then approximate inference, approximate sampling, and exact sampling are all inter-reducible. With the above results, this is proved by a distributed JVV sampler that successfully terminates within  $\text{polylog}(n)$  rounds with high probability, and conditioning on success, returns a solution distributed precisely as the desired distribution (Theorem 4.2).
- If further, property ( $\star\star$ ) is satisfied, then all above tasks are easy if and only if the joint distribution exhibits certain degrees of strong spatial mixing (Theorem 5.1 and Corollary 5.3), a decay of correlation property which is critically related to the computational complexity of approximate counting [23, 19].

**Minor results:** As by-products of above results, we obtain two boosting lemmas:

- If property ( $\star$ ) is satisfied, then approximate inference with bounded additive (total variation) error can be boosted to with bounded multiplicative error (Lemma 4.1).
- If further, property ( $\star\star$ ) is satisfied, then strong spatial mixing in total variation distance implies strong spatial mixing with decay in multiplicative error (Corollary 5.2).

This explains why so far the known strong spatial mixing results for several major problems (e.g. independent sets, matchings, and graph colorings) were all proved in this stronger form with decay in multiplicative error [23, 2, 5, 6].

Interestingly, the second boosting result states a proposition in probability theory, which seems unrelated to distributed algorithms, but is proved by us via local computation.

**Applications in the LOCAL model:** The main results in above together with the state of the arts of strong spatial mixing [2, 23, 6, 12, 20], imply the following exact sampling algorithms for various important sampling problems and statistical physics models:

- An  $O(\sqrt{\Delta} \log^3 n)$ -round algorithm for exact sampling matchings in graphs with maximum degree  $\Delta$ .
- An  $O(\log^3 n)$ -round exact sampling according to the hardcore model (weighted independent sets) in the uniqueness regime, which along with the  $\Omega(\text{diam})$  lower bound in a previous work [3] for sampling according to the hardcore model in the non-uniqueness regime, gives the first *computational phase transition* for distributed sampling and counting.
- $O(\log^3 n)$ -round exact sampling algorithms for sampling according to various anti-ferromagnetic models, including: anti-ferromagnetic 2-spin model in the uniqueness regime, weighted hypergraph matchings in the uniqueness regime, proper  $q$ -colorings of triangle-free graphs when  $q \geq \alpha\Delta$  where  $\alpha > \alpha^*$  and  $\alpha^* \approx 1.763$  is the positive root of the equation  $x = e^{1/x}$ .

## 1.2 Organization of the paper

Models and definitions are introduced in Section 2. In Section 3, we apply a generic transformation from local sequential sampling to local distributed sampling, which proves the first main result. In Section 4, we develop the techniques of local self-reductions, which proves the second main result, giving the distributed JVV sampler, along with the boosting lemma for approximate inference. In Section 5, we explore the intrinsic relation between decay of correlations and distributed counting and sampling, proving the third main result.

## 2 Models and Definitions

**Notation for Graphs:** Let  $G = (V, E)$  be a simple, undirected graph. For any two vertices  $u, v \in V$ , the distance between  $u$  and  $v$  in  $G$ , denoted by  $\text{dist}_G(u, v)$ , is the length of the shortest path between  $u$  and  $v$  in  $G$ ; and for a subset  $S \subseteq V$ , we define  $\text{dist}_G(v, S) \triangleq \min_{u \in S} \text{dist}_G(u, v)$ . For any vertex  $v \in V$  and  $r > 0$ , let  $B_r(v) \triangleq \{u \in V \mid \text{dist}_G(u, v) \leq r\}$  denote the  $r$ -ball centered at  $v$  in  $G$ .

**Notation for Joint Distributions:** Let  $V$  be a set of size  $n = |V|$  and  $\Sigma$  an *alphabet* of size  $q = |\Sigma| \leq \text{poly}(n)$ . Let  $\Omega = \Sigma^V$  be the *sample space*. Each  $\sigma \in \Sigma^V$  is called a *configuration*. For  $S \subseteq V$ , we use  $\sigma_S$  or  $\sigma(S)$  to denote the restriction of  $\sigma$  on subset  $S$ .

Let  $\mu$  be a distribution over  $\Sigma^V$ , called a *joint distribution*, because each  $Y \sim \mu$  is a random vector  $Y = (Y_v)_{v \in V}$  consisting of  $n$  jointly distributed random variables. For  $R \subseteq V$ , let  $\mu_R$  denote the *marginal distribution* over  $\Sigma^R$  induced by  $\mu$  on subset  $R$ , formally:

$$\forall \tau \in \Sigma^R : \quad \mu_R(\tau) = \Pr_{Y \sim \mu} [Y_R = \tau] = \sum_{\sigma \in \Sigma^V : \sigma_R = \tau} \mu(\sigma).$$

In particular, when  $R = \{v\}$  for some  $v \in V$ , we write  $\mu_v = \mu_{\{v\}}$ . A configuration  $\sigma \in \Sigma^V$  is *feasible* with respect to  $\mu$  if  $\mu(\sigma) > 0$ . A configuration  $\tau \in \Sigma^\Lambda$  on a subset  $\Lambda \subseteq V$  is feasible with respect to  $\mu$  if  $\mu_\Lambda(\tau) > 0$ , i.e. if there is a feasible  $\sigma \in \Sigma^V$  such that  $\sigma_\Lambda = \tau$ . By convention an empty configuration  $\emptyset$  is always feasible. Given a feasible configuration  $\tau \in \Sigma^\Lambda$  on a subset  $\Lambda \subseteq V$ , we use  $\mu^\tau$  to denote the distribution over  $\Sigma^V$  induced by  $\mu$  conditioning on  $\tau$ , formally:

$$\forall \sigma \in \Sigma^V : \quad \mu^\tau(\sigma) = \Pr_{Y \sim \mu} [Y = \sigma \mid Y_\Lambda = \tau].$$

The conditional marginal distributions  $\mu_R^\tau$  and  $\mu_v^\tau$  are accordingly defined.

Suppose that  $\mu$  and  $\nu$  are two distributions over the same sample space  $\Omega$ . The *total variation distance* between  $\mu$  and  $\nu$ , denoted by  $d_{\text{TV}}(\mu, \nu)$ , is defined as:

$$d_{\text{TV}}(\mu, \nu) \triangleq \frac{1}{2} \|\mu - \nu\|_1 = \max_{A \subseteq \Omega} |\mu(A) - \nu(A)|.$$

**Distributed Graph Problems:** We reformulate the notion of *distributed graph problems* in [8] by classes of joint distributions.

**Definition 2.1** (distributed graph problems). A distributed graph problem is given by a class of joint distributions  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$ , indexed by labeled graphs  $(G, \mathbf{x})$ , where  $G = (V, E)$  is a simple, undirected graph and  $\mathbf{x} = (x_v)_{v \in V}$  is a  $|V|$ -dimensional vector. Each  $\mu_{(G, \mathbf{x})}$  is a joint distribution over  $\Sigma^V$ , where  $\Sigma = \Sigma_{(G, \mathbf{x})}$  is an alphabet of size  $q = |\Sigma|$  bounded in polynomial of  $n = |V|$ .

The class of distributions  $\mathfrak{M}$  is *translation-invariant*, which means that if the labeled graphs  $(G, \mathbf{x})$  and  $(\tilde{G}, \tilde{\mathbf{x}})$  are isomorphic under bijection  $\phi$  on vertices, then  $\mu_{(G, \mathbf{x})} \in \mathfrak{M}$  is well-defined if and only if  $\mu_{(\tilde{G}, \tilde{\mathbf{x}})}$  is, and the distributions  $\mu_{(G, \mathbf{x})}$  and  $\mu_{(\tilde{G}, \tilde{\mathbf{x}})}$  are identical under bijection  $\phi$ .

Initially, each node  $v \in V$  knows  $x_v$ . For classic distributed graph problems, the nodes need to construct a feasible  $\mathbf{y}$  that  $\mu_{(G, \mathbf{x})}(\mathbf{y}) > 0$ , where each node  $v \in V$  outputs  $y_v$ .

**Remark 2.1.** In [8], a distributed graph problem is defined by a relation  $\mathcal{T}$  that contains all satisfying tuples  $(G, \mathbf{x}, \mathbf{y})$  instead of as a class of distributions  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$  over  $\mathbf{y}$ . The two definitions are equivalent: Given a relation  $\mathcal{T}$ , for any  $(G, \mathbf{x})$ ,  $\mu_{(G, \mathbf{x})}$  is just a positive distribution over all feasible solutions  $\mathbf{y}$  satisfying  $(G, \mathbf{x}, \mathbf{y}) \in \mathcal{T}$  (e.g. the uniform distribution over feasible solutions); and conversely, given  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$ , the relation  $\mathcal{T}$  can be constructed as  $\mathcal{T} = \{(G, \mathbf{x}, \mathbf{y}) \mid \mu_{(G, \mathbf{x})}(\mathbf{y}) > 0\}$ .

**Distributed Sampling and Counting:** A distributed sampling or counting problem is also given by a class of joint distributions  $\mathfrak{M} = \{\mu_{(G,\mathbf{x})}\}$  as defined in Definition 2.1.

**Definition 2.2** (instances for distributed sampling/counting). Let  $\mathfrak{M} = \{\mu_{(G,\mathbf{x})}\}$  be a class of joint distributions as defined in Definition 2.1. An instance for distributed sampling/counting is a tuple  $(G, \mathbf{x}, \tau)$ , where  $(G, \mathbf{x})$  specifies a joint distribution  $\mu = \mu_{(G,\mathbf{x})}$  over  $\Sigma^V$ , and  $\tau \in \Sigma^\Lambda$  is an arbitrary configuration on a subset  $\Lambda \subseteq V$  that is feasible with respect to  $\mu$ . We call the distribution  $\mu^\tau$  the *target distribution*.

Given an instance  $(G, \mathbf{x}, \tau)$  where  $\tau$  is specified on the subset  $\Lambda \subseteq V$ , initially each node  $v \in V$  knows  $x_v$ , and also  $\tau_v$  if  $v \in \Lambda$ . We assume that  $x_v$  includes a unique ID for  $v$ , a global polynomial upper bound of  $n = |V|$ , and a global upper bound on errors if approximation is involved.

**Remark 2.2.** The reason to include an arbitrary partially specified configuration  $\tau$  into the problem instance, is to explicitly enforce the *self-reducibility*, a property that is essential to problems such as sampling and counting [10]. In our context, it means that arbitrarily fixing a feasible evaluation  $\tau$  of a subset of variables, the conditional distribution  $\mu^\tau$  forms a valid instance over the remaining free variables.<sup>1</sup>

For example, consider  $\mu$  as the uniform distribution over all proper  $(\Delta + 1)$ -colorings of  $G$ . For any proper  $(\Delta + 1)$ -coloring  $\tau$  of vertices in a subset  $\Lambda \subseteq V$ ,  $\mu^\tau$  is the uniform distribution over all proper  $(\Delta + 1)$ -colorings of  $G$  consistent with  $\tau$ . This equivalently specifies a uniform distribution over list-colorings of the subgraph  $G[V \setminus \Lambda]$  induced by subset  $V \setminus \Lambda$ , where each node  $v \in V$  holds a list of available colors  $L_v = [q] \setminus \{\tau_u \mid uv \in E\}$ .

We assume that the time complexity of a distributed algorithm is fixed. Upon termination the algorithm either successfully returns or fails. We assume that the algorithm succeeds with high probability, and all failures are *locally certifiable*. Upon termination, each node  $v \in V$  besides the regular output explicitly outputs a random bit  $F_v$  indicating whether the algorithm fails locally at  $v$ , and it is guaranteed that  $\sum_{v \in V} \mathbb{E}[F_v] = O(\frac{1}{n})$ . This is a well accepted notion of the Las Vegas algorithms for local computation [7].

The goal of *distributed sampling* is to draw a random sample according to the target distribution  $\mu^\tau$  given by the instance  $(G, \mathbf{x}, \tau)$ .

**Exact sampling:** Given any instance  $(G, \mathbf{x}, \tau)$ , the nodes need to sample a random  $\mathbf{Y} = (Y_v)_{v \in V}$  upon successful termination, where each node  $v \in V$  outputs  $Y_v$  or fails, such that conditioning on that no one fails the distribution of  $\mathbf{Y}$  is precisely  $\mu^\tau$ .

---

<sup>1</sup>Alternatively, one may enforce the self-reducibility implicitly by assuming it as a property of the class of joint distributions  $\mathfrak{M} = \{\mu_{(G,\mathbf{x})}\}$ : For every  $\mu_{(G,\mathbf{x})} \in \mathfrak{M}$  where  $G = (V, E)$ , for any feasible configuration  $\tau \in \Sigma^\Lambda$  on a subset  $\Lambda \subseteq V$ , there exists a  $\mu_{(G',\mathbf{x}')} \in \mathfrak{M}$  such that  $G' = (V, E')$  is a subgraph of  $G$  with identical vertex set and  $\mu_{(G,\mathbf{x})}^\tau = \mu_{(G',\mathbf{x}')}$ , where the new instance  $(G', \mathbf{x}')$  can be constructed locally from  $(G, \mathbf{x})$ , providing  $\tau_v$  to each node  $v \in \Lambda$ . This alternative formulation of self-reducibility is equivalent to the one we used above.

**Approximate sampling:** Given any instance  $(G, \mathbf{x}, \tau)$ , for any  $\delta > 0$ , the nodes need to output a random  $\mathbf{Y} = (Y_v)_{v \in V}$  upon successful termination, such that conditioning on success the distribution  $\hat{\mu}$  of  $\mathbf{Y}$  satisfies  $d_{\text{TV}}(\hat{\mu}, \mu^\tau) \leq \delta$ .

We use the *distributed inference* to represent counting in distributed settings. Here, the goal is to estimate the marginal distribution  $\mu_v^\tau$  for each node  $v \in V$ , where  $\mu_v^\tau$  is the target distribution given by the instance  $(G, \mathbf{x}, \tau)$ . Due to an information-theoretical argument, exact inference with local information is impossible for joint distributions with nonzero long-range correlations. Hence we focus on approximate inference.

**Approximate inference:** Given any instance  $(G, \mathbf{x}, \tau)$ , for any  $\delta > 0$ , each node  $v \in V$  needs to output a marginal distribution  $\hat{\mu}_v$  over  $\Sigma$ , which is a vector  $\hat{\mu}_v \in [0, 1]^\Sigma$  and  $\|\hat{\mu}_v\|_1 = 1$ , satisfying that  $d_{\text{TV}}(\hat{\mu}_v, \mu_v^\tau) \leq \delta$ .

A more accurate approximate inference with bounded multiplicative error is discussed in Section 4.1.

**Joint Distributions Defined by Local Constraints:** We use the Gibbs distributions defined by the *weighted constraint satisfaction problems (CSPs)*, also known as the *factor graphs* [14], to model joint distributions characterized by local constraints.

**Definition 2.3** (Gibbs distributions). A *Gibbs distribution*  $\mu$  is specified by a tuple  $(G, \Sigma, \mathcal{F})$ , where  $G = (V, E)$  is an undirected graph,  $\Sigma$  is an alphabet of size  $q = |\Sigma|$  bounded in polynomial of  $n = |V|$ , and  $\mathcal{F}$  is a collection of *constraints* (also called *factors*). A constraint  $(f, S) \in \mathcal{F}$  consists of a nonnegative-valued function  $f : \Sigma^S \rightarrow \mathbb{R}_{\geq 0}$  defined on the *scope*  $S \subseteq V$ . A constraint  $(f, S)$  is a *soft constraint* if the function  $f$  is positive-valued, otherwise it is a *hard constraint*.

Each configuration  $\sigma \in \Sigma^V$  is assigned a weight:

$$w(\sigma) = \prod_{(f,S) \in \mathcal{F}} f(\sigma_S). \quad (1)$$

The Gibbs distribution  $\mu$  over all configurations in  $\Sigma^V$  is defined proportional to the weights

$$\mu(\sigma) = \frac{w(\sigma)}{Z},$$

where the normalizing factor  $Z = \sum_{\sigma \in \Sigma^V} w(\sigma)$  is known as the *partition function*, which can be seen as a function of the specification  $(G, \Sigma, \mathcal{F})$  of the distribution  $\mu$ .

In particular, when all constraint functions  $f$  are Boolean-valued functions, the distribution  $\mu$  is the uniform distribution over all feasible configurations, and  $Z$  gives the total number of feasible configurations.



**Definition 2.4** (locality of Gibbs distributions). A class of Gibbs distributions specified by  $(G, \Sigma, \mathcal{F})$  are *local* if for any constraint  $(f, S) \in \mathcal{F}$ , the diameter of the scope  $S$  in graph  $G$  is bounded by a constant, that is,  $\max_{u,v \in S} \text{dist}_G(u, v) = O(1)$ .

The local Gibbs distributions are the counterparts of the LCL problems [15] in the world of distributed sampling/counting. Just as that the LCL problems are the distributed graph problems that are defined by local constraints, the local Gibbs distributions are the joint distributions that are defined by local factors.

When an instance  $(G, \mathbf{x}, \tau)$  is provided to distributed algorithm, if the joint distribution  $\mu_{(G, \mathbf{x})}$  is a local Gibbs distribution specified by  $(G, \Sigma, \mathcal{F})$ , we assume that for each node  $v \in V$ ,  $x_v$  includes the descriptions of all local constraints  $(f, S) \in \mathcal{F}$  that  $v \in S$ .

An important property of Gibbs distribution is the *spatial Markovian* property, also known as conditional independence, which is stated formally by the following proposition.

**Proposition 2.1** (conditional independence). *Let  $\mu$  be a Gibbs distribution specified by  $(G, \Sigma, \mathcal{F})$ , where  $G = (V, E)$ . Let  $H = (V, F)$  denote the hypergraph with vertices  $V$  and hyperedges  $F = \{S \mid (f, S) \in \mathcal{F}\}$ . Suppose that  $A, B, C \subset V$  are disjoint nonempty subsets and  $C$  is a vertex separator whose removal disconnects  $A$  and  $B$  in  $H$ . For a random vector  $Y \in \Sigma^V$  distributed according to  $\mu$ ,  $Y_A$  and  $Y_B$  are conditionally independent given that  $Y_C$  is arbitrarily and feasibly fixed. Formally, for any  $\sigma_C \in \Sigma^C$  that is feasible with respect to  $\mu$ , any  $\sigma_A \in \Sigma^A$  and  $\sigma_B \in \Sigma^B$ , it holds that*

$$\Pr_{Y \sim \mu} [Y_A = \sigma_A \wedge Y_B = \sigma_B \mid Y_C = \sigma_C] = \Pr_{Y \sim \mu} [Y_A = \sigma_A \mid Y_C = \sigma_C] \Pr_{Y \sim \mu} [Y_B = \sigma_B \mid Y_C = \sigma_C].$$

We also consider a restrictive class of Gibbs distributions, with the following property.

**Definition 2.5** (locally admissible). Let  $\mu$  be a Gibbs distribution specified by  $(G, \Sigma, \mathcal{F})$  where  $G = (V, E)$ . Let  $\Lambda \subseteq V$  be a subset. A configuration  $\sigma \in \Sigma^\Lambda$  on a subset  $\Lambda \subseteq V$  is *locally feasible* if  $\sigma$  itself does not violate any constraint onsite, formally:

$$\prod_{\substack{(f, S) \in \mathcal{F} \\ S \subseteq \Lambda}} f(\sigma_S) > 0.$$

Recall that a  $\sigma \in \Sigma^\Lambda$  is feasible if there is a  $\tau \in \Sigma^V$  such that  $\mu(\tau) > 0$  and  $\tau_\Lambda = \sigma$ . Clearly  $\sigma$  is locally feasible if it is feasible. A Gibbs distribution  $\mu$  is said to be *locally admissible* if the converse is also true:

$$\forall \Lambda \subseteq V, \forall \sigma \in \Sigma^\Lambda : \quad \sigma \text{ is feasible} \iff \sigma \text{ is locally feasible.}$$

**Remark 2.3.** The locally admissible, local Gibbs distributions represent the LCL problems that can be solved by sequential local oblivious algorithms. For any class of locally admissible, local Gibbs distributions  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$ , the problem of constructing a  $\mathbf{y}$  that is feasible with respect to  $\mu_{(G, \mathbf{x})}$ , can always be solved by a sequential local oblivious algorithm on any vertex ordering, i.e. the problem is in  $\text{SLOCAL}(O(1))$ .

**The LOCAL Model:** In the LOCAL model [15, 18], the network is a simple, undirected graph  $G = (V, E)$ . Initially, each node  $v \in V$  receives a local input and an arbitrary long random bit string sampled independently at  $v$ . For a LOCAL algorithm with time complexity  $t$ , each node  $v \in V$  gathers all information within radius  $t$  from  $v$ , including the topology of the graph, the inputs and random bits of the nodes within that radius, and performs an arbitrary local computation with the information to compute an output.

### 3 From Sequential to Parallel for Distributed Sampling

In this section, we establish the computational equivalence (up to polylogarithmic factor) between approximate inference and approximate sampling in the LOCAL model. The results in this section hold for general classes of joint distributions  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$ .

A key step is to first resolve the problems sequentially with bounded locality. The sequential local mode (SLOCAL) is introduced in a recent breakthrough [8]. We adopt the randomized version. Let  $G = (V, E)$  be a simple, undirected graph with  $n = |V|$ . Each node  $v \in V$  maintains a local state  $S_v$  in its unbounded local memory. Initially,  $S_v$  contains  $v$ 's local input and an arbitrarily long random bit string generated by  $v$ . An SLOCAL algorithm  $\mathcal{A}$  with locality  $r(n)$  scans the nodes in an arbitrary ordering  $\pi = (v_1, v_2, \dots, v_n)$  provided by an adversary. When processing node  $v$ ,  $\mathcal{A}$  reads the states  $S_u$  for all  $u \in B_{r_v}(v)$  with  $r_v \leq r(n)$ , then performs unbounded computation to update the state  $S_v$  and compute the output  $Y_v$  at  $v$ .

The standard SLOCAL algorithm scans all the vertices in one pass. One important generalization is to allow the algorithm scan all the vertices in  $k$  passes. This generalized algorithm is called the  $k$ -pass SLOCAL algorithm.

An important property of the SLOCAL model is that any SLOCAL algorithm with locality  $r$  can be transformed into a LOCAL algorithm with time complexity  $O(r)$  multiplying the cost for network decomposition [16]. We restate this in the following lemma in a slightly more refined way, in order to cover more general problems such as distributed sampling.

**Lemma 3.1** (Ghaffari, Kuhn, Maus [8]). *Let  $\mathcal{A}$  be a  $k$ -pass SLOCAL algorithm. Given any instance  $\mathcal{I}$  on graph  $G = (V, E)$  with  $n = |V|$ , any ordering  $\pi$  of nodes in  $V$ ,  $\mathcal{A}$  scans all nodes in the same ordering  $\pi$  by  $k$  passes with locality  $r_i(n)$  in the  $i$ -th pass.  $\mathcal{A}$  returns a random vector  $Y = (Y_v)_{v \in V}$ , where each node  $v \in V$  outputs  $Y_v$ , such that  $Y$  follows the distribution  $\hat{\mu}_{\mathcal{I}, \pi}$ .*

*Then there is a LOCAL algorithm  $\mathcal{B}$  which given any instance  $\mathcal{I}$  on graph  $G = (V, E)$  with  $n = |V|$ , outputs within time complexity  $O(kr(n) \log^2 n)$  a pair  $(Y_v, F_v)$  at each node  $v \in V$ , where  $r(n) = \max_{1 \leq i \leq k} r_i(n)$  and  $F_v$  is a Boolean random variable indicating whether the algorithm fails locally at  $v$ , satisfying that  $\sum_{v \in V} \mathbb{E}[F_v] < \frac{1}{n^2}$  and conditioning on that  $F_v = 0$  for all nodes  $v \in V$ , the distribution of  $Y = (Y_v)_{v \in V}$  is precisely  $\hat{\mu}_{\mathcal{I}, \pi}$  for some ordering  $\pi$  of nodes in  $V$ .*

The proof of Lemma 3.1 is based on the *network decomposition*.

**Definition 3.1** (network decomposition [1]). A  $(C, D)$ -network decomposition of a graph  $G = (V, E)$  is a partition of  $V$  into vertex-disjoint clusters, with each cluster assigned a color from  $\{1, 2, \dots, C\}$ . The diameter of each cluster in graph  $G$  is at most  $D$ . The graph derived by contracting clusters is properly colored.

*Proof of Lemma 3.1.* Lemma 3.1 is proved in the same way as Theorem 1.11 in [8]. Specifically, the LOCAL algorithm  $\mathcal{B}$  first computes an  $(O(\log n), O(\log n))$ -network decomposition [13] on power graph  $G^{r(n)+1}$ , where every pair of  $u$  and  $v$  that  $\text{dist}_G(u, v) \leq r(n) + 1$  are connected by an edge in  $G^{r(n)+1}$ . Let  $q_v$  denote the color of  $v$ 's cluster after the network decomposition. We assign each node  $v \in V$  a label  $(q_v, \text{ID}_v)$ , where  $\text{ID}_v$  is  $v$ 's unique id. Let  $\pi$  be the lexicographically increasing order of the node labels  $(q_v, \text{ID}_v)$ . Then  $\mathcal{B}$  simulates  $\mathcal{A}$  with  $k$  epochs, each having  $O(r(n) \log^2 n)$  rounds. In the  $i$ -th epoch,  $\mathcal{B}$  simulates the  $i$ -th pass of  $\mathcal{A}$  according to the ordering  $\pi$ .

One slight modification is to note that all failures caused by network decomposition [13] are locally certifiable and the output distribution  $\hat{\mu}_{\mathcal{I}, \pi}$  is preserved conditioning on success.  $\square$

### 3.1 Approximate Inference $\implies$ Approximate Sampling

**Theorem 3.2.** *For any class of joint distributions  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$ , if there is a LOCAL algorithm for approximate inference (within arbitrary total variation error  $\delta > 0$ ) with time complexity at most  $t(n, \delta)$ , then there is a LOCAL algorithm for approximate sampling (within arbitrary total variation error  $\delta > 0$ ) with time complexity  $O(t(n, \frac{\delta}{n}) \log^2 n)$ .*

First, observe that for the inference problems, any randomized LOCAL algorithm with certifiable local failures can be transformed into a deterministic LOCAL algorithm with no failure by taking average over all random bits accessed by the randomized LOCAL algorithm at each node  $v$  that produce the successful output at  $v$ . This actually holds more generally for all problems with *no symmetry*, where the correctness of the output of a node  $v$  depends only on the instance but not on other nodes' outputs.

**Proposition 3.3.** *For any class of joint distributions  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$ , if there is a randomized LOCAL algorithm for approximate inference with time complexity at most  $t(n, \delta)$  and failure probability  $\sum_{v \in V} \mathbb{E}[F_v] < 1$ , then there is a deterministic LOCAL algorithm for approximate inference with time complexity at most  $t(n, \delta)$ .*

With this proposition, from now on we assume without loss of generality any LOCAL algorithm for approximate inference is deterministic and has no failure.

*Proof of Theorem 3.2.* With Lemma 3.1, it is sufficient to give an SLOCAL algorithm for approximate sampling, which is quite standard with the access to marginal probabilities.

Let  $(G, \mathbf{x}, \tau)$  be an instance, where  $G = (V, E)$  and  $\tau \in \Sigma^\Lambda$  is a feasible configuration on a subset  $\Lambda \subseteq V$ . A joint distribution  $\mu = \mu_{(G, \mathbf{x})}$  is specified by the labeled graph  $(G, \mathbf{x})$  and the target distribution is  $\mu^\tau = \mu_{(G, \mathbf{x})}$ .

Let  $v_1, v_2, \dots, v_n$  be an arbitrary ordering of vertices in  $V$ . The algorithm will sample a random  $\sigma \in \Sigma^V$  such that  $\sigma_\Lambda = \tau$ , by randomly generating  $\sigma(v_i)$  vertex by vertex. For  $i = 1, 2, \dots, n$ , let  $\sigma_{i-1}$  denote the configuration over  $\{v_1, v_2, \dots, v_{i-1}\}$  that has been randomly generated so far, if  $v_i \in \Lambda$  then  $\sigma(v_i) = \tau(v_i)$ ; and if otherwise,  $\sigma(v_i)$  is randomly generated according to the marginal distribution  $\hat{\mu}_{v_i}^{\tau \wedge \sigma_{i-1}}$  at  $v_i$  conditioning on  $\sigma_{i-1}$  and  $\tau$ , where the marginal distribution  $\hat{\mu}_{v_i}^{\tau \wedge \sigma_{i-1}}$  is computed by simulating the LOCAL algorithm for approximate inference with total variation error  $\frac{\delta}{n}$  on instance  $(G, \mathbf{x}, \tau \wedge \sigma_{i-1})$  at node  $v_i$  with locality  $t = t(n, \frac{\delta}{n})$ . It is guaranteed that

$$d_{\text{TV}} \left( \hat{\mu}_{v_i}^{\tau \wedge \sigma_{i-1}}, \mu_{v_i}^{\tau \wedge \sigma_{i-1}} \right) \leq \frac{\delta}{n}.$$

Finally,  $\sigma = \sigma_n \in \Sigma^V$  is returned. Let  $\hat{\mu}$  denote its distribution. By a coupling argument, it can be verified that  $d_{\text{TV}}(\hat{\mu}, \mu^\tau) \leq \delta$ .  $\square$

### 3.2 Approximate Sampling $\implies$ Approximate Inference

**Theorem 3.4.** *For any class of joint distributions  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$ , if there is a LOCAL algorithm for approximate sampling (within arbitrary total variation error  $\delta > 0$ ) with time complexity at most  $t(n, \delta)$ , then there is a LOCAL algorithm for approximate inference (within total variation error  $\delta + \epsilon_0$ ) with time complexity at most  $t(n, \delta)$ , where  $\epsilon_0 \geq \sum_{v \in V} \mathbb{E}[F_v]$  is the probability that the approximate sampling algorithm fails.*

*Proof.* Let  $\mathcal{A}_\delta$  be the LOCAL algorithm for approximate sampling with total variation error  $\delta$  and time complexity at most  $t(n, \delta)$ . Given any instance  $(G, \mathbf{x}, \tau)$ , upon termination the algorithm  $\mathcal{A}_\delta$  outputs a random  $(y_v, F_v)$  at each node  $v \in V$ , where  $F_v$  is a Boolean random variable indicating the local failure at node  $v$ . It is guaranteed that  $\sum_{v \in V} \mathbb{E}[F_v] \leq \epsilon_0$  and conditioning on  $\sum_{v \in V} F_v = 0$ , the random vector  $\mathbf{y} = (y_v)_{v \in V}$  follows a distribution  $\hat{\mu}$  such that  $d_{\text{TV}}(\hat{\mu}, \mu^\tau) \leq \delta$  where  $\mu^\tau$  is the target distribution. Therefore, for the distribution  $\tilde{\mu}$  of the random vector  $\mathbf{y}$  (without any condition), it holds that

$$d_{\text{TV}}(\tilde{\mu}_v, \mu_v^\tau) \leq d_{\text{TV}}(\tilde{\mu}, \mu^\tau) \leq d_{\text{TV}}(\hat{\mu}, \mu^\tau) + \Pr \left[ \sum_{v \in V} F_v > 0 \right] \leq \delta + \epsilon_0.$$

The marginal distribution  $\tilde{\mu}_v$  can be reconstructed at node  $v$  with locality at most  $t = t(n, \delta)$  by enumerating the random bits used in  $\mathcal{A}_\delta$  to generate the random variable  $y_v$ . This gives a LOCAL algorithm for approximate inference within total variation error  $\delta + \epsilon_0$  and time complexity  $t(n, \delta)$ .  $\square$

## 4 Local Self-Reductions

It is well-known that on Turing machines, for self-reducible problems the accuracy of approximate counting can be boosted and approximate counting implies exact sampling (the Jerrum-Valiant-Vazirani sampler) [10].

In this section, we give two boosting results for local distributed counting and sampling:

- a *local boosting* for approximate inference which transforms approximate inference with bounded total variation error to the one with bounded multiplicative error;
- a *distributed JVV sampler* which uses approximate inference to achieve exact sampling via the *local rejection sampling*.

Both results utilizes *local self-reductions*, the self-reductions with bounded locality. The correctness of such reductions relies on the spatial Markovian (conditional independence) property of local Gibbs distributions.

## 4.1 The Boosting Lemma

We consider approximate inference with a stronger accuracy guarantee. The *multiplicative error function*  $\text{err}(\cdot, \cdot)$  is defined as follows: for any two distributions  $\mu$  and  $\hat{\mu}$  over the same sample space  $\Sigma$ ,

$$\text{err}(\mu, \hat{\mu}) \triangleq \max_{x \in \Sigma} |\ln \mu(x) - \ln \hat{\mu}(x)|, \quad (2)$$

with the convention that  $0/0 = 1$  and  $\ln 0 - \ln 0 = \ln(0/0) = 0$ .

**Approximate inference (with multiplicative error  $\epsilon$ ):** For any instance  $(G, \mathbf{x}, \tau)$ , any  $0 < \epsilon < 1$ , each node  $v$  returns a marginal distribution  $\hat{\mu}_v$ , such that  $\text{err}(\hat{\mu}_v, \mu_v^\tau) \leq \epsilon$ .

For sufficiently small  $\epsilon > 0$ , the condition  $\text{err}(\hat{\mu}_v, \mu_v^\tau) \leq \epsilon$  implies

$$\forall c \in \Sigma: \quad 1 - \epsilon \approx e^{-\epsilon} \leq \frac{\hat{\mu}_v(c)}{\mu_v^\tau(c)} \leq e^\epsilon \approx 1 + \epsilon,$$

which gives a more accurate approximation than the bounded total variation error.

The boosting lemma stated below says that for local Gibbs distributions, approximate inference with total variation error can be boosted into that with multiplicative error.

**Lemma 4.1** (boosting lemma). *For any class of local Gibbs distributions  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$ , if there is a LOCAL algorithm for approximate inference (within arbitrary total variation error  $\delta > 0$ ) with time complexity at most  $t(n, \delta)$ , then there is a LOCAL algorithm for approximate inference (within arbitrary multiplicative error  $0 < \epsilon < 1$ ) with time complexity  $O(t(n, \frac{\epsilon}{5qn}))$ , where  $q = |\Sigma| \leq \text{poly}(n)$  is the size of the alphabet.*

*Proof.* Let  $\mathcal{A}_\delta^+$  denote the LOCAL algorithm for approximate inference with arbitrary total variation error  $\delta > 0$  whose time complexity is  $t(n, \delta)$ . We construct a LOCAL algorithm  $\mathcal{A}_\epsilon^\times$  for approximate inference with arbitrary multiplicative error  $0 < \epsilon < 1$ .

Let  $(G, \mathbf{x}, \tau)$  be an instance, where  $G = (V, E)$ . The joint distribution  $\mu = \mu_{(G, \mathbf{x})}$  is a Gibbs distribution given by  $(G, \Sigma, \mathcal{F})$ , where  $q = |\Sigma|$ . Since the Gibbs distribution is local, we assume that there is an  $\ell = O(1)$  such that  $\forall (f, S) \in \mathcal{F}: \max_{u, v \in S} \text{dist}_G(u, v) \leq \ell$ .

**Algorithm  $\mathcal{A}_\epsilon^\times$ :** Let  $v \in V$ . We assume  $v \notin \Lambda$ , otherwise the inference problem is trivial. Let  $\delta = \frac{\epsilon}{5qn}$  and  $t = t(n, \frac{\epsilon}{5qn})$  be the time complexity of the LOCAL algorithm  $A_\delta^+$ . Node  $v$  collects all information up to distance  $2t + \ell$  and simulates the following algorithm locally.

Recall that  $B_r(v)$  denotes the  $r$ -ball centered at  $v$  in  $G$ . We define

$$\Gamma = B_{t+\ell}(v) \setminus (B_t(v) \cup \Lambda).$$

Let  $v_1, v_2, \dots, v_m$ , where  $m = |\Gamma|$ , be vertices in  $\Gamma$  enumerated in the increasing order of their unique IDs. A sequence of configurations  $\tau_i \in \Sigma^{\Lambda_i}$  on subsets  $\Lambda_i$ ,  $0 \leq i \leq m$ , is constructed as follows:

- Initially, let  $\Lambda_0 = \Lambda$  and  $\tau_0 = \tau$ .
- For  $i = 1, 2, \dots, m$ , let  $\Lambda_i = \Lambda_{i-1} \cup \{v_i\}$ , and the configurations  $\tau_i \in \Sigma^{\Lambda_i}$  is constructed such that  $\tau_i$  is consistent with  $\tau_{i-1}$  over  $\Lambda_{i-1}$ , and  $\tau_i(v_i) = c_i$  for the  $c_i \in \Sigma$  that maximizes the marginal probability  $\hat{\mu}_{v_i}^{\tau_{i-1}}(c_i)$  where  $\hat{\mu}_{v_i}^{\tau_{i-1}}$  is the marginal distribution returned by  $A_\delta^+$  at node  $v_i$  on the instance  $(G, \mathbf{x}, \tau_{i-1})$ .

Finally, the marginal distribution  $\mu_v^{\tau_m}$  is returned. Due to the conditional independence guaranteed by Proposition 2.1,  $\mu_v^{\tau_m}$  is fully determined by the information in  $B_{t+\ell}(v)$ . Specifically, denoted  $B = B_{t+\ell}(v)$ , and define  $\mathcal{C}$  to be the set of all configurations  $\sigma \in \Sigma^B$  consistent with  $\tau_m$  over  $B \cap \Lambda_m$ , i.e.

$$\mathcal{C} = \{\sigma \in \Sigma^B \mid \forall u \in B \cap (\Gamma \cup \Lambda) : \sigma_u = \tau_m(u)\}.$$

Then for every  $c \in \Sigma$ , the marginal probability  $\mu_v^{\tau_m}(c)$  is computed as

$$\mu_v^{\tau_m}(c) = \frac{\sum_{\sigma \in \mathcal{C} : \sigma_v = c} w_B(\sigma)}{\sum_{\sigma \in \mathcal{C}} w_B(\sigma)},$$

where  $w_B(\sigma) = \prod_{(f,S) \in \mathcal{F} : S \subseteq B} f(\sigma_S)$ . This finishes the definition of Algorithm  $\mathcal{A}_\epsilon^\times$ .

We then show that  $\tau_m$  is feasible with respect to  $\mu$ , so the marginal distribution  $\mu_v^{\tau_m}$  is well-defined. Furthermore, it holds that

$$\forall c \in \Sigma : \quad e^{-\epsilon} \mu_v^\tau(c) \leq \mu_v^{\tau_m}(c) \leq e^\epsilon \mu_v^\tau(c), \quad (3)$$

which proves the Theorem.

For the sequence of configurations  $\tau_i \in \Sigma^{\Lambda_i}$  on subsets  $\Lambda_i$ ,  $0 \leq i \leq m$ , constructed in Algorithm  $\mathcal{A}_\epsilon^\times$ , for every  $c \in \Sigma$ , let  $\tau_i^c \in \Sigma^{\Lambda_i \cup \{v\}}$  denote the configuration on  $\Lambda_i \cup \{v\}$  such that  $\tau_i^c$  is consistent with  $\tau_i$  over  $\Lambda_i$  and  $\tau_i^c(v) = c$ .

**Claim.** *If  $\tau_0^c$  is feasible with respect to  $\mu$ , then all  $\tau_i^c$  are feasible with respect to  $\mu$ .*

*Proof.* We prove this by induction on  $i$ . For  $i = 0$ , the claim holds trivially. For general  $i$ , suppose that  $\tau_{i-1}^c$  is feasible. Then  $\tau_{i-1}$  must be feasible in the first place because  $\tau_{i-1}^c$  extends  $\tau_{i-1}$ . The two configurations  $\tau_{i-1}^c$  and  $\tau_{i-1}$  differ only at vertex  $v$  and  $\text{dist}_G(v, v_{i-1}) > t$ , where  $t = t(n, \frac{\epsilon}{5qn})$ , Algorithm  $\mathcal{A}_\delta^\dagger$  will output the same marginal distribution  $\hat{\mu}_{v_{i-1}}^{\tau_{i-1}^c}$  at node  $v_{i-1}$  on the instances  $(G, \mathbf{x}, \tau_{i-1}^c)$  and  $(G, \mathbf{x}, \tau_{i-1})$ , such that  $d_{\text{TV}}(\hat{\mu}_{v_{i-1}}^{\tau_{i-1}^c}, \hat{\mu}_{v_{i-1}}^{\tau_{i-1}}) \leq \frac{\epsilon}{5qn}$  and  $d_{\text{TV}}(\hat{\mu}_{v_{i-1}}^{\tau_{i-1}^c}, \mu_{v_{i-1}}^{\tau_{i-1}}) \leq \frac{\epsilon}{5qn}$ . By triangle inequality,

$$d_{\text{TV}}(\mu_{v_{i-1}}^{\tau_{i-1}^c}, \mu_{v_{i-1}}^{\tau_{i-1}}) \leq \frac{2\epsilon}{5nq}. \quad (4)$$

Recall that  $\tau_i$  is constructed from  $\tau_{i-1}$  in such a way that  $\tau_i(v_i) = c_i$  for the  $c_i \in \Sigma$  that maximizes the marginal probability  $\hat{\mu}_{v_i}^{\tau_{i-1}}(c_i)$ , and  $q = |\Sigma|$ . Therefore, we have

$$\mu_{v_i}^{\tau_{i-1}}(\tau_i(v_i)) \geq \frac{1}{q} - \frac{\epsilon}{5nq}. \quad (5)$$

Note that  $\mu_{\Lambda_i \cup \{v\}}(\tau_i^c) = \mu_{\Lambda_{i-1} \cup \{v\}}(\tau_{i-1}^c) \cdot \mu_{v_i}^{\tau_{i-1}^c}(c_i)$ , where  $c_i = \tau_i(v_i)$ . By induction hypothesis  $\tau_{i-1}^c$  is feasible, then it holds that  $\mu_{\Lambda_{i-1} \cup \{v\}}(\tau_{i-1}^c) > 0$ . Combining with (4) and (5), we have

$$\mu_{\Lambda_i \cup \{v\}}(\tau_i^c) \geq \mu_{v_i}^{\tau_{i-1}^c}(c_i) \geq \mu_{v_i}^{\tau_{i-1}}(c_i) - \frac{2\epsilon}{5nq} \geq \frac{1}{q} - \frac{3\epsilon}{5nq} > 0,$$

which implies that  $\tau_i^c$  is feasible with respect to  $\mu$ . The claim is proved.  $\square$

Recall that  $\tau_0 = \tau$  for a feasible  $\tau \in \Sigma^\Lambda$ . There must exist  $c \in \Sigma$  such that  $\tau_0^c$  is feasible, which according to above claim implies that all  $\tau_i^c$  are feasible, and hence in particular,  $\tau_m^c$  is feasible, which means  $\tau_m$  is feasible in the first place since  $\tau_m^c$  extends  $\tau_m$ .

Consider each  $c \in \Sigma$  that  $\mu_v^\tau(c) > 0$ . For such  $c \in \Sigma$ ,  $\tau_0^c$  must be feasible since  $\tau_0 = \tau$ , which according to above claim implies that  $\tau_0^c$  is feasible. Denote that  $c_i = \tau_i(v_i)$  for  $0 \leq i \leq m$ . Apply the chain rule in two different orders of vertices, we have

$$\begin{aligned} \mu_{\Lambda \cup \{v\}}^\tau(\tau_m^c) &= \left( \prod_{i=1}^m \mu_{v_i}^{\tau_{i-1}}(c_i) \right) \mu_v^{\tau_m}(c), \\ \text{and } \mu_{\Lambda \cup \{v\}}^\tau(\tau_m^c) &= \mu_v^\tau(c) \left( \prod_{i=1}^m \mu_{v_i}^{\tau_{i-1}^c}(c_i) \right). \end{aligned}$$

Solving the above equations gives us

$$\mu_v^{\tau_m}(c) = \left( \prod_{i=1}^m \frac{\mu_{v_i}^{\tau_{i-1}^c}(c_i)}{\mu_{v_i}^{\tau_{i-1}}(c_i)} \right) \mu_v^\tau(c).$$

For each  $1 \leq i \leq m$ , by (4) and (5), it holds that

$$e^{-\epsilon/n} \leq 1 - \frac{2\epsilon}{5n - \epsilon} \leq \frac{\mu_{v_i}^{\tau_i^{c-1}}(c_i)}{\mu_{v_i}^{\tau_i}(c_i)} \leq 1 + \frac{2\epsilon}{5n - \epsilon} \leq e^{\epsilon/n}.$$

Since  $m = |\Lambda| \leq n$ , we have

$$e^{-\epsilon} \leq \frac{\mu_v^{\tau_m}(c)}{\mu_v^{\tau}(c)} \leq e^{\epsilon}.$$

For those  $c \in \Sigma$  that  $\mu_v^{\tau}(c) = 0$ , it holds that  $\tau_0^c$  is infeasible. It must hold that  $\mu_v^{\tau_m}(c) = 0$ , since if otherwise  $\mu_v^{\tau_m}(c) > 0$ , then  $\tau_m^c$  is feasible, contradicting that  $\tau_0^c$  is infeasible since  $\tau_m^c$  extends  $\tau_0^c$ .  $\square$

## 4.2 Distributed JVV Sampler

The Jerrum-Valiant-Vazirani (JVV) sampler [10] is a general global reduction from exact sampling to approximate counting for self-reducible problems via rejection sampling. Here we give a local distributed JVV sampler by realizing a local rejection sampling.

**Theorem 4.2.** *For a class of local Gibbs distributions  $\mathfrak{M} = \{\mu_{(G,\mathbf{x})}\}$ , if there is a LOCAL algorithm for approximate inference (with total variation error  $\leq 1/5qn^4$ ) with time complexity at most  $t(n)$ , where  $q = |\Sigma| \leq \text{poly}(n)$  is the size of the alphabet, then there is a LOCAL algorithm for exact sampling with time complexity  $O(t(n) \log^2 n)$ .*

By the boosting lemma (Lemma 4.1), the algorithm for approximate inference with total variation error in the assumption of Theorem 4.2 can be boosted to an approximate inference algorithm with multiplicative error  $1/n^3$  and time complexity  $O(t(n))$ . Theorem 4.2 is then a consequence of the following proposition.

**Proposition 4.3.** *For a class of local Gibbs distributions  $\mathfrak{M} = \{\mu_{(G,\mathbf{x})}\}$ , if there is a LOCAL algorithm for approximate inference (with multiplicative error  $\leq 1/n^3$ ) with time complexity at most  $t(n)$ , then there is a LOCAL algorithm for exact sampling with time complexity  $O(t(n) \log^2 n)$ .*

Let  $\mathcal{A}$  be the LOCAL algorithm for approximate inference with multiplicative error  $1/n^3$  and time complexity at most  $t = t(n)$ . We construct an SLOCAL algorithm called local-JVV for exact sampling with time complexity  $O(t)$ . For convenience, the local-JVV algorithm is presented as a multi-pass SLOCAL algorithm with the ability of writing nearby nodes' internal memories. The following results was observed in [8].

**Lemma 4.4** (Observation 2.2 in [8]). *Any SLOCAL algorithm  $\mathcal{A}$  with locality  $R$  in which each node  $v$  can write into the local memory  $S_u$  of other nodes  $u$  within its radius  $r \leq R$  can be transformed into an SLOCAL algorithm  $\mathcal{B}$  with locality  $r + R$  in which  $v$  writes only in its own memory  $S_v$ .*



Let  $(G, \mathbf{x}, \tau)$  be an instance, where  $G = (V, E)$ . The joint distribution  $\mu = \mu_{(G, \mathbf{x})}$  is a Gibbs distribution given by  $(G, \Sigma, \mathcal{F})$ , where  $q = |\Sigma|$ . Since the Gibbs distribution is local, we assume that there is an  $\ell = O(1)$  such that  $\forall (f, S) \in \mathcal{F} : \max_{u, v \in S} \text{dist}_G(u, v) \leq \ell$ . The  $\tau \in \Sigma^\Lambda$  is an arbitrary feasible configuration on an arbitrary subset  $\Lambda \subseteq V$ . The distribution  $\mu^\tau$  is the target distribution that we want to sample from.

#### 4.2.1 The local-JVV algorithm

The local-JVV algorithm is an SLOCAL algorithm for the *local rejection sampling*: Upon termination, the algorithm returns a  $(Y, F') = ((Y_v)_{v \in V}, (F'_v)_{v \in V})$ , where each node  $v \in V$  outputs  $Y_v$  and  $F'_v$ , such that  $Y \in \Sigma^V$  is a random configuration, and each  $F'_v \in \{0, 1\}$  indicates that the algorithm fails (rejects) locally at node  $v$ .

The SLOCAL algorithm consists of three passes. In each pass, the algorithm scans the nodes in the *same* ordering  $\pi = v_1, v_2, \dots, v_n$  provided to the algorithm by an adversary. The locality of each pass is  $R = 6t + 2\ell = O(t)$ .

In the first pass, a configuration  $\sigma_0 \in \Sigma^V$  that is feasible with respect to the target distribution  $\mu^\tau$ , called the *ground state*, is constructed by the following procedure:

- Initially,  $\sigma_0 \in \Sigma^\Lambda$  and  $\sigma_0 = \tau$ . Then  $\sigma_0$  is updated at each step.
- In the  $i$ -th step, simulate the algorithm  $\mathcal{A}$  at node  $v_i$  within radius  $t = t(n)$  on the instance  $(G, \mathbf{x}, \sigma_0)$ , which returns a marginal distribution  $\hat{\mu}_{v_i}^{\sigma_0}$ . Pick an arbitrary  $c_i \in \Sigma$  with  $\hat{\mu}_{v_i}^{\sigma_0}(c_i) > 0$ , and extend the current  $\sigma_0$  further onto  $v_i$  by setting  $\sigma_0(v_i) \leftarrow c_i$ .

In the second pass, a random configurations  $Y \in \Sigma^V$  is generated independently by the following procedure:

- Initially,  $Y \in \Sigma^\Lambda$  and  $Y = \tau$ . Then  $Y$  is randomly updated at each step.
- In the  $i$ -th step, simulate the algorithm  $\mathcal{A}$  at node  $v_i$  on the instance  $(G, \mathbf{x}, Y)$ , which returns a marginal distribution  $\hat{\mu}_{v_i}^Y$ . Sample a random  $Y_i \in \Sigma$  independently according to  $\hat{\mu}_{v_i}^Y$ , and extend the current  $Y$  further onto  $v_i$  by setting  $Y(v_i) \leftarrow Y_i$ .

Let  $\hat{\mu}^\tau$  be the distribution of  $Y \in \Sigma^V$  generated as above. It closely approximates the  $\mu^\tau$ .

**Claim 4.5.** For any  $\sigma \in \Sigma^V$ , it holds that  $e^{-1/n^2} \leq \frac{\hat{\mu}^\tau(\sigma)}{\mu^\tau(\sigma)} \leq e^{1/n^2}$ .

In the third pass, each node  $v_i$  carefully computes a probability  $q_{v_i}$ , and uses  $q_{v_i}$  to independently sample a random  $F'_{v_i} \in \{0, 1\}$  indicating the local failure at  $v_i$ .

- Initially, let  $\sigma_0 \in \Sigma^V$  be the ground state constructed in the first pass.
- A sequence of configurations  $\sigma_1, \sigma_2, \dots, \sigma_n \in \Sigma^V$  is constructed, where  $\sigma_n = Y$  is the configuration randomly generated in the second pass, such that the following

invariants hold for all  $0 \leq i \leq n$ :

$$\sigma_i \text{ is feasible with respect to } \mu^\tau; \quad (6)$$

$$\forall 1 \leq j \leq i: \quad \sigma_i(v_j) = Y(v_j); \quad (7)$$

$$\sigma_i \text{ and } \sigma_{i-1} \text{ differ only over the } t\text{-ball } B_t(v_i). \quad (8)$$

The invariants hold trivially for  $\sigma_0$ . In the  $i$ -th step, assume the above invariants for  $\sigma_{i-1}$ . Construct a  $\sigma_i \in \Sigma^V$  satisfying these invariants by enumerating all configurations  $\sigma' \in \Sigma^B$  over the  $t$ -ball  $B = B_t(v_i)$  and trying replacing the assignment of  $\sigma_{i-1}(B)$  with  $\sigma'$ . The following claim guarantees the existence of such  $\sigma_i \in \Sigma^V$ .

**Claim 4.6.** *Assume that  $\sigma_{i-1} \in \Sigma^V$  is feasible with respect to  $\mu^\tau$  and  $\sigma_{i-1}(v_j) = Y(v_j)$  for all  $1 \leq j \leq i-1$ . There exists a  $\sigma_i \in \Sigma^V$  satisfying (6), (7) and (8).*

Assuming this claim, the algorithm can always find a good  $\sigma_i$  and verify the invariants within bounded radius from  $v_i$ . Specifically, assuming the invariants for  $\sigma_{i-1}$ , the invariants (7) and (8) can be verified at  $v_i$  within radius  $t$ , and (6) can be verified within radius  $t + \ell$  due to the conditional independence property stated in Proposition 2.1. Once a good  $\sigma_i$  is found, node  $v_i$  updates the internal states of all nodes in  $B_t(v_i)$  to update the current configuration to  $\sigma_i$ .

Node  $v_i$  computes the value of

$$q_{v_i} = q_{v_i}(Y) \triangleq \frac{\hat{\mu}^\tau(\sigma_{i-1})w(\sigma_i)}{\hat{\mu}^\tau(\sigma_i)w(\sigma_{i-1})} e^{-3/n^2}, \quad (9)$$

where  $\hat{\mu}^\tau$  stands for the distribution of the random configuration  $Y$  generated in the second pass, and  $w(\cdot)$  is the weight for the Gibbs distribution defined in (1).

**Claim 4.7.** *The  $q_{v_i}$  defined in (9) can be computed at  $v_i$  within radius  $3t + \ell = O(t)$  and it always holds that  $e^{-5/n^2} \leq q_{v_i} \leq 1$ .*

At last,  $v_i$  samples a random  $F'_{v_i} \in \{0, 1\}$  such that  $F'_{v_i} = 0$  with probability  $q_{v_i}$ .

Finally, each node  $v_i$  returns a pair  $(Y_{v_i}, F'_{v_i})$  where  $Y = (Y_{v_i})_{1 \leq i \leq n}$  is the random configuration sampled in the second pass and the algorithm fails locally at  $v_i$  if  $F'_{v_i} = 1$ .

**Remark 4.1** (two-pass local-JVV algorithm). The third pass of local-JVV algorithm is not adapted to the second pass, because the invariant in (7) requires  $\sigma_i(v_j) = Y(v_j)$  only for  $1 \leq j \leq i$ . Hence, we can merge the second and the third pass into one pass. This gives a two-pass local-JVV algorithm, and the locality of each pass is  $R = 6t + 2\ell = O(t)$ .

### 4.2.2 Proofs of the three claims

*Proof of Claim 4.5.* Let  $Y^i$  denote the partially specified random configuration  $Y$  constructed in the  $i$ -th step in the second pass of the the local-JVV algorithm. For any  $\sigma \in \Sigma^V$ , let  $\sigma^i$  denote the  $Y^i$  when  $Y = \sigma$ , where  $\sigma^0 = \tau$  and  $\sigma^n = Y = \sigma$ . Clearly, for the distribution  $\hat{\mu}^\tau$  of  $Y$ ,

$$\hat{\mu}^\tau(\sigma) = \prod_{i=1}^n \hat{\mu}_{v_i}^{\sigma^{i-1}}(\sigma_{v_i}). \quad (10)$$

On the other hand, apply the chain rule to the Gibbs distribution  $\mu^\tau$ . We have

$$\mu^\tau(\sigma) = \prod_{i=1}^n \mu_{v_i}^{\sigma^{i-1}}(\sigma_{v_i}).$$

Since each marginal probability  $\hat{\mu}_{v_i}^{\sigma^{i-1}}(\sigma_{v_i})$  is computed by an approximate inference algorithm  $\mathcal{A}$  with multiplicative error  $1/n^3$ , it holds that

$$e^{-1/n^3} \leq \frac{\hat{\mu}_{v_i}^{\sigma^{i-1}}(\sigma_{v_i})}{\mu_{v_i}^{\sigma^{i-1}}(\sigma_{v_i})} \leq e^{1/n^3}.$$

The claim follows.  $\square$

*Proof of Claim 4.6.* Assume that  $\sigma_{i-1} \in \Sigma^V$  is feasible with respect to  $\mu^\tau$  and  $\sigma_{i-1}(v_j) = Y(v_j)$  for all  $1 \leq j \leq i-1$ . We show that there exists a  $\sigma_i \in \Sigma^V$  satisfying (6), (7), and (8). Define set of nodes

$$\Gamma = \{v_j \in V \mid v_j \notin B_t(v) \text{ or } j \leq i-1\} \cup \Lambda.$$

Let  $\tau_1 = \sigma_{i-1}(\Gamma)$  and  $\tau_2 = Y(\Gamma)$  be two configurations on subset  $\Gamma$ . By assumption and Claim 4.5,  $\sigma_{i-1}$  and  $Y$  are feasible, so  $\tau_1$  and  $\tau_2$  are also feasible and  $\sigma_{i-1}(\Lambda) = Y(\Lambda) = \tau$ .

The marginal distributions returned by  $\mathcal{A}$  at node  $v_i$  on the two instances  $(G, \mathbf{x}, \tau_1)$  and  $(G, \mathbf{x}, \tau_2)$  are identical, because  $\tau_1$  and  $\tau_2$  agree with each other over  $B_t(v_i)$ . We denote this marginal distribution as  $\hat{\mu}_{v_i}^{\tau_1} = \hat{\mu}_{v_i}^{\tau_2}$ , which is an approximation of the marginal distributions  $\mu_{v_i}^{\tau_1}, \mu_{v_i}^{\tau_2}$  with multiplicative error  $1/n^3$ . It holds that

$$e^{1/n^3} \cdot \mu_{v_i}^{\tau_1}(Y(v_i)) \geq \hat{\mu}_{v_i}^{\tau_1}(Y(v_i)) \geq e^{-1/n^3} \cdot \mu_{v_i}^{\tau_2}(Y(v_i)) > 0.$$

where the last inequality is due to that  $Y$  is feasible.

This shows that  $\mu_{v_i}^{\tau_1}(Y(v_i)) > 0$ , which means there exists a feasible configuration  $\sigma \in \Sigma^V$  such that  $\sigma(\Gamma) = \tau_1 = \sigma_{i-1}(\Gamma)$  and  $\sigma(v_i) = Y(v_i)$ . Such  $\sigma$  is the  $\sigma_i$  we want. It can be easily verified that it satisfies the invariants (6), (7) and (8).  $\square$

*Proof of Claim 4.7.* Recall that  $q_{v_i}$  is defined as

$$q_{v_i} = \frac{\hat{\mu}^\tau(\sigma_{i-1})w(\sigma_i)}{\hat{\mu}^\tau(\sigma_i)w(\sigma_{i-1})} e^{-3/n^2}.$$

Due to Claim 4.6, every  $\sigma_i$  is feasible with respect to  $\mu^\tau$ , so  $w(\sigma_i) > 0$  and moreover by Claim 4.5, we have  $\hat{\mu}^\tau(\sigma_i) > 0$ . The ratio  $q_{v_i}$  is well-defined.

Consider the configurations  $\sigma_0, \sigma_1, \dots, \sigma_n \in \Sigma^V$  constructed in the third pass of the local-JVV algorithm. For each  $\sigma_i$ , we define a sequence of configurations  $\sigma_i^0, \sigma_i^1, \dots, \sigma_i^n$ , where  $\sigma_i^0 = \tau$  and  $\sigma_i^n = \sigma_i$ . Let  $\sigma_i^j$  denote the  $Y^j$  when  $Y = \sigma_i$ , where  $Y^j$  denotes the partially specified random configuration  $Y$  constructed in the  $j$ -th step in the second pass of the the local-JVV algorithm. Due to (10), we have

$$\frac{\hat{\mu}^\tau(\sigma_{i-1})}{\hat{\mu}^\tau(\sigma_i)} = \prod_{j=1}^n \frac{\hat{\mu}_{v_j}^{\sigma_{i-1}^{j-1}}(\sigma_{i-1}(v_j))}{\hat{\mu}_{v_j}^{\sigma_i^{j-1}}(\sigma_i(v_j))}.$$

Note that the two configurations  $\sigma_i$  and  $\sigma_{i-1}$  differ only on the  $t$ -ball  $B_t(v_i)$ . Therefore, for all  $\sigma_i$  and  $\sigma_{i-1}$  agree with each other over  $B_t(v_k)$  for any  $v_k \notin B_{2t}(v_i)$ . The two instances  $(G, \mathbf{x}, \sigma_{i-1}^{k-1})$  and  $(G, \mathbf{x}, \sigma_i^{k-1})$  are indistinguishable to the  $t$ -local approximate inference algorithm  $\mathcal{A}$ , thus

$$\hat{\mu}_{v_k}^{\sigma_{i-1}^{k-1}}(\sigma_{i-1}(v_k)) = \hat{\mu}_{v_k}^{\sigma_i^{k-1}}(\sigma_i(v_k)),$$

which means

$$\frac{\hat{\mu}^\tau(\sigma_{i-1})}{\hat{\mu}^\tau(\sigma_i)} = \prod_{v_j \in B_{2t}(v_i)} \frac{\hat{\mu}_{v_j}^{\sigma_{i-1}^{j-1}}(\sigma_{i-1}(v_j))}{\hat{\mu}_{v_j}^{\sigma_i^{j-1}}(\sigma_i(v_j))}, \quad (11)$$

where each marginal probability  $\hat{\mu}_{v_j}^{\sigma_i^{j-1}}(\cdot)$  can be computed by the local algorithm  $\mathcal{A}$  at node  $v_j$  within radius  $t$ . In addition, to compute the ratio in (11), we still need to know the relative ordering of all nodes in  $B_{3t+\ell}(v_i)$  with respect to the ordering  $\pi = (v_1, v_2, \dots, v_n)$ . Recall  $R = 6t + 2\ell$ . In the first path, we can compute a value  $\text{rank}(v_i) \in \mathbb{N}$  for each vertex  $v_i \in V$ . Initially, assume  $\text{rank}(v_i) = 0$  from all  $v_i \in V$ . When processing the node  $v_i$  in the first pass, we update

$$\text{rank}(v_i) \leftarrow 1 + \max\{\text{rank}(v_j) \mid v_j \in B_R(v_i)\}, \quad (12)$$

and write the updated value  $\text{rank}(v_i)$  into the memory  $S_{v_i}$ . The value  $\text{rank}(v_i)$  can be computed within locality  $R = 6t + 2\ell$ . It is easy to verify that all the values  $\text{rank}(u)$  are distinct for  $u \in B_{3t+\ell}(v_i)$ , and for any  $v_j, v_k \in B_{3t+\ell}(v_i)$ ,  $j < k$  if and only if  $\text{rank}(v_j) < \text{rank}(v_k)$ . Thus, in the third pass, we can find the relative ordering of all nodes in  $B_{3t+\ell}(v_i)$  by comparing their values  $\text{rank}(\cdot)$ . We remark that when transforming an SLOCAL algorithm into

a LOCAL algorithm, a network decomposition is computed at the first step. Hence, the LOCAL algorithm knows the relative ordering even without the values  $\text{rank}(\cdot)$ .

On the other hand, since the two configurations  $\sigma_i$  and  $\sigma_{i-1}$  differ only on nodes in  $B_t(v_i)$ , by the definition of the weight  $w(\cdot)$  in (1), we have

$$\frac{w(\sigma_i)}{w(\sigma_{i-1})} = \prod_{\substack{(S,f) \in \mathcal{F} \\ S \subseteq B_{t+\ell}(v_i)}} \frac{f(\sigma_i(S))}{f(\sigma_{i-1}(S))}. \quad (13)$$

Equations (11) and (13) imply that the quantity  $q_{v_i}$  can be computed at node  $v_i$  by gathering all information up to distance  $3t + \ell$ .

For the Gibbs distribution  $\mu$ , it always holds that  $\frac{\mu^\tau(\sigma_{i-1})w(\sigma_i)}{\mu^\tau(\sigma_i)w(\sigma_{i-1})} = 1$  since the measure  $\mu$  is defined proportional to the weights  $w(\cdot)$ . Due to Claim 4.5, we have

$$e^{-2/n^2} \leq \frac{\hat{\mu}^\tau(\sigma_{i-1})w(\sigma_i)}{\hat{\mu}^\tau(\sigma_i)w(\sigma_{i-1})} \leq e^{2/n^2},$$

which implies  $e^{-5/n^2} \leq q_{v_i}(\sigma) \leq e^{-1/n^2} \leq 1$ .  $\square$

### 4.2.3 Proof of Proposition 4.3

The correctness of the local-JVV algorithm is guaranteed by the following lemma.

**Lemma 4.8.** *On any ordering of nodes, the local-JVV algorithm fails with probability at most  $\sum_{i=1}^n \mathbb{E}[F'_{v_i}] = 1 - O(1/n)$  and conditioning on that  $F'_{v_i} = 0$  for all  $v_i$ , the  $Y \in \Sigma^V$  randomly sampled by the algorithm follows precisely the target distribution  $\mu^\tau$ .*

*Proof.* Let  $\mathcal{G}$  denote the event that the local-JVV algorithm succeeds, i.e.  $F'_{v_i} = 0$  for all  $v_i$ . Recall that for any  $\sigma \in \Sigma^V$ ,

$$\Pr[F'_{v_i} = 1 \mid Y = \sigma] = q_{v_i}(\sigma),$$

where  $q_{v_i}(\sigma) = q_{v_i}(Y)|_{Y=\sigma}$  is as defined in (9). Due to Claim 4.7, for any  $\sigma \in \Sigma^V$ ,

$$\Pr[\mathcal{G} \mid Y = \sigma] = \prod_{i=1}^n q_{v_i}(\sigma) \geq e^{-5/n} = 1 - O\left(\frac{1}{n}\right).$$

This proves that the algorithm succeeds with probability  $\Pr[\mathcal{G}] = 1 - O\left(\frac{1}{n}\right)$ .

Next, we show that  $\Pr[Y = \sigma \mid \mathcal{G}] = \mu^\tau(\sigma)$  for any  $\sigma \in \Sigma^V$ , which proves the lemma.

For any  $\sigma \in \Sigma^V$  that  $\mu^\tau(\sigma) = 0$ , by Claim 4.5, we have  $\hat{\mu}^\tau(\sigma) = \Pr[Y = \sigma] = 0$ , therefore  $\Pr[Y = \sigma \mid \mathcal{G}] = 0$ .

For any  $\sigma \in \Sigma^V$  that  $\mu^\tau(\sigma) > 0$ , by Claim 4.5, we have  $\hat{\mu}^\tau(\sigma) = \Pr[Y = \sigma] > 0$ . The probability that the algorithm succeeds and outputs  $Y = \sigma$  is given by

$$\begin{aligned} \Pr[Y = \sigma \wedge \mathcal{G}] &= \hat{\mu}^\tau(\sigma) \Pr[\mathcal{G} \mid Y = \sigma] \\ &= \hat{\mu}^\tau(\sigma) \prod_{i=1}^n q_{v_i}(\sigma) \\ &= \hat{\mu}^\tau(\sigma) \prod_{i=1}^n \left( \frac{\hat{\mu}^\tau(\sigma_{i-1}) w(\sigma_i)}{\hat{\mu}^\tau(\sigma_i) w(\sigma_{i-1})} e^{-3/n^2} \right) \Big|_{\sigma_n=Y=\sigma} \\ &= \left( \frac{\hat{\mu}^\tau(\sigma_0)}{w(\sigma_0)} \cdot e^{-3/n} \right) w(\sigma). \end{aligned}$$

Note that the factor  $\left( \frac{\hat{\mu}^\tau(\sigma_0)}{w(\sigma_0)} \cdot e^{-3/n} \right)$  is independent of  $\sigma$ . Therefore,

$$\begin{aligned} \Pr[Y = \sigma \mid \mathcal{G}] &= \frac{\Pr[Y = \sigma \wedge \mathcal{G}]}{\sum_{\sigma': \mu^\tau(\sigma') > 0} \Pr[Y = \sigma' \wedge \mathcal{G}]} \\ &= \frac{w(\sigma)}{\sum_{\sigma': \mu^\tau(\sigma') > 0} w(\sigma')} \\ &= \mu^\tau(\sigma). \end{aligned}$$

□

Finally, we are going to prove Proposition 4.3. It can be easily verified that the local-JVV algorithm has locality  $O(t)$  where  $t = t(n)$  is the time complexity of the LOCAL algorithm  $\mathcal{A}$ .

We apply Lemma 4.4 and Lemma 3.1 to transform the local-JVV algorithm defined in the SLOCAL model to a LOCAL algorithm. Due to Lemma 3.1, upon successful termination, the LOCAL algorithm preserves the distribution of the output  $(Y, F')$  of the SLOCAL algorithm on some ordering of nodes, where  $Y = (Y_v)_{v \in V}$  is the randomly sampled configuration and  $F' = (F'_v)_{v \in V}$  is the vector of random indicators for local failures generated in the final pass of the local-JVV algorithm. The transformation to the LOCAL model will introduce another random failure  $F''_v \in \{0, 1\}$  to each node  $v$  which is independent of  $(Y, F')$ , where  $F''_v = 1$  indicates the failure at node  $v$  caused by the network decomposition and  $\sum_{v \in V} \mathbb{E}[F''_v] = O(1/n^2)$ . We combine the two failures and define  $F_v = F'_v \vee F''_v$  to indicate the failure of the LOCAL algorithm at node  $v$ . Clearly,

$$\sum_{v \in V} \mathbb{E}[F_v] \leq \sum_{v \in V} \mathbb{E}[F'_v] + \sum_{v \in V} \mathbb{E}[F''_v] = O\left(\frac{1}{n}\right).$$

Furthermore, since  $(F''_v)_{v \in V}$  is independent of  $(Y, F')$ , it still holds that conditioning on that  $F_v = 0$  for all nodes  $v \in V$ , the distribution of  $Y$  is precisely the target distribution.

**Remark 4.2** (one-pass local-JVV algorithm in stronger SLOCAL model). One can consider a stronger SLOCAL model such that given an arbitrary ordering  $\pi = (v_1, v_2, \dots, v_n)$ , the stronger SLOCAL algorithm with locality  $r(n)$  knows the relative ordering (with respect to  $\pi$ ) of all nodes in  $B_{r(n)}(v_i)$  when processing the node  $v_i$ . Consider the two-pass local-JVV algorithm (Remark 4.1) in this stronger SLOCAL model. In the first pass, the algorithm only needs to compute a feasible configuration  $\sigma_0 \in [q]^V$ , but does not need to compute the value  $\text{rank}(\cdot)$  in (12). It can be verified that the following two facts hold in the stronger SLOCAL model:

- the two-pass local-JVV algorithm is still correct even if the two passes are scanned in two arbitrary orderings (not necessarily the same ordering);
- the two-pass local-JVV algorithm does not need to use the extra power of the stronger SLOCAL model in the first pass.

Also note that Lemma 4.4 holds for the stronger SLOCAL model. Thus, by the reduction in the proof of [8, Lemma 2.3], we can merge the two passes and obtain a one-pass local-JVV algorithm with locality  $4R = 24t + 8\ell = O(t)$  in the stronger SLOCAL model.

In [8, Theorem 1.11], when transforming an SLOCAL algorithm into a LOCAL algorithm, a network decomposition is computed in the first step. Each node is assigned a unique label after the network decomposition. Then, the SLOCAL algorithm is simulated according to the ordering of the labels. Hence, the relative ordering of all nodes can be computed by comparing their labels. This implies Lemma 3.1 still holds for the stronger SLOCAL model. We can directly transform this one-pass local-JVV algorithm in stronger SLOCAL model into a LOCAL algorithm with locality  $O(t \log^2 n)$  by the same transformation.

## 5 Approximate Inference and Strong Spatial Mixing

In this section, we explore the intrinsic relation between distributed sampling/counting problems and decays of correlation in joint distributions. The main result in this section holds for locally admissible, local Gibbs distributions.

An important decay of correlation property for joint distributions is the *strong spatial mixing*. We adopt the definition of strong spatial mixing in [23] into our context.

**Definition 5.1** (strong spatial mixing). Let  $\delta_n : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$  be a sequence of non-increasing functions. A class of joint distributions  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$  is said to exhibit *strong spatial mixing* with rate  $\delta_n(\cdot)$  if for every distribution  $\mu_{(G, \mathbf{x})} \in \mathfrak{M}$  over  $\Sigma^V$ , where  $G = (V, E)$  and  $n = |V|$ , for every  $v \in V$ ,  $\Lambda \subseteq V$ , and any two feasible configurations  $\sigma, \tau \in \Sigma^\Lambda$ ,

$$d_{\text{TV}}(\mu_v^\sigma, \mu_v^\tau) \leq \delta_n(\text{dist}_G(v, D)), \quad (14)$$

where  $D \subseteq \Lambda$  is the subset on which  $\sigma$  and  $\tau$  differ.

In particular, the strong spatial mixing is said to be *with exponential decay at rate  $\alpha$* , for some  $0 < \alpha < 1$ , if the mixing rate  $\delta_n(\cdot)$  is in the form  $\delta_n(t) = \text{poly}(n) \cdot \alpha^t$ .

The strong spatial mixing is intrinsically related to the approximate inference in the LOCAL model. In fact, the strong spatial mixing can be thought as a weaker form of approximate inference in the LOCAL model, where every node knows the graph  $G$  and distribution  $\mu$  but not the partially specified feasible configuration  $\tau$ . Therefore, it is quite natural that the approximate inference always implies strong spatial mixing. Meanwhile, the converse also holds for locally admissible, local Gibbs distributions.

**Theorem 5.1.** *For any class of joint distributions  $\mathfrak{M} = \{\mu_{(G,\mathbf{x})}\}$ , if there is a LOCAL algorithm for approximate inference (within arbitrary total variation error  $\delta > 0$ ) with time complexity at most  $t(n, \delta)$ , then  $\mathfrak{M}$  exhibits strong spatial mixing with rate*

$$\delta_n(t) = 2 \min\{\delta \mid t(n, \delta) \leq t - 1\}.$$

*Conversely, for any class of locally admissible, local Gibbs distributions  $\mathfrak{M} = \{\mu_{(G,\mathbf{x})}\}$ , if  $\mathfrak{M}$  exhibits strong spatial mixing with rate  $\delta_n(t)$ , then there is a LOCAL algorithm for approximate inference (within arbitrary total variation error  $\delta > 0$ ) with time complexity*

$$t(n, \delta) = \min\{t \mid \delta_n(t) \leq \delta\} + O(1).$$

*Proof.* Let  $\mu = \mu_{(G,\mathbf{x})} \in \mathfrak{M}$  be a joint distribution over  $\Sigma^V$ , where  $G = (V, E)$ . Let  $\sigma, \tau \in \Sigma^\Lambda$  be two feasible configurations on subset  $\Lambda \subseteq V$  that differ over  $D \subseteq \Lambda$ . Fix any vertex  $v \notin D$ . Suppose that  $\text{dist}_G(v, D) = t$ .

Let  $\mathcal{A}$  denote the LOCAL algorithm for approximate inference with time complexity  $t(n, \delta)$ . By Proposition 3.3, we can assume without loss of generality that  $\mathcal{A}$  is deterministic.

For any  $\delta > 0$ , if  $t(n, \delta) \leq t - 1$ , then algorithm  $\mathcal{A}$  (given the total variation error  $\delta$ ) will return the same marginal distribution  $\hat{\mu}_v$  at node  $v$  on the two instances  $(G, \mathbf{x}, \sigma)$  and  $(G, \mathbf{x}, \tau)$ , because the two instances are indistinguishable for the algorithm at node  $v$ , and it is guaranteed that  $d_{\text{TV}}(\hat{\mu}_v, \mu_v^\sigma) \leq \delta$  and  $d_{\text{TV}}(\hat{\mu}_v, \mu_v^\tau) \leq \delta$ . Since this holds for any  $\delta > 0$  such that  $t(n, \delta) \leq t - 1$ , we have

$$\begin{aligned} d_{\text{TV}}(\hat{\mu}_v, \mu_v^\sigma) &\leq \min\{\delta \mid t(n, \delta) \leq t - 1\}, \\ d_{\text{TV}}(\hat{\mu}_v, \mu_v^\tau) &\leq \min\{\delta \mid t(n, \delta) \leq t - 1\}, \end{aligned}$$

which implies  $d_{\text{TV}}(\mu_v^\sigma, \mu_v^\tau) \leq 2 \min\{\delta \mid t(n, \delta) \leq t - 1\}$  by triangle inequality. Therefore,  $\mathfrak{M}$  exhibits strong spatial mixing with rate  $\delta_n(t) = 2 \min\{\delta \mid t(n, \delta) \leq t - 1\}$ .

Conversely,  $\mathfrak{M}$  is a class of locally admissible, local Gibbs distributions and  $\mathfrak{M}$  exhibits strong spatial mixing with rate  $\delta_n(t)$ . We construct a LOCAL algorithm  $\mathcal{A}$  for approximate inference with arbitrary total variation error  $\delta > 0$ .

Let  $(G, \mathbf{x}, \tau)$  be an instance, where  $G = (V, E)$ . The joint distribution  $\mu = \mu_{(G,\mathbf{x})} \in \mathfrak{M}$  is a Gibbs distribution specified by  $(G, \Sigma, \mathcal{F})$ , where  $q = |\Sigma|$ . Since the Gibbs distribution is local, we assume that there is an  $\ell = O(1)$  such that  $\forall (f, S) \in \mathcal{F} : \max_{u,v \in S} \text{dist}_G(u, v) \leq \ell$ .



The  $\tau \in \Sigma^\Lambda$  is an arbitrary feasible configuration on an arbitrary subset  $\Lambda \subseteq V$  and  $\mu^\tau$  is the target distribution.

The algorithm  $\mathcal{A}$  is described as follows. For each node  $v \in V$ , node  $v$  gathers all information up to distance  $t + 2\ell$ , where

$$t = \min\{t' \mid \delta_n(t') \leq \delta\},$$

and simulates the following procedure locally:

- Node  $v$  extends  $\tau$  to a feasible configuration  $\tau'$  on subset  $\Lambda \cup \Gamma$ , where

$$\Gamma = B_{t+\ell}(v) \setminus (B_t(v) \cup \Lambda),$$

by enumerating all configurations in  $\Sigma^\Gamma$ . Such feasible  $\tau'$  must exist since  $\tau$  is feasible. Since  $\tau$  itself is feasible and the distribution  $\mu$  is locally admissible, then  $\tau'$  is feasible if and only if

$$\prod_{\substack{(f,S) \in \mathcal{F} \\ S \subseteq A}} f(\tau'_S) > 0, \quad (15)$$

where  $A = B_{t+2\ell}(v) \cap (\Gamma \cup \Lambda)$ . Hence, this condition can be checked by  $v$  locally.

- Node  $v$  returns the marginal distribution  $\mu_v^{\tau'}$ . Due to the conditional independence guaranteed by Proposition 2.1,  $\mu_v^{\tau'}$  is fully determined by the information in  $B_{t+\ell}(v)$ . Specifically, denoted  $B = B_{t+\ell}(v)$ , and define the set of configurations

$$\mathcal{C} = \{\sigma \in \Sigma^B \mid \forall u \in B \cap (\Gamma \cup \Lambda) : \sigma_u = \tau'_u\}.$$

Then for every  $c \in \Sigma$ , the marginal probability  $\mu_v^{\tau'}(c)$  can be computed as

$$\mu_v^{\tau'}(c) = \frac{\sum_{\sigma \in \mathcal{C} : \sigma_v = c} w_B(\sigma)}{\sum_{\sigma \in \mathcal{C}} w_B(\sigma)},$$

where  $w_B(\sigma) = \prod_{(f,S) \in \mathcal{F} : S \subseteq B} f(\sigma_S)$ .

Let  $\mathcal{S}$  be the set of all feasible configurations  $\sigma$  on subset  $\Lambda \cup \Gamma$  such that  $\sigma$  is consistent with  $\tau$  over  $\Lambda$ . Fix any  $\sigma \in \mathcal{S}$ . Let  $D$  denote the subset on which  $\sigma$  and  $\tau'$  disagree. Note that  $\text{dist}_G(v, D) \geq t$ . Due to the strong spatial mixing,

$$d_{\text{TV}}(\mu_v^{\tau'}, \mu_v^\sigma) \leq \delta_n(t) \leq \delta.$$

We couple the two distributions  $\mu_v^{\tau'}$  and  $\mu_v^\sigma$  as follows: First sample a  $\sigma \in \mathcal{S}$  with probability  $\mu_{\Lambda \cup \Gamma}^\tau(\sigma)$ , then use the optimal coupling between  $\mu_v^{\tau'}$  and  $\mu_v^\sigma$  to sample a pair  $(x, y) \in \Sigma \times \Sigma$ .

It is easy to verify that the marginal distribution of  $y$  resulting from this two-step sampling is just  $\mu_v^\tau$ . By the coupling Lemma

$$\begin{aligned} d_{\text{TV}}(\mu_v^{\tau'}, \mu_v^\tau) &\leq \Pr[x \neq y] \\ &= \sum_{\sigma \in \mathcal{S}} \mu_{\Lambda \cup \Gamma}^\tau(\sigma) \cdot \Pr[x \neq y \mid \sigma] \\ &= \sum_{\sigma \in \mathcal{S}} \mu_{\Lambda \cup \Gamma}^\tau(\sigma) \cdot d_{\text{TV}}(\mu_v^{\tau'}, \mu_v^\sigma) \\ &\leq \delta. \end{aligned}$$

Hence,  $\mathcal{A}$  is a LOCAL algorithm for approximate inference with time complexity at most  $t(n, \delta) = t + 2\ell = \min\{t \mid \delta_n(t) \leq \delta\} + O(1)$ .  $\square$

The strong spatial mixing defined in Definition 5.1 measures the decay of correlation in terms of total variation distance. If we replace the total variation distance  $d_{\text{TV}}(\cdot, \cdot)$  in (14) with the multiplicative error function  $\text{err}(\cdot, \cdot)$  defined in (2), we have an even stronger form of strong spatial mixing, namely the one with decay in multiplicative error. Several well-known strong spatial mixing results for important classes of Gibbs distributions (e.g. independent sets, matchings, and graph colorings) were actually established in this stronger form [23, 2, 5, 6]. Here we see this is not a coincidence. Combing Theorem 5.1 with the boosting lemma (Lemma 4.1), we have the following corollary.

**Corollary 5.2.** *A class of locally admissible, local Gibbs distributions  $\mathfrak{M}$  exhibits strong spatial mixing with exponential decay at rate  $\alpha$  in total variation distance, if and only if it exhibits strong spatial mixing with exponential decay at rate  $\alpha$  in multiplicative error.*

Interestingly, the corollary gives a result in probability theory proved by local computation.

Combining Theorem 5.1 with the distributed JVV sampler (Theorem 4.2), we have the followings.

**Corollary 5.3.** *For any class of locally admissible, local Gibbs distributions  $\mathfrak{M} = \{\mu_{(G, \mathbf{x})}\}$ , if  $\mathfrak{M}$  exhibits strong spatial mixing with exponential decay at rate  $\alpha$  for some  $\alpha < 1$ , then there is a LOCAL algorithm for exact sampling with time complexity  $O(\frac{1}{1-\alpha} \log^3 n)$ .*

Combining with the state-of-the-arts for strong spatial mixing in [2, 23, 6, 12, 20], the corollary gives us the following LOCAL algorithm for exact sampling:

- an  $O(\sqrt{\Delta} \log^3 n)$ -round algorithm for sampling matchings in graphs with maximum degree  $\Delta$  due to the strong spatial mixing of matchings with exponential decay at rate  $1 - \Omega(1/\sqrt{\Delta})$  [2];

- an  $O(\log^3 n)$ -round algorithm for sampling independent sets in graphs with max-degree  $\Delta \leq 5$ , or more generally, for sampling according to the hardcore model (weighted independent sets) with fugacity  $\lambda$  up to the uniqueness threshold (where  $\lambda < \lambda_c(\Delta) \triangleq (\Delta - 1)^{(\Delta-1)}/(\Delta - 2)^\Delta$ ), due to the strong spatial mixing of the model up to the uniqueness threshold [23];
- an  $O(\log^3 n)$ -round algorithm for sampling  $q$ -colorings of triangle-free graphs when  $q \geq \alpha\Delta$  for  $\alpha > \alpha^*$  where  $\alpha^* \approx 1.763\dots$  satisfies  $\alpha^* = \exp(\frac{1}{\alpha^*})$ , due to the strong spatial mixing proved in [6];
- an  $O(\log^3 n)$ -round algorithm for sampling according to the anti-ferromagnetic 2-spin model in the interior of the uniqueness regime, due to the strong spatial mixing of the model in the uniqueness regime [12];
- an  $O(\log^3 n)$ -round algorithm for sampling weighted hypergraph matchings up to the uniqueness threshold (when the weight  $\lambda < \lambda_c(r, \Delta) \triangleq \frac{(\Delta-1)^{(\Delta-1)}}{(r-1)(\Delta-2)^\Delta}$ , where  $r$  is the rank of the hypergraph), due to the strong spatial mixing of the model up to the uniqueness threshold [20].

The definitions of these models are given in the referred papers. All these joint distributions are either locally admissible, local Gibbs distributions, or in the case of edge models (e.g. graph/hypergraph matchings) can be represented as such joint distributions through dualities of graphs/hypergraphs, which preserve the distances.

For lower bounds, the long-range correlation established in a previous work ([3], Theorem 5.3) implies an  $\Omega(\text{diam})$  lower bound for approximate sampling according to the hardcore model with fugacity  $\lambda$  in the non-uniqueness regime (where  $\lambda > \lambda_c(\Delta)$ ). Along with the  $O(\log^3 n)$  upper bound for exact sampling according to the hardcore model in the uniqueness regime obtained above, we discover for the first time a computational phase transition for local distributed sampling and counting, at the same critical threshold for the computational phase transition discovered for sampling and counting on polynomial-time Turing machines [23, 19].

## 6 Conclusion

We study the complexities of sampling and counting in the LOCAL model, where the counting is represented by a local variant, namely the inference problem. We found that for self-reducible problems, the well known generic relations between sampling and counting on classic polynomial-time Turing machines hold similarly for local computation. Meanwhile, the tractability of these problems by local computation is captured by a decay of correlation property known as the strong spatial mixing.

Perhaps a lesson we could learn from this research is that it is helpful to model local computation problems as joint distributions, and hence studying the complexities of these

problems is reduced to studying the discrepancies between such problem-specified joint distributions and the distributions that can be generated by local algorithms. This new approach for local computation seems to have much potential.

Several open problems are worth investigating. First, can we make the LOCAL algorithms in this paper use bounded-size messages and bounded local computation? Such efficient distributed algorithms would necessarily improve the state of the arts of sampling and approximate counting on polynomial-time Turing machines. Second, how should we classify complexities of sampling and counting in local computation, and does there exist a complexity hierarchy? Third, the distributed JVV sampler given in this paper terminates in a fixed number of rounds with bounded locally certifiable failure. Can we make this algorithm Las Vegas, in a sense that the time complexity of the algorithm may be random but once it terminates the algorithm always outputs precisely according to the correct distribution, and still being local? This requires a strategy for non-biased local resampling, which is far from being well understood. So far, it was only discovered for the Lovász-local-lemma-based sampler for restrictive problems under strict conditions [9].

## References

- [1] Baruch Awerbuch, M Luby, A V Goldberg, and Serge A Plotkin. Network decomposition and locality in distributed computation. In *Proceedings of the 30th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 364–369, 1989.
- [2] Mohsen Bayati, David Gamarnik, Dimitriy Katz, Chandra Nair, and Prasad Tetali. Simple deterministic approximation algorithms for counting matchings. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC)*, pages 122–127, 2007.
- [3] Weiming Feng, Yuxin Sun, and Yitong Yin. What can be sampled locally? In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 121–130, 2017.
- [4] Weiming Feng and Yitong Yin. Distributed symmetry breaking in sampling: (optimal distributed randomly coloring with fewer colors). Submitted.
- [5] David Gamarnik and Dimitriy Katz. Correlation decay and deterministic FPTAS for counting list-colorings of a graph. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1245–1254, 2007.
- [6] David Gamarnik, Dimitriy Katz, and Sidhant Misra. Strong spatial mixing of list coloring of graphs. *Random Structures & Algorithms*, 2013.
- [7] Mohsen Ghaffari, David G Harris, and Fabian Kuhn. On derandomizing local distributed algorithms. *arXiv preprint arXiv:1711.02194*, 2017.

- [8] Mohsen Ghaffari, Fabian Kuhn, and Yannic Maus. On the complexity of local distributed graph problems. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC)*, pages 784–797, 2017.
- [9] Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the Lovász local lemma. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 342–355, 2017.
- [10] M R Jerrum, L G Valiant, and V V Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.
- [11] Mark Jerrum. *Counting, Sampling and Integrating: Algorithms and Complexity*. Lectures in Mathematics. ETH Zurich. Birkhäuser Basel, 2003.
- [12] Liang Li, Pinyan Lu, and Yitong Yin. Correlation decay up to uniqueness in spin systems. In *Proceedings of the 24th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 67–84, 2013.
- [13] Nathan Linial and Michael Saks. Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993.
- [14] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.
- [15] Moni Naor and Larry Stockmeyer. What can be computed locally? *SIAM Journal on Computing (SICOMP)*, 24(6):1259–1277, 1995.
- [16] Alessandro Panconesi and Aravind Srinivasan. On the complexity of distributed network decomposition. *Journal of Algorithms*, 20(2):356–374, 1996.
- [17] Mark A Paskin and Carlos E Guestrin. Robust probabilistic inference in distributed systems. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence (UAI)*, pages 436–445, 2004.
- [18] David Peleg. *Distributed computing: a locality-sensitive approach*. SIAM, 2000.
- [19] Allan Sly. Computational transition at the uniqueness threshold. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 287–296, 2010.
- [20] Renjie Song, Yitong Yin, and Jinman Zhao. Counting hypergraph matchings up to uniqueness threshold. In *RANDOM*, 2016.
- [21] Larry J. Stockmeyer. The complexity of approximate counting. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC)*, pages 118–126, 1983.

- [22] Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.
- [23] Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 140–149, 2006.