Parallelize Single-Site Dynamics up to Dobrushin Criterion

Hongyang Liu *                              Yitong Yin*

Abstract

Single-site dynamics are canonical Markov chain based algorithms for sampling from high-dimensional distributions, such as the Gibbs distributions of graphical models. We introduce a simple and generic parallel algorithm that faithfully simulates single-site dynamics. Under a much relaxed, asymptotic variant of the $\ell_p$-Dobrushin’s condition—where the Dobrushin’s influence matrix has a bounded $\ell_p$-induced operator norm for an arbitrary $p \in [1, \infty]$—our algorithm simulates $N$ steps of single-site updates within a parallel depth of $O(N/n + \log n)$ on $\tilde{O}(m)$ processors, where $n$ is the number of sites and $m$ is the size of the graphical model. For Boolean-valued random variables, if the $\ell_p$-Dobrushin’s condition holds—specifically, if the $\ell_p$-induced operator norm of the Dobrushin’s influence matrix is less than 1—the parallel depth can be further reduced to $O(\log N + \log n)$, achieving an exponential speedup.

These results suggest that single-site dynamics with near-linear mixing times can be parallelized into RNC sampling algorithms, independent of the maximum degree of the underlying graphical model, as long as the Dobrushin influence matrix maintains a bounded operator norm. We show the effectiveness of this approach with RNC samplers for the hardcore and Ising models within their uniqueness regimes, as well as an RNC SAT sampler for satisfying solutions of CNF formulas in a local lemma regime. Furthermore, by employing non-adaptive simulated annealing, these RNC samplers can be transformed into RNC algorithms for approximate counting.

*State Key Laboratory for Novel Software Technology, New Cornerstone Science Laboratory, Nanjing University, China.
E-mails: liuhongyang@smail.nju.edu.cn, yinyt@nju.edu.cn
1 Introduction

Drawing random samples according to prescribed probability distributions is a fundamental computational problem. Historically, Monte Carlo simulations, which rely on random sampling, were among the earliest computer programs [M+87]. Today, sampling from high-dimensional distributions remains a central challenge across various fields of computer science and data science.

The Markov chain Monte Carlo (MCMC) method is one of the primary methods for sampling. A significant portion of Markov chains used for sampling from high-dimensional distributions belongs to the class of single-site dynamics. Let $V$ be a set of $n$ sites, and $Q$ be a set of $q = |Q|$ spins. Let $\mu$ be a distribution defined over all configurations $\sigma \in Q^V$. A canonical Markov chain for sampling from $\mu$ is the following single-site dynamics, known as the Glauber dynamics (also called the Gibbs sampler or heat-bath dynamics):

- Given the current configuration $\sigma \in Q^V$, a site $v \in V$ is picked uniformly at random, and its spin $\sigma_v$ is updated by drawing a new spin independently according to the marginal distribution $\mu_{\sigma|\setminus v}^V$.

It is well known that the stationary distribution of this chain is $\mu$. Furthermore, when $\mu$ is a Gibbs distribution of a graphical model defined by local constraints, there exists an underlying graph $G = (V, E)$ such that the marginal distributions $\mu_{\sigma|\setminus v}^V$ depend only on $\sigma_{N(v)}$, the values of $\sigma$ at the neighbors of $v$.

Abstractly, a single-site dynamics on a graph $G = (V, E)$ can be defined by a class of local update distributions $\{P^v_\tau\}$, where each $P^v_\tau$ is a distribution over $Q$. This distribution is determined by the site $v \in V$ chosen for update and the current configuration $\tau = \sigma_{N(v)}$ on $v'$s inclusive neighborhood $N_{v'}$.

Parallel MCMC sampling. MCMC sampling plays a pivotal role in efficiently performing Monte Carlo calculations for a variety of complex tasks, including volume estimation and integration [DFK91], approximation of partition functions [SVV09, J93], permanent computation [JSV04], and counting satisfying solutions [FGYZ21]. These tasks are fundamental to statistical inference and data analysis, which have become major focuses in the era of big data. As data scales, the demand for solving these tasks efficiently by leveraging parallel computing resources has grown significantly, making the parallelization of MCMC sampling increasingly critical. A substantial body of research, both in practice [JLY19, NRRW11, SN10, HUDO16, GLGG11, AGC22, SBB22, DDJ18, FX14, MS15] and in theory [CFV24, AHL23, ABTV23, FHY21a, FVY21, AHSS21, BRY20, GLG19, FY18, FG18, DDJ18, SY17, DSO16, GLGG11], is dedicated to advancing parallel MCMC methods. This ongoing effort aims to address the challenge of efficiently conducting Monte Carlo calculations on massive datasets in contemporary computational environments.

The single-site dynamics are inherently defined in a sequential manner. This raises intriguing theoretical questions about whether such sequentiality is intrinsic to the problems they solve. In a seminal work on parallel computing, Mulmuley, Vazirani and Vazirani [MVV87] asked whether an efficient parallel algorithm could exist for sampling bipartite perfect matchings, a key step towards approximating permanents in parallel. However, Teng [Ten95] later provided negative evidence to this question by identifying barriers to parallelizing Markov chains, particularly those related to perfect matchings. Teng further conjectured that 0-1 permanents are not efficiently approximable in the $\text{NC}$ class, making it a rare example of a problem believed to be intrinsically sequential but not known to be $\text{P}$-complete. This conjecture, along with the barrier, reflects the belief that simulating a dynamical system may be inevitably sequential.

In fact, single-site dynamics originated as an idealized continuous-time parallel process [Gla63].

---

1 We use the inclusive neighborhood $N_{v'} = \{u \mid \{u, v\} \in E \} \cup \{v\}$ in this abstraction of single-site dynamics because, in some single-site dynamics, the rule for updating a site $v$ may depend on the current spin of $v$ itself, e.g. in Metropolis chains.
The Continuous-Time Single-Site Dynamics

The continuous-time process \((X_t)_{t \in \mathbb{R}_{\geq 0}}\) on space \(Q^V\) evolves as:

- each site \(v \in V\) holds an independent rate-1 Poisson clock;
- when a clock at a site \(v \in V\) rings, the spin \(X_t(v)\) is redrawn according to \(P_{v}^{X_t(N_v^+)}\).

This continuous-time process effectively parallelizes the single-site dynamics, achieving a speedup factor of \(n = |V|\). This is because a continuous time \(T \in \mathbb{R}_{\geq 0}\) corresponds to \(N\) discrete steps, where \(N \sim \text{Pois}(nT)\).

Although this idealized parallel process has been known for over half a century, little is known about how to implement it correctly and efficiently on a parallel computer. A major obstacle is a classical conundrum in concurrency: if two adjacent sites concurrently update their spins based on the current configurations of their neighborhoods, it can lead to an incorrect simulation of the original continuous-time process, where updates are atomic. On the other hand, avoiding this issue by disallowing concurrent updates of adjacent sites incurs an extra time complexity factor of the degree \(\Delta_G\) (or more precisely, the chromatic number \(\chi_G\)) of the graphical model \(G\), since at most one site per neighborhood can be updated at any moment. This obstacle is encountered by traditional parallel samplers based on chromatic scheduler [GLGG11]. For general graphical models \(G\) with \(n\) sites, the maximum degree \(\Delta_G\) or the chromatic number \(\chi_G\) can be as large as \(\Omega(n)\), making it challenging to simulate MCMC sampling using RNC programs within polylogarithmic depth of parallel computations.

We are therefore concerned with the following fundamental question: Can we simulate continuous-time single-site dynamics both faithfully and efficiently on parallel computers? The question concerns simulating an idealization of our physical world [Gla63] on man-made computing systems with minimal errors or overheads. This goal appears ambitious, especially considering the barriers identified in [Ten95]. Recently, some positive results have been shown for a subclass of single-site dynamics known as Metropolis chains [FHY21a]. However, this approach was specifically tailored to these chains, where updates can be reduced to coin flipping. For general single-site dynamics, particularly Glauber dynamics, where updates involve more complex decisions beyond Boolean choices, it remains largely uncertain whether these processes can be efficiently and faithfully simulated in parallel.

### 1.1 Our Results

We present a simple and generic parallel algorithm that can faithfully simulate single-site dynamics.

**Dobrushin’s criterion.** The parallel simulation is efficient under a condition formulated similarly to Dobrushin’s conditions, but is significantly weaker in scale.

**Definition 1.1** (Dobrushin’s influence matrix [Dob70]). The Dobrushin’s influence matrix \(\rho \in \mathbb{R}_{\geq 0}^{V \times V}\) for the single-site dynamics on the graph \(G = (V, E)\) specified by \(\{P_v^\tau\}\) is defined as:

\[
\forall u, v \in V : \quad \rho(u, v) \triangleq \max_{\tau, \tau' : \tau \oplus \tau' \subseteq \{u\}} d_{TV}(P_v^\tau, P_v^{\tau'}),
\]

where \(d_{TV}(\cdot, \cdot)\) denotes the total variation distance. The maximum is taken over all pairs of configurations \(\tau, \tau' \in Q^{N_v^+}\) on the inclusive neighborhood \(N_v^+ = N_v \cup \{v\}\) of \(v\), such that \(\tau\) and \(\tau'\) agree with each other everywhere except at \(u\).
We define the following condition for the operator norm of Dobrushin’s influence matrix to be arbitrarily bounded, but not necessarily by a constant:

**Condition 1.** There is a $p \in [1, \infty]$ such that $\|\rho\|_p \leq \rho$ for some $\rho > 0$.

It is well known that contraction of Dobrushin’s influence matrix in operator norms implies the mixing of the chain. Specifically, $\|\rho\|_1 = \max_u \sum_v \rho(u,v) < 1$ corresponds to the Dobrushin condition \[\text{Dob70}], and $\|\rho\|_\infty = \max_u \sum_v \rho(u,v) < 1$ corresponds to the Dobrushin-Shlosman condition \[\text{DS85a, DS85b}], both of which imply optimal mixing times. In a seminal work \[\text{Hay06}], the $\ell_2$-Dobrushin’s condition $\|\rho\|_2 < 1$ was studied, which can be related to several key approaches for mixing, including coupling \[\text{Hay06}], spectral gap \[\text{EKZ21}], log-Sobolev \[\text{Mar19}] and modified log-Sobolev inequalities \[\text{BCC+22}]. For spin systems, this condition was generalized by Dyer, Goldberg, and Jerrum to $\|\rho\| < 1$ for arbitrary matrix norms \[\text{DGJ09}].

**Main Theorems.** Let $(X_t)_{t\in \mathbb{R}_{\geq 0}}$ denote a continuous-time single-site dynamics on the state space $Q^V$, defined by local update distributions $P^\tau_v$ specified on the graph $G = (V, E)$. The single-site dynamics are presented to the algorithm as oracles for evaluating the distributions $P^\tau_v$. Specifically, given any $v \in V$, $\tau \in Q^V_x$, and $x \in Q$, the oracle returns the probability value $P^\tau_v(x)$. We present a parallel algorithm in CRCW-PRAM for simulating single-site dynamics.

**Theorem 1.2** (informal). For any single-site dynamics on a graph $G = (V, E)$ that satisfies Condition 1 with parameter $\rho$, there exists a parallel algorithm that can simulate the continuous-time chain $(X_t)_{0 \leq t \leq T}$ from any initial state $X_0 \in Q^V$ up to any time $0 < T < \text{poly}(|V|)$, with $O(\rho \cdot (T + \log |V|))$ depth on $\tilde{O}(|E| + |V| \cdot |Q|^2)$ processors with high probability.

The algorithm is a Las Vegas algorithm that guarantees to return a faithful copy of $X_T$ upon termination, while the complexity bounds hold with high probability $(1 - |V|^{-\tilde{O}(1)})$. The $\tilde{O}()$ notation hides poly-logarithmic factors.

Furthermore, when a stronger Dobrushin condition is satisfied, essentially the same parallel algorithm for perfect simulation can terminate much faster, achieving an exponential speedup. This is showcased by the following theorem.

**Theorem 1.3** (informal). For any single-site dynamics on graph $G = (V, E)$ with domain size $|Q| = 2$ satisfying Condition 1 with parameter $\rho < 1$, there exists a parallel algorithm that can simulate the continuous-time chain $(X_t)_{0 \leq t \leq T}$ from any initial state $X_0 \in Q^V$ up to any time $0 < T < \text{poly}(|V|)$, with $O(\frac{1}{1-\rho} (\log T + \log |V|))$ depth on $O(|E| \cdot T)$ processors with high probability.

**Remark 1.4.** We note that when applying the parallel simulation for the purpose of drawing approximate samples, the exponential speedup achieved in Theorem 1.3 may not offer a significant advantage over Theorem 1.2. This is because, as shown in \[\text{DGJ09}], for spin systems, under the condition $\|\rho\|_p < 1$, the continuous chain mixes within time $T = \tilde{O}(\log n)$. With such $T$, the complexity bounds in Theorems 1.2 and 1.3 are essentially the same.
Nevertheless, we still present Theorem 1.3 separately, because it holds more generally beyond spin systems, showing that super-linear speedup is achievable for perfect simulation of single-site dynamics, which is conceptually significant.

The parallel algorithm described above can also be implemented in the CONGEST model, providing an efficient distributed algorithm for simulating single-site dynamics.

**Theorem 1.5** (informal). For any single-site dynamics on a graph \( G = (V, E) \) that satisfies Condition 1 with parameter \( \rho \), there exists a distributed algorithm that can simulate the continuous-time chain \( (X_t)_{0 \leq t \leq T} \) from any initial state \( X_0 \in Q^V \) up to any time \( 0 < T < \text{poly}(|V|) \), within \( O(\rho \cdot (T + \log |V|)) \) rounds on the network \( G \), using messages each of \( O(\log |V| \cdot \log |Q|) \) bits.

In the CONGEST model, where each node has unlimited computational power and local memory, the update distributions \( P_v^T \) can be provided to each node \( v \) as local input. The algorithm is a Monte Carlo algorithm that terminates in a fixed number of rounds and succeeds with high probability.

**Useful Consequences.** Note that in the above theorems, \( T \) represents the continuous-time duration, which corresponds to \( N \) discrete update steps where \( N \sim \text{Pois}(nT) \), with \( n = |V| \). According to a generic lower bound \([HS07]\), single-site dynamics requires \( \Omega(n \log n) \) discrete steps to mix. Consequently, when applied to such mixing chains, the above algorithms achieve a parallel speedup of \( \Theta(n) \).

For example, for any reversible and ergodic (irreducible and aperiodic) single-site dynamics, if the Dobrushin-Shlosman condition \( \|\rho\|_\infty < 1 \) holds (which implies contraction of path coupling), the chain mixes within \( n^{-O(1)} \) total variation distance to the stationary distribution \( \pi \) in \( O(n \log n) \) steps or \( O(\log n) \) continuous time. Combined with **Theorem 1.2**, this leads to the following straightforward corollary for MCMC sampling in the RNC class (problems solvable by randomized parallel algorithms with poly-logarithmic depth and polynomial processors).

**Corollary 1.6.** For any reversible and ergodic single-site dynamics with RNC-computable initial states and local update distributions, if the Dobrushin’s influence matrix satisfies \( \|\rho\|_\infty < 1 \), there exists an RNC algorithm that can draw approximate samples (within any \( n^{-O(1)} \) total variation distance) from the stationary distribution \( \pi \).

One can replace the above condition \( \|\rho\|_\infty < 1 \) in Corollary 1.6 with any sufficient condition that implies both \( O(n) \) mixing time and Condition 1 with \( \rho = O(1) \). In such cases, the corollary still holds. The same applies to any sufficient condition that implies both \( \log(n) \) mixing time and Condition 1 with \( \rho < 1 \). However, in many cases, especially for spin systems, the mixing time bound may already be implied by Condition 1 with \( \rho < 1 \) \([DG09]\).

**Remark 1.7.** Condition 1 with \( \rho = O(1) \) is significantly weaker than known Dobrushin conditions for mixing or known sufficient conditions for efficient parallelizing single-site dynamics. For example, Condition 1 with \( \rho = O(1) \) holds if a weakened Dobrushin-Shlosman condition, \( \|\rho\|_\infty = O(1) \), is satisfied. This condition essentially means that the discrepancy in path coupling can grow by at most an \( O(1) \) factor per step, implying that “disagreements do not propagate super-exponentially.”

For various graphical models, such as the Ising model, hardcore model, and proper coloring on graphs with bounded maximum degree, this condition is no stronger than the necessary conditions for the chains to mix. Moreover, it is much weaker than the Lipschitz condition in \([FHY21]\) for parallelizing Metropolis chains, which would imply that \( \|\rho\|_1 = O(1) \) for the influence matrix \( \rho \) of the Metropolis chain.
Hardcore and Ising samplers. Graphical models can represent high-dimensional (Gibbs) distributions using local factors. Let $G = (V, E)$ be an undirected graph. Let $\lambda > 0$ be the fugacity (or external field, vertex activity) and $\beta > 0$ be the temperature (or edge activity). The hardcore Gibbs distribution, denoted $\mu^{\text{hardcore}}$, is defined over all $\sigma \in \{0, 1\}^V$ indicating independent sets in $G$ as follows:

$$
\forall \text{independent set } \sigma \in \{0, 1\}^V, \quad \mu^{\text{hardcore}}(\sigma) \propto \lambda^{\|\sigma\|_1}.
$$

The Ising Gibbs distribution, denoted $\mu^{\text{Ising}}$, is defined over all $\sigma \in \{0, 1\}^V$ as follows:

$$
\forall \sigma \in \{0, 1\}^V, \quad \mu^{\text{Ising}}(\sigma) \propto \beta^{m(\sigma)}\lambda^{\|\sigma\|_1},
$$

where $m(\sigma) \triangleq \sum_{(u, v) \in E} I[\sigma_u = \sigma_v]$ counts the number of monochromatic edges in $\sigma$.

The normalizing factors of these Gibbs distributions are known as the partition functions, which are essential for counting algorithms. For graphs $G$ with maximum degree $\Delta = \Delta_G \geq 3$, the uniqueness conditions for the hardcore and Ising models are respectively given by $\lambda < \lambda_c(\Delta) \triangleq (\frac{\Delta-1}{\Delta-2})^\frac{1}{\Delta}$ and $\beta < (\frac{\Delta-2}{\Delta-\Delta})$. Beyond these regimes, either the single-site dynamics is slow mixing or the sampling problem itself becomes computationally intractable [GM07, SS14, GˇSV16].

Under the uniqueness condition, optimal $O(n \log n)$ mixing times for the Glauber dynamics have been established for these models through a series of breakthroughs [CFYZ22, CE22, AJK+22, CFYZ21, CLV21, CLV20, ALO20]. Additionally, for the hardcore model with $\lambda = O(\frac{1}{\sqrt{n}})$ and the Ising model with $\beta \geq 1 \pm O(\frac{1}{\sqrt{n}})$, Condition 1 holds with $\rho = O(1)$. Consequently, we obtain the following corollary.

**Corollary 1.8** (Hardcore and Ising samplers). For graphs $G = (V, E)$ with maximum degree $\Delta = \Delta_G \geq 3$, there exist parallel samplers for the Gibbs distributions of the following models:

- hardcore model with fugacity $\lambda \leq (1 - \delta)\lambda_c(\Delta) = (1 - \delta)(\frac{(\Delta-1)^{\frac{1}{\Delta}}}{\Delta-2})$ for $\delta \in (0, 1)$,

- Ising model with temperature $\beta \in (\frac{\Delta-2+\delta}{\Delta-\delta}, \frac{\Delta-\delta}{\Delta-2+\delta})$ for $\delta \in (0, 1)$,

such that an approximate sample within $\frac{1}{\text{poly}(|V|)}$ total variation distance from the Gibbs distribution can be obtained within $O_\delta(\log \Delta \cdot \log |V|)$ depth on $\tilde{O}(|E| + |V|)$ processors.

To the best of our knowledge, these are the first RNC samplers for any graphical model with unbounded maximum degree up to the critical condition. Similar results can be obtained for general two-state anti-ferromagnetic spin systems under the critical conditions characterized in [CFYZ22].

The parallel samplers can be implemented as $O(\log n)$-round CONGEST($\log n$) algorithms on the network $G$, where the local update distributions can be evaluated using 1-round communication.

The parallel samplers in Corollary 1.8 can be transformed into parallel approximate counting algorithms using non-adaptive simulated annealing. By applying the Chebyshev’s cooling schedule from [DFK91] and [SVV09, Corollary 2.4], the hardcore and Ising samplers in Corollary 1.8 can be transformed into $O(\log n \cdot \log \Delta)$-depth algorithms on $\tilde{O}(mn^2/\epsilon^2)$ processors for $\epsilon$-approximations of the hardcore and Ising partition functions, under the same respective uniqueness conditions. To the best of our knowledge, these are the first RNC approximation algorithms for the partition function of any graphical model with unbounded maximum degree.

**SAT sampler.** Consider conjunctive normal form (CNF) formulas over $n$ Boolean variables. A CNF formula $\Phi$ is called a $(k, d)$-formula if all of its clauses have size $k$ and each variable appears in at most $d$ clauses. In [FCYZ21, Moi19], a poly$(d, k) \cdot n^{1+2^{-20}}$-time algorithm is presented for sampling almost uniform satisfying solutions, under a local lemma condition $k \geq 20\log k + 20\log d + 60$. Here we show that this algorithm can be parallelized within the RNC class.
**Corollary 1.9 (SAT sampler).** There is a parallel algorithm such that for any \((k, d)\)-formula \(\Phi\) satisfying
\[
k \geq 20 \log k + 20 \log d + 60,
\]
the algorithm outputs an almost uniform satisfying assignment for \(\Phi\) within \(\frac{1}{\text{poly}(n)}\) total variation distance. This is achieved using \(\text{polylog}(n)\) depth on \(\text{poly}(d, n)\) processors, where \(n\) is the number of variables in \(\Phi\).

The Glauber dynamics is implemented on a more complicated joint distribution \(\mu_{\Delta}\), constructed by projecting the uniform distribution \(\mu\) over all SAT solutions of \(\Phi\) onto a carefully chosen subset of variables \(M \subset V\). This projected Markov chain was introduced because the original solution space may be disconnected through local Markov chains. Tighter analyses in the follow-up works \([FHY21b, JPV21, HSW21]\) have proved the rapid mixing and efficient simulations of the projected Markov chains under weaker local lemma conditions, with improved constants, than the one in (2).

Similar improved bounds can also be achieved by the RNC samplers using more involved analyses.

Similarly, by the non-adaptive annealing provided in \([FGYZ21, Algorithm 7]\), we obtain a \(\text{polylog}(n)\)-depth \(\text{poly}(d, n)\)-processor algorithm for approximately counting the number of satisfying solutions of any \((k, d)\)-CNF satisfying (2).

Proofs of Corollary 1.8 and 1.9 are straightforward and provided in Section 6 for completeness.

### 1.2 Related Work in Correlated Sampling

A key step in our algorithm for parallel simulation of single-site dynamics relies on generating each single-site update with a “universal coupling” of randomness (formally defined in Definition 4.1), which synchronizes all local update distributions \(P^\tau\) across different neighborhood configurations \(\tau\). While this coupling does not change the definition or sequential simulation of the Markov chain, it is essential for achieving efficient parallel simulation while maintaining the accuracy of the simulation. Notably, it overcomes the barrier identified in \([Len95]\).

Perhaps due to its natural definition, this machinery of universally coupling distributions over the same sample space has actually been explored in various independent contexts to solve diverse problems. These include MinHash sketching \([Bro97, Cha02]\), rounding linear programming relaxations \([KT02]\), parallel repetition of 2-player 1-round games \([Hol07, Rao08, BHH08]\), cryptography \([Riv16]\), and replicability and differential privacy of learning \([BGH+23, GKM21, KKMV+23, KYYZ23]\).

The problem is now commonly referred as correlated sampling and was formalized in \([BGH+20]\) as follows.

**Definition 1.10 (correlated sampling).** A correlated sampling strategy for a finite sample space \(\Omega\) with error rate \(\epsilon: [0, 1] \rightarrow [0, 1]\) is a procedure \(\text{Sample} : \Delta(\Omega) \times [0, 1] \rightarrow \Omega\), such that

- **Correctness:** \(\forall p \in \Delta(\Omega), x \in \Omega, \Pr_{\mathcal{R} \sim [0,1]}[\text{Sample}(p, \mathcal{R}) = x] = p(x)\);
- **Error rate:** \(\forall p, q \in \Delta(\Omega)\) with \(d_{TV}(p, q) = \delta, \Pr_{\mathcal{R} \sim [0,1]}[\text{Sample}(p, \mathcal{R}) \neq \text{Sample}(q, \mathcal{R})] \leq \epsilon(\delta)\).

In above, \(\Delta(\Omega) \triangleq \{ p \in [0, 1]^\Omega \mid \sum_{x \in \Omega} p(x) = 1 \}\) denotes the probability simplex on the sample space \(\Omega\).

In particular, the correlated sampling strategy used in our paper achieves an optimal error rate \(\epsilon(\delta) = 2\delta/(1 + \delta)\), and was independently discovered in \([KT02]\) and \([Hol07]\). This will be discussed in detail in Section 4.

To the best of our knowledge, the present work is the first to apply the correlated sampling method to parallelize stochastic processes. Very recently, Anari, Gao, and Rubinstein \([AGR24]\) generalized and applied the correlated sampling approach to derive a parallel sampler for arbitrary high-dimensional distributions, achieving sub-linear depth and polynomial total work, assuming the availability of a counting oracle.
2 Preliminaries

Single-site dynamics. Let $V$ be a set of $n$ sites, and let $Q = \{1, 2, \ldots, q\}$ be a finite set of $q \geq 2$ spins.

A configuration on $S \subseteq V$ is an assignment $\sigma \in Q^S$ of spins to the sites in $S$.

Let $G = (V, E)$ be an undirected graph on vertex set $V$. For each vertex $v \in V$, let $N_v \triangleq \{u \in V \mid \{u, v\} \in E\}$ denote the neighborhood of $v$ in $G$, and let $N_v^+ \triangleq N_v \cup \{v\}$ denote the inclusive neighborhood of $v$ in $G$.

We use the term single-site dynamics on graph $G$ to refer to a specific type of Markov chain on space $Q^V$. Let $\{P^\tau_v\}$ be a collection of local update distributions, such that for every site $v \in V$ and every configuration $\tau \in N_v^+$ on $v$’s inclusive neighborhood, $P^\tau_v$ is a distribution over all spins in $Q$. A Markov chain on space $Q^V$ is specified by $\{P^\tau_v\}$. At the current configuration $\sigma \in Q^V$, the chain transitions as follows:

1. Pick $v \in V$ uniformly at random.
2. Replace the spin $\sigma(v)$ of $v$ with a spin drawn independently according to $P^\tau_v$, where $\tau = \sigma(N_v^+)$.

The collection $\{P^\tau_v\}$ is also called the local transition rule for the chain.

Examples. Given a distribution $\mu$ over $Q^V$, for any $v \in V$ and any possible $\tau \in Q^S$ where $S \subseteq V \setminus \{v\}$, let $\mu^\tau_v$ denote the marginal distribution at $v$ induced by $\mu$, given the configuration on $S$ being fixed as $\tau$. Formally, for each $x \in Q$,

$$\mu^\tau_v (x) = \Pr_{\sigma \sim \mu} [\sigma_v = x \mid \sigma_S = \tau].$$

Consider a Gibbs distribution $\mu$ defined on graph $G$ over sample space $Q^V$, such that for any $v \in V$ and any possible $\sigma \in Q^V$, the marginal distribution $\mu^\tau_v$ depends only on the values of $\sigma$ at $v$’s neighbors. That is, $\mu^\tau_v = \mu^\tau_{N_v}$. Examples include:

- **Hardcore model:** Let $\lambda > 0$. A Gibbs distribution $\mu$ is defined over all such $\sigma \in \{0, 1\}^V$ such that $\mu(\sigma) \propto \lambda|\{v \in V : \sigma_v = 1\}|$ if $\{v \in V : \sigma_v = 1\}$ forms an independent set in $G$, and $\mu(\sigma) = 0$ otherwise.

- **Ising/Potts/coloring models:** Let $\beta \geq 0$. A Gibbs distribution $\mu$ is defined over all $\sigma \in Q^V$, such that $\mu(\sigma) \propto \beta^{m(\sigma)}$, where $m(\sigma)$ denotes the number of monochromatic edges in $\sigma$.

Two basic classes of single-site dynamics are:

- **Glauber dynamics:** (a.k.a. Gibbs sampler, heat-bath dynamics): The local update distributions are $P^\tau_v = \mu^{\tau N_v}$.

- **Metropolis chain:** To update $v$’s spin in $\sigma \in Q^V$, first propose to replace $\sigma(v)$ with a spin $x \in Q$ chosen uniformly at random, and then accept it with probability $\min \left\{ 1, \frac{\mu^{\tau N_v}(x)}{\mu^{\tau N_v}(\sigma_v)} \right\}$.

Formally, the local update distributions are defined as $P^\tau_v (x) = \frac{1}{|Q|} \min \left\{ 1, \frac{\mu^{\tau N_v}(x)}{\mu^{\tau N_v}(\tau_v)} \right\}$ for $x \neq \tau_v$, and $P^\tau_v (\tau_v) = 1 - \sum_{x \neq \tau_v} P^\tau_v (x)$.

It is well known that both chains are reversible with respect to the stationary distribution $\mu$. These chains can also be defined for general (not necessarily Gibbs) distributions $\mu$ over $Q^V$ by setting $G$ as the complete graph, thus making the marginal distributions $\mu^{\tau N_v} = \mu^{\tau V \setminus \{v\}}$.

For Gibbs distributions $\mu$ defined by hard constraints, where infeasible configurations have zero probability, one can routinely extend the definition of $\mu^{\tau N_v}$ to infeasible conditions $\tau$ by ignoring the constraints violated locally by $\tau$. For instance, for the uniform distribution $\mu$ over proper $q$-colorings of
G, if $\tau \in Q^N_C$ is already improper, $\mu^*_v$ can still be defined as the uniform distribution over all available colors in $Q \setminus \{ \tau_u \mid u \in N_v \}$, provided $q > \Delta_G$. This ensures that the single-site dynamics may absorb to feasible states even if starting from an infeasible one.

Therefore, in this paper, we assume that for a single-site dynamics the local update distributions $\{ P^*_v \}$ are defined for all sites $v \in V$ and neighborhood configurations $\tau \in Q^N_C$.

**The continuous-time chain.** The continuous-time variant of the single-site dynamics is defined as follows:

1. Time progresses continuously from 0, with each site $v \in V$ associated with an independent rate-1 Poisson clock. Let $S_i = \sum_{1 \leq j \leq i} T_j$, where $T_1, T_2, \ldots$ are independent and identically distributed exponential random variables of rate 1. The Poisson clock rings at time $S_1, S_2, \ldots$.

2. Whenever the clock at a site $v \in V$ rings, the spin of $v$ is updated according to the local transition rule $\{ P^*_v \}$ as in the discrete-time chain.

Let $(X^C_t)_{t \in \mathbb{R}_{\geq 0}}$ denote this continuous-time process, and let $(X^D_t)_{t \in \mathbb{N}_{\geq 0}}$ denote its discrete-time counterpart. The two processes are equivalent up to a speedup factor $n = |V|$.

**Proposition 2.1.** Conditioning on the same initial configuration $X^C_0 = X^D_0 \in Q^C$, for any $T > 0$, $X^C_T$ is identically distributed as $X^D_N$, where $N$ follows the Poisson distribution Pois$(nT)$, where $n = |V|$.

**Mixing time.** According to the Markov chain convergence theorem [LPW17], an irreducible and aperiodic Markov chain on a finite state space converges to a unique stationary distribution $\pi$. The mixing time measures the rate at which the chain converges to this stationary distribution. For two distributions $\mu, \nu$ over the same sample space $\Omega$, the total variation distance between $\mu$ and $\nu$ is defined as $d_{TV}(\mu, \nu) \triangleq \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)|$. For a chain $X_t$ on the space $Q^C$ that converges to the stationary distribution $\pi$, and for any configuration $\sigma \in Q^C$, let $p^*_\sigma$ denote the distribution of $X_t$, given $X_0 = \sigma$. The mixing time is defined as $t_{mix}(\epsilon) \triangleq \max_{\sigma \in Q^C} \min \{ t > 0 \mid d_{TV}(p^*_\sigma, \pi) \leq \epsilon \}$.

Let $t_{mix}^C(\epsilon)$ and $t_{mix}^D(\epsilon)$ denote the mixing times of the continuous-time single-site dynamics $(X^C_t)_{t \in \mathbb{R}_{\geq 0}}$ and its discrete-time counterpart $(X^D_t)_{t \in \mathbb{N}_{\geq 0}}$, respectively, where both are defined by the same local transition rule $\{ P^*_v \}$. The following relation between the two mixing times follows immediately from Proposition 2.1:

$$n \cdot t_{mix}^C(\epsilon) = O \left( t_{mix}^D(\epsilon/2) + \log \frac{1}{\epsilon} \right),$$

where $n = |V|$ is the number of sites.

In particular, an $O(n \log n)$ mixing time for the discrete-time chain implies an $O(\log n)$ mixing time for the continuous-time chain. Furthermore, due to a general lower bound [HS07], these mixing times are optimal for single-site dynamics.

**Computation models.** In this paper, we assume the concurrent-read concurrent-write (CRCW) parallel random access machine (PRAM) model with arbitrary write, where an arbitrary value written concurrently is stored. The depth of an algorithm in this model refers to the number of time steps required for its execution.

We use NC to denote both the class of parallel algorithms with poly-logarithmic depth and polynomial processors and the class of problems solvable by such algorithms. The class RNC refers to the randomized counterpart of NC.
The CONGEST model is defined on an undirected network $G = (V, E)$, where the nodes represent processors. Initially, each node receives its local input and private random bits. Communications are synchronized and proceed in rounds. In each round, each node may perform arbitrary local computation and send a $B$-bit message to each of its neighbors. The messages are received by the end of the round. The model is denoted as CONGEST(B).

### 3 A Locally-Iterative Algorithm for Simulating Markov Chain

We introduce a locally-iterative algorithm that simulates single-site dynamics. The algorithm falls into the general framework of message passing algorithms, such as the well-known belief propagation (BP) algorithms [Pea82, MM09]. In this algorithm, each site maintains an internal state, and all internal states are updated iteratively based on the current internal states within their local neighborhoods until a fixpoint is reached.

Let $(X_t)_{t \in \mathcal{S}_{0}}$ be a continuous-time single-site dynamics on a graph $G = (V, E)$, defined by local update distributions $\{P^v_t\}$. Our goal is to simulate this Markov chain up to a fixed time $T > 0$. For each site $v \in V$, a rate-1 Poisson clock runs independently up to time $T$, generating a sequence of times:

$$0 = t^v_0 < t^v_1 < \cdots < t^v_m < T, \text{where } m_v \sim \text{Pois}(T).$$

With probability $1$, all $t^v_i$ values for $i \geq 1$ are distinct. We refer to such a collection of time sequences for all sites $\mathcal{S} = (t^v_i)_{v \in V, 0 \leq i \leq m_v}$ as an update schedule up to time $T$.

Given an update schedule $\mathcal{S}$, we identify the $i$-th update at site $v$ (which occurs at time $t^v_i$) by the pair $(v, i)$. To perform this update, the current spin $X_t(v)$ at site $v$ is replaced by a random spin generated independently according to $P^v_t$. Here, $\tau \in Q^v_{\mathcal{S}}$ is constructed as follows: for each $u \in \mathcal{N}(v)$, $\tau_u = X_{t^v_\tau}(u)$ represents the spin generated during the $j_u$-th update at site $u$, where $j_u = \max\{j \geq 0 \mid t^u_j < t^v_i\}$.

The evolution of this process can then be described by the following abstract dynamical system:

$$X_t(v) \leftarrow \text{Sample}\left(P^v_T, \mathcal{R}_{(v,i)}\right), \quad (4)$$

where $\mathcal{R}_{(v,i)}$ denotes the random seed used for the update $(v, i)$. The function $\text{Sample}(\mu, \mathcal{R})$ is a deterministic subroutine that, given the description of any distribution $\mu$ over $Q$ and access to the random seed $\mathcal{R}$, guarantees to return a spin distributed according to $\mu$ by utilizing the random bits in $\mathcal{R}$.

Given the initial configuration $X_0 \in Q^V$, once the update schedule $\mathcal{S} = (t^v_i)_{v \in V, 0 \leq i \leq m_v}$ up to time $T$ and the random bits $\mathcal{R} = (\mathcal{R}_{(v,i)})_{v \in V, 1 \leq i \leq m_v}$ used in all updates are generated, the entire evolution of the chain $(X_t)_{0 \leq t \leq T}$ is uniquely determined by sequentially executing (4).

A parallel procedure that faithfully simulates this process is given in Algorithm 1.

**Remark 3.1** (consistent iterative updating). The algorithm maintains an array $\hat{X}^{(\ell)}$, where each entry $\hat{X}^{(\ell)}_v[i]$ corresponds to the update $(v, i)$ in the chain. In every iteration $\ell \geq 1$, each entry $\hat{X}^{(\ell)}_v[i]$ is updated from $\hat{X}^{(\ell-1)}$ according to the rule (4), using the same assignment of random bits $\mathcal{R}_{(v,i)}$ regardless of the iteration $\ell$.

Algorithm 1 is a locally iterative parallel algorithm that simulates the abstract dynamical system described in (4), utilizing coupled randomness as noted in Remark 3.1 to accelerate convergence while ensuring accurate simulation. Unlike the belief propagation (BP) algorithm used to calculate marginal probabilities, whose correctness is not guaranteed on general graphs [MM09], Algorithm 1 always converges to the correct chain within a finite number of steps. This guarantee is based on the following two observations:


**Algorithm 1:** Locally-iterative algorithm for simulating single-site dynamics

**Input:** initial configuration \(X_0 \in Q^V\); update schedule \(\Sigma = (t^i_v)_{v \in V, 0 \leq i \leq m_v}\); assignment \(\mathfrak{R} = (R_{(v,i)})_{v \in V, 1 \leq i \leq m_v}\) of random bits for resolving updates.

1. initialize \(\ell \leftarrow 0\) and \(\tilde{X}^{(0)}_v[i] \leftarrow X_0(v)\) for all \(v \in V, 0 \leq i \leq m_v\);
2. repeat
   1. \(\ell \leftarrow \ell + 1\);
   2. \(\forall v \in V\) in parallel do \(\tilde{X}^{(\ell)}_v[0] \leftarrow X_0(v)\);
   3. \(\forall v \in V\), updates \((v,i), v \in V, 1 \leq i \leq m_v\), in parallel do
      1. let \(\tau \in Q^{N^+_v}\) be constructed as:
      2. \(\forall u \in N^+_v, \tau_u \leftarrow \tilde{X}^{(\ell-1)}_v[j_u]\) for \(j_u = \max\{j \geq 0 \mid t^u_v < t^v_i\}\);
      3. \(\tilde{X}^{(\ell)}_v[i] \leftarrow \text{Sample}(P^*_v, R_{(v,i)})\);
   4. end
3. until \(\tilde{X}^{(\ell)} = \tilde{X}^{(\ell-1)}\);

1. The abstract dynamical system in (4) is acyclic due to its monotonicity over time.
2. The correct evolution of the Markov chain \((X_t)_{0 \leq t \leq T}\) from time 0 to time \(T\) corresponds to a fixed point solution satisfying (4).

We now elaborate on these observations. Fix an update schedule \(\Sigma = (t^i_v)_{v \in V, 0 \leq i \leq m_v}\). For any updates \((u,j)\) and \((v,i)\) where \(i,j \geq 1\), define the relation:

\[(u,j) \rightarrow (v,i) \iff u \in N^+_v \land j = \max\{j \geq 0 \mid t^u_j < t^v_i\}.\]  

(5)

Intuitively, \((u,j) \rightarrow (v,i)\) means that the update \((v,i)\) is determined based on the outcome of update \((u,j)\). This relation \(\rightarrow\) between updates naturally defines a directed graph \(D\rightarrow = (U, E\rightarrow)\), where the vertices are \(U = \{(v,i) \mid v \in V, 1 \leq i \leq m_v\}\) and the edges are:

\[E\rightarrow = \{(u,j), (v,i) \mid (u,j), (v,i) \in U, (u,j) \rightarrow (v,i)\}.\]  

(6)

It is easy to verify that \(D\rightarrow\) is a directed acyclic graph (DAG) due to the monotonicity in time \(t^i_v\). We refer to \(D\rightarrow\) as the dependency graph, which depicts the dependencies between updates.

The following lemma is established through structural induction on the dependency graph \(D\rightarrow\).

**Lemma 3.2** (convergence & correctness). Fix any initial configuration \(X_0 \in Q^V\), update schedule \(\Sigma = (t^i_v)_{v \in V, 0 \leq i \leq m_v}\) up to time \(T\), and assignment of random bits \(\mathfrak{R} = (R_{(v,i)})_{v \in V, 1 \leq i \leq m_v}\) for resolving updates. Let \((X_t)_{0 \leq t \leq T}\) be the continuous-time chain fully determined by \(X_0, \Sigma\) and \(\mathfrak{R}\).

1. **Algorithm 1** terminates within \(m + 1\) iterations (of the repeat loop), where \(m = \sum_{v \in V} m_v\).
2. Upon termination, \(\tilde{X}\) correctly gives all transitions of the chain \((X_t)_{0 \leq t \leq T}\), such that \(\tilde{X}[v][i] = X_{t^i_v}(v)\) for every update \((v,i)\). In particular, \((\tilde{X}_v[m_v])_{v \in V}\) gives the configuration at time \(T\).

**Fast convergence.** For a random pair \((\Sigma, \mathfrak{R})\) generated as in a continuous-time chain \((X_t)_{0 \leq t \leq T}\) up to time \(T\), calculations from [PHY21a, HS07] show that, with high probability, the length of any path in the dependency graph \(D\rightarrow\) is bounded by \(O(\Delta T + \log n)\), where \(\Delta\) is the maximum degree of the underlying graph \(G\). Consequently, **Algorithm 1** returns within \(O(\Delta T + \log n)\) iterations with high probability.
Indeed, Algorithm 1 can converge to the correct fixpoint significantly faster than the length of the longest path in $D_\tau$. To understand this, recall that in Line 7 of Algorithm 1 the array $\hat{X}$ is updated as follows:

$$\hat{X}_\tau[i] \leftarrow \text{Sample} \left( P^\tau_{\tau'}, R_{\tau,\tau'} \right).$$

Here, the entry $\hat{X}_\tau[i]$ (which corresponds to the update $(\tau, i)$) can locally stabilize (i.e. remains the same in two consecutive iterations) even before its neighborhood configuration $\tau$ has stabilized, under the following conditions:

1. The distributions $P^\tau_{\tau'}, P^\tau_{\tau'}$ are close to each other when their boundary conditions $\tau, \tau' \in Q^{N_\tau}$ are similar.

2. For $P^\tau_{\tau'}, P^\tau_{\tau'}$ that are close to each other, it is likely that $\text{Sample}(P^\tau_{\tau'}, R_{\tau,\tau'}) = \text{Sample}(P^\tau_{\tau'}, R_{\tau,\tau'})$.

The first property is captured by Condition 1 and the second is formalized by the following notion of $\alpha$-competitiveness:

**Definition 3.3** ($\alpha$-competitiveness). Let $\alpha \geq 1$. A procedure $\text{Sample}(\mu, R)$ that returns a sample from distribution $\mu$ arbitrarily specified over $Q$, is said to be $\alpha$-competitive, if for any $\mu, \nu$ over $Q$,

$$\Pr \left[ \text{Sample}(\mu, R) \neq \text{Sample}(\nu, R) \right] \leq \alpha \cdot d_{TV}(\mu, \nu).$$

The following convergence bound for Algorithm 1 on random input justifies the above intuition. On random choices $(\Xi, R)$ generated as in the continuous-time chain $(X_t)_{0 \leq t \leq T}$ up to time $T$, Algorithm 1 faithfully simulates the chain using $O(T + \log n)$ iterations with high probability, assuming Condition 1 with parameter $\rho = O(1)$ and an $O(1)$-competitive Sample subroutine.

**Lemma 3.4** (fast convergence with linear speedup). If Condition 1 holds with parameter $\rho$ and the Sample subroutine is $\alpha$-competitive, then for every $T > 0$ and $\epsilon \in (0, 1)$, Algorithm 1 on any initial configuration $X_0 \in Q^V$ and random $(\Xi, R)$ up to time $T$ terminates within $O\left(\alpha \rho \cdot T + 2\log \left(\frac{nT}{\epsilon} \right)\right)$ iterations with probability $\geq 1 - \epsilon$.

Furthermore, the following lemma gives a much improved bound on the convergence rate (with an exponential speedup) for Algorithm 1 under a more restricted condition $\alpha \rho < 1$.

**Lemma 3.5** (fast convergence with exponential speedup). Assume the condition of Lemma 3.4. If further $\alpha \rho < 1$, then for every $T > 0$ and $\epsilon \in (0, 1)$, Algorithm 1 on any initial configuration $X_0 \in Q^V$ and random $(\Xi, R)$ up to time $T$ terminates within $O\left(\frac{1}{1-\alpha \rho} \log \left(\frac{nT}{\epsilon} \right)\right)$ iterations with probability $\geq 1 - \epsilon$.

**Universal coupling.** It remains to investigate whether there exists a Sample procedure that always achieves small competitiveness when sampling from an arbitrarily specified distribution $\mu$ over $Q$. Such a sampling procedure would provide a universal coupling, simultaneously coupling all distributions $\mu$ over the same sample space $Q$.

We first show that for Boolean domain $Q$ of size $|Q| = 2$, the standard inverse transform sampling provides a 1-competitive Sample subroutine that can perfectly couple all pairs of distributions $\mu$ over $Q$.

**Definition 3.6** (inverse transform sampling). Let $R$ be uniformly distributed over $[0, 1]$. For any distribution $\mu$ over the Boolean domain $Q = \{0, 1\}$, define

$$\text{Sample}(\mu, R) = \begin{cases} 0 & \mu < \mu(0), \\ 1 & \mu \geq \mu(0). \end{cases}$$

(8)
This Sample(μ, ℛ) implements the inverse transform sampling method for the Bernoulli distribution μ, where the uniform random seed ℛ ∈ [0, 1] is used as the quantile.

It is straightforward to verify that Sample(μ, ℛ) is distributed as μ, and for any distributions μ, ν over Q = {0, 1},

\[ \Pr_{\mathcal{R}}[\text{Sample(μ, ℛ)} \neq \text{Sample(ν, ℛ)}] = d_{TV}(μ, ν). \]

This confirms that the Sample(μ, ℛ) procedure as defined in (5) for sampling from Boolean domain is 1-competitive. As a result, we can state the following corollary of Lemma 3.5 for Boolean domains.

**Corollary 3.7.** If |Q| = 2 and Condition 7 holds with parameter ρ < 1, then for every T > 0 and ε ∈ (0, 1), Algorithm 1 when applied to any initial configuration X_0 ∈ Q^V and random (Ξ, Θ) up to time T, terminates within \( O\left(\frac{1}{1-\rho} \log \left(\frac{nT}{\epsilon}\right)\right) \) iterations with probability at least 1 − ε.

For general finite non-Boolean domains Q, the competitive ratio achieved by the inverse transform sampling method increases linearly with |Q|. However, it is surprising that a universal coupling within twice the optimal coupling can always be achieved between any pair of distributions μ, ν over any finite sample space Q.

**Theorem 3.8 (twice-optimal universal coupling).** For any finite sample space Q, there exists a 2-competitive Sample(μ, ℛ) procedure.

This sampling procedure, now known as correlated sampling, has been discovered in multiple independent works. Notably, it was identified by Kleinberg and Tardos [KT02] for rounding linear programs and by Holenstein [Hol07] for parallel repetition of 2-player 1-round games. The procedure will be formally described and analyzed in Section 4.

The rest of this section is dedicated to proving Lemma 3.2, Lemma 3.4, and Lemma 3.5.

### 3.1 Worst-case Convergence and Correctness

The dependency graph \( D \rightarrow \) defined in (6) gives rise to a notion of depth for updates. For each update \((v, i) \in U\), its depth, denoted by depth(v, i), is defined as follows:

\[
\text{depth}(v, i) = \begin{cases} 
1 + \max_{(u, j) \rightarrow (v, i)} \text{depth}(u, j) & \text{if } \neg \exists (u, j) \text{ s.t. } (u, j) \rightarrow (v, i), \\
1 & \text{otherwise.}
\end{cases}
\]

Note that depth(v, i) represents the length of the longest path that ends at (v, i) in the DAG D →.

Now consider the \( \hat{X} \) maintained by Algorithm 1. Note that for every update \((v, i)\) and any possible iteration number \( \ell \geq 0 \), the entry \( \hat{X}^{(\ell)}[i] \) is assigned only once in Line 7 of Algorithm 1. For any \( \ell \geq 1 \) where the algorithm terminates before reaching the \( \ell \)-th iteration, we assume \( X^{(\ell)} = X^{(\ell-1)} \). Therefore, \( \hat{X}^{(\ell)} \) is well-defined for all \( \ell \geq 0 \), given the input \( X_0, \Xi \) and \( \Theta \).

The next lemma shows that for every update \((v, i)\), the corresponding entry \( \hat{X}^{(\ell)}[i] \) stops changing, i.e. it locally terminates, after depth(v, i) iterations.

**Lemma 3.9.** For each update \((v, i)\), where \( v \in V \) and \( 1 \leq i \leq m_v \), for any \( \ell > \text{depth}(v, i) \), it holds that

\[ \hat{X}^{(\ell)}[i] = \hat{X}^{(\ell-1)}[i]. \]

This immediately proves Item 1 of Lemma 3.2, establishing the convergence of Algorithm 1 within \( m + 1 \) steps. Since \( m = \sum_{v \in V} m_v = |U| \) provides a trivial upper bound on the length of any path in \( D \rightarrow = (U, E \rightarrow) \), Lemma 3.9 implies that the entire \( \hat{X} \) must have stopped changing after at most \( m + 1 \) steps.
\begin{proof}[Proof of Lemma 3.9] A key observation is that in the \( r \)-th iteration, the neighborhood configuration \( \tau \in Q^{N_v} \) constructed in Line 6 for resolving an update \((v, i)\) satisfies the following: for every \( u \in N_v^+ \), \( \tau_u = \tilde{X}^{(r-1)}_{u}[j_u] \), where either \((u, j_u) \rightarrow (v, i)\) (and hence \( \text{depth}(u, j_u) \leq \text{depth}(v, i) \)) or \( j_u = 0 \). Intuitively, this observation indicates that every spin in the neighborhood configuration \( \tau \) for resolving an update \((v, i)\) either results from a previous update or is part of the initial configuration.

We now prove the lemma by induction on depth.

**Induction Basis:** For the updates \((v, i)\) with \( \text{depth}(v, i) = 1 \), there is no \((u, j)\) such that \((u, j) \rightarrow (v, i)\). Therefore, by the above observation, the neighborhood configuration \( \tau \in Q^{N_v^+} \) constructed in Line 6 for resolving the update \((v, i)\) always satisfies \( \tau_u = \tilde{X}^{(r-1)}_{u}[0] = X_0(u) \) for every \( u \in N_v^+ \). This implies that \( \tilde{X}^{(r)}_v[i] \leftarrow \text{Sample} \left(P^r_v, \mathcal{R}_{(v,i)} \right) \) stops changing after the first iteration, proving the lemma for \( \text{depth}(v, i) = 1 \).

**Induction Step:** Now, assume the lemma holds for all updates \((u, j)\) with \( \text{depth}(u, j) < \text{depth}(v, i) \), and consider an update \((v, i)\) with \( \text{depth}(v, i) > 1 \). By the same observation and the induction hypothesis, the neighborhood configuration \( \tau \in Q^{N_v^+} \) constructed in Line 6 for resolving the update \((v, i)\) stops changing after \( \text{depth}(v, i) - 1 \) iterations. Consequently, \( \tilde{X}^{(r)}_v[i] \leftarrow \text{Sample} \left(P^r_v, \mathcal{R}_{(v,i)} \right) \) stops changing after \( \text{depth}(v, i) \) iterations, finishing the induction. \( \square \)

We now prove Item 2 of Lemma 3.2, which addresses the correctness of Algorithm 1. Let \( \tilde{X} \) denote the \( \tilde{X}^{(r)} \) in Algorithm 1 when the algorithm terminates. Recall that \((X_t)_{0 \leq t \leq T}\) represents the continuous-time chain, which is fully determined by \(X_0, \mathcal{S}\) and \(\mathcal{H}\). Item 2 of Lemma 3.2 can be restated as follows:

**Lemma 3.10.** For each update \((v, i)\), where \( v \in V \) and \( 1 \leq i \leq m_v \), it holds that
\[ \tilde{X}_v[i] = X_t^i(v). \]

**Proof.** As the fixpoint of Algorithm 1, the array \( \tilde{X} \) satisfies that for every update \((v, i)\) where \( v \in V \) and \( 1 \leq i \leq m_v \),
\[ \tilde{X}_v[i] = \text{Sample} \left(P^r_v, \mathcal{R}_{(v,i)} \right), \]
where \( \tau \in Q^{N_v^+} \) is such that for every \( u \in N_v^+ \), \( \tau_u = \tilde{X}_u[j_u] \) for \( j_u = \max \{ j \geq 0 \mid t^u_i < t^v_j \} \).

By definition of the continuous-time chain \((X_t)_{0 \leq t \leq T}\), for every update \((v, i)\) where \( v \in V \) and \( 1 \leq i \leq m_v \),
\[ X_t^i(v) = \text{Sample} \left(P^r_v, \mathcal{R}_{(v,i)} \right), \]
where \( \tau \in Q^{N_v^+} \) is such that for every \( u \in N_v^+ \), \( \tau_u = X_{t^u_i}'(u) \) for \( j_u = \max \{ j \geq 0 \mid t^u_i < t^v_j \} \).

Additionally, \( \tilde{X}_v[0] = X_0(v) \) for all \( v \in V \). The lemma follows by inductively verifying the equality \( \tilde{X}_v[i] = X_t^i(v) \) through the topological order of \( D_{\supseteq} \) defined in (6). \( \square \)

### 3.2 Fast Average-case Convergence

In a continuous-time chain \((X_t)_{t \in \mathbb{R}_{\geq 0}}\), the update schedule \( \mathcal{S} \) is randomly generated by independent Poisson clocks. Algorithm 1 terminates in significantly fewer steps with such a randomly generated update schedule.

**Lemma 3.11** ([FHY21a], [HS07]). For a random update schedule \( \mathcal{S} \) where \( \mathcal{S} = (T^v_i)_{v \in V, 0 \leq i \leq m_v} \) is generated up to time \( T \) by \( n \) independent rate-1 Poisson clocks on graph \( G \) of maximum degree \( \Delta \), the probability that there exists an update \((v, i)\) with \( \text{depth}(v, i) \geq \ell \) is at most \( n \cdot \left( e^{(\Delta+1)\ell} \right)^{\ell} \) for any \( \ell \geq 1 \).
Proof. Fix any path \((v_1, \ldots, v_\ell) \in V^\ell\) where \(v_{i+1} \in N^+_v\) for \(1 \leq i < \ell\). Consider the event where there exist \(0 < t_1 < \cdots < t_\ell < T\) such that the Poisson clock at \(v_i\) rings at time \(t_i\). As shown in [HS07, Observation 3.2], the probability of this event is at most \(\left(\frac{e\Delta}{T}\right)^\ell\). By applying the union bound over all possible paths, where there are at most \(n(\Delta + 1)\) such paths, the probability that there is a path with this property is bounded by \(n \cdot \left(\frac{e(\Delta + 1)T}{T}\right)^\ell\). 

Combining Lemmas 3.9 and 3.11, we conclude that for a randomly generated update schedule \(\Xi\) up to time \(T\), Algorithm 1 will terminate within \(\left[2e(\Delta + 1)T + \log_2 \left(\frac{n}{\epsilon}\right)\right]\) iterations with probability at least \(1 - \epsilon\).

We now prove Lemma 3.4 and Lemma 3.5 establishing the faster convergence of Algorithm 1 when the Dobrushin’s influence matrix has a bounded operator norm. We will actually prove slightly stronger results, for which we first need to introduce the following variant of Dobrushin’s criterion.

Definition 3.12 (Dobrushin’s influence matrix for sampling). Given a single-site dynamics on a graph \(G = (V, E)\) specified by \(\{P_v\}\) and realized by a \(\text{Sample}(\cdot, \cdot)\) subroutine, the Dobrushin’s influence matrix for sampling is a matrix \(S \in \mathbb{R}^{V_xV}\) defined by:

\[
\forall u, v \in V: \quad S(u, v) \triangleq \max_{\tau, \tau' : \tau \lor \tau' \subseteq \{u\}} \Pr \left[\text{Sample}(P^T_v, \mathcal{R}) \neq \text{Sample}(P^T_v, \mathcal{R})\right],
\]

where the maximum is taken over all pairs of configurations \(\tau, \tau' \in Q^{N^+_v}\) on the inclusive neighborhood \(N^+_v = N_v \cup \{v\}\) of \(v\), such that \(\tau\) and \(\tau'\) agree with each other everywhere except at \(u\).

We also define the following variant of Condition 1, tailored for the Dobrushin’s influence matrix for sampling.

Condition 2. There is a \(p \in [1, \infty]\) such that \(\|S\|_p \leq \theta\) for some \(\theta > 0\).

It is straightforward to observe that \(S \leq \alpha \rho\) entry-wise, assuming that the \(\text{Sample}(\cdot, \cdot)\) subroutine is \(\alpha\)-competitive (as defined in Definition 3.3), where \(\rho\) denotes the Dobrushin’s influence matrix (as defined in Definition 1.1). Consequently, we have \(\|S\|_p \leq \alpha \|\rho\|_p\) for any \(p \in [1, \infty]\), since \(S\) and \(\rho\) are non-negative matrices and \(S \leq \alpha \rho\) entry-wise.

Then, Lemma 3.4 and Lemma 3.5 are implied by the following lemma.

Lemma 3.13 (fast convergence of Algorithm 1). The following results hold for Algorithm 1 on any initial configuration \(X_0 \in Q^V\) and random \((\Xi, \mathcal{R})\) up to time \(T\):

1. (linear speedup) If Condition 2 holds with parameter \(\theta\), then for any \(\epsilon > 0\), Algorithm 1 terminates within \(O \left(\theta \cdot T + \log \left(\frac{n}{\epsilon}\right)\right)\) iterations with probability at least \(1 - \epsilon\).

2. (exponential speedup) If Condition 2 holds with parameter \(\theta < 1\), then for any \(\epsilon > 0\), Algorithm 1 terminates within \(O \left(\frac{T}{1 - \theta} \log \left(\frac{n}{\epsilon}\right)\right)\) iterations with probability at least \(1 - \epsilon\).

To see that Lemma 3.4 and Lemma 3.5 are indeed consequences of Lemma 3.13, observe that Condition 2 with parameter \(\theta = \alpha\rho\) together with an \(\alpha\)-competitive \(\text{Sample}(\cdot, \cdot)\) subroutine, implies Condition 2 with parameter \(\theta = \alpha\rho\).

We now prove Lemma 3.13. Fix an update schedule \(\Xi = (t^v)_{v \in V, 0 \leq i \leq m_v}\). This defines a directed acyclic graph \(D_{\rightarrow}\) as in (6). For a chain \(\pi\) of size \(\ell\) in \(D_{\rightarrow}\):

\[\pi : (v_1, i_1) \rightarrow (v_2, i_2) \rightarrow \cdots \rightarrow (v_\ell, i_\ell),\]

define its weight as follows: For \(\ell = 1\) set \(w(\pi) \triangleq 1\); and for \(\ell > 1\), define

\[
w(\pi) \triangleq \prod_{i=1}^{\ell-1} S(v_i, v_{i+1}).
\]


Lemma 3.14. Fix an arbitrary update schedule $\mathcal{T} = (t^v_i)_{v \in V, 0 \leq i \leq m_v}$. For any update $(v, i)$, and for any $\ell \geq 1$, we have

$$\Pr \left[ \hat{X}_v^{(\ell)}[i] \neq \hat{X}_v^{(\ell-1)}[i] \right] \leq \sum_{\text{chain } \pi \text{ of size } \ell \text{ that ends at } (v,i)} w(\pi),$$

(11)

where the probability is taken over the random $\mathcal{R} = (\mathcal{R}_v^\ell)_{v \in V, 1 \leq i \leq m_v}$.

Proof. The lemma is proved by establishing the following inequality for all $\ell \geq 2$:

$$\Pr \left[ \hat{X}_v^{(\ell)}[i] \neq \hat{X}_v^{(\ell-1)}[i] \right] \leq \sum_{(u,j);(u,j)\rightarrow(v,i)} \mathcal{S}(u,v) \cdot \Pr \left[ \hat{X}_u^{(\ell-1)}[j] \neq \hat{X}_u^{(\ell-2)}[j] \right],$$

(12)

where the sum is taken over all updates $(u, j)$ in $\mathcal{T}$, where $u \in V, 1 \leq j \leq m_u$ such that $(u, j) \rightarrow (v, i)$.

The lemma follows from (12) by induction. Induction Basis: For $\ell = 1$, the right hand side of (11) becomes $w((v, i)) = 1$. Thus, $\Pr \left[ \hat{X}_v^{(1)}[i] \neq \hat{X}_v^{(0)}[i] \right] \leq 1$, and hence (11) holds. Induction Step: Assume that (11) holds for chains of size $\ell - 1$. To prove (11) for chains of size $\ell$, consider (12). We have:

$$\Pr \left[ \hat{X}_v^{(\ell)}[i] \neq \hat{X}_v^{(\ell-1)}[i] \right] \leq \sum_{(u,j);(u,j)\rightarrow(v,i)} \mathcal{S}(u,v) \sum_{\text{chain } \pi \text{ of size } \ell - 1 \text{ that ends at } (u,j)} w(\pi)$$

$$\leq \sum_{\text{chain } \pi \text{ of size } \ell \text{ that ends at } (v,i)} w(\pi).$$

This completes the induction and proves the lemma. It remains to prove (12).

For any $\ell \geq 1$, the variable $\hat{X}_v^{(\ell)}[i]$ is updated in Line 7 of Algorithm 1. Specifically, it is assigned as:

$$\hat{X}_v^{(\ell)}[i] \leftarrow \text{Sample} \left( P_v^{x(\ell)}, \mathcal{R}_{(v,i)} \right),$$

where $\tau^{(\ell)} \in \mathcal{Q}_{N_v^+}$ is constructed such that for each $u \in N_v^+$:

$$\tau^{(\ell)}_u = \hat{X}_u^{(\ell-1)}[j_u], \quad \text{where } j_u \triangleq \max\{j \geq 0 \mid t_j^u < t_i^v\}.$$  

(13)

This means that $\tau^{(\ell)}$ is constructed based on the most recent update time before $t_i^v$ for each neighbor $u$ of $v$.

We use $\mathcal{R}_{<t}$ to denote the random bits used for resolving updates before time $t$. Formally, we define:

$$\mathcal{R}_{<t} \triangleq (\mathcal{R}_{(v,i)})_{v \in V, 1 \leq i \leq m_v, t^v_i < t}.$$  

Observe that for any $\ell \geq 0$, the value of $\hat{X}_v^{(\ell)}[i]$ is fully determined by $\mathcal{R}_{<t}$ if $t^v_i < t$.

Let $\tau^{(\ell)}$ and $\tau^{(\ell-1)}$ be constructed as in (13). Both $\tau^{(\ell)}$ and $\tau^{(\ell-1)}$ are determined by $\mathcal{R}_{<t}$ and are independent of the random choice of $\mathcal{R}_{(v,i)}$. Consider the vertices in $N_v^+$, denoted as $u_1, u_2, \ldots, u_k$, where $k = |N_v^+|$. We define a sequence of configurations $\tau_0, \tau_1, \ldots, \tau_k \in \mathcal{Q}_{N_v^+}$ that transitions from $\tau_0 = \tau^{(\ell-1)}$ to $\tau_k = \tau^{(\ell)}$ by modifying one spin at a time. Specifically, for each $1 \leq i \leq k$, $\tau_i$ is obtained by changing the spin of $u_i$ in $\tau_{i-1}$ from $\tau_{(\ell-1)}^{u_i}$ to $\tau^{(\ell)}_{u_i}$. Note that $\tau_{i-1}$ and $\tau_i$ might identical if $\tau_{(\ell-1)}^{u_i} = \tau^{(\ell)}_{u_i}$. 
By the triangle inequality, we have:

\[
\begin{align*}
\Pr \left[ \hat{X}_v^{(t)}[i] \neq \hat{X}_v^{(t-1)}[i] \mid \mathcal{R}_{<t} \right] &= \Pr \left[ \text{Sample} \left( P_v^{(t)}, \mathcal{R}_{(v,i)} \right) \neq \text{Sample} \left( P_v^{(t-1)}, \mathcal{R}_{(v,i)} \right) \mid \mathcal{R}_{<t} \right] \\
&\leq \sum_{i=1}^{k} \Pr \left[ \tau_{i-1} \neq \tau_i \mid \mathcal{R}_{<t} \right] \Pr \left[ \text{Sample} \left( P_v^{(t)}, \mathcal{R}_{(v,i)} \right) \neq \text{Sample} \left( P_v^{(t-1)}, \mathcal{R}_{(v,i)} \right) \mid \tau_{i-1} \neq \tau_i \right] \\
&= \sum_{i=1}^{k} \Pr \left[ \tau_{i-1} \neq \tau_i \mid \mathcal{R}_{<t} \right] \Pr \left[ \text{Sample} \left( P_v^{(t)}, \mathcal{R}_{(v,i)} \right) \neq \text{Sample} \left( P_v^{(t-1)}, \mathcal{R}_{(v,i)} \right) \mid \tau_{i-1} \neq \tau_i \right] \\
&\leq \sum_{i=1}^{k} \Pr \left[ \tau_{i-1} \neq \tau_i \mid \mathcal{R}_{<t} \right] \cdot \mathcal{S}(u_i, v) \\
&= \sum_{u \in N_v^+} \Pr \left[ \hat{X}_u^{(t-1)}[u] \neq \hat{X}_u^{(t-2)}[u] \mid \mathcal{R}_{<t} \right] \cdot \mathcal{S}(u, v),
\end{align*}
\]

(14)

where the last inequality follows because the sequence \(\tau^{(t-1)} = \tau_0, \tau_1, \ldots, \tau_k = \tau^{(t)}\) is constructed such that \(\tau_{i-1}\) and \(\tau_i\) \(i \in Q_{N_v^+}\) differ only at site \(u_i\) and by Definition 3.12, the probability that Sample \(\left( P_v^{(t-1)}, \mathcal{R}_{(v,i)} \right)\) differs from Sample \(\left( P_v^{(t-1)}, \mathcal{R}_{(v,i)} \right)\) is upper bounded by \(\mathcal{S}(u_i, v)\) for any pair \(\tau_{i-1}, \tau_i\) that differ only at site \(u_i\).

Observe that in (14), the sum is actually taken over all updates \((u, j_u)\) such that \((u, j_u) \rightarrow (v, i)\) (with \(j_u \geq 1\), since otherwise \(\hat{X}_u^{(t-1)}[0] = \hat{X}_u^{(t-2)}[0] = X_0(u)\)). Therefore, we have

\[
\Pr \left[ \hat{X}_v^{(t)}[i] \neq \hat{X}_v^{(t-1)}[i] \mid \mathcal{R}_{<t} \right] \leq \sum_{(u,i) : (u,i) \rightarrow (v,i)} \mathcal{S}(u, v) \cdot \Pr \left[ \hat{X}_u^{(t-1)}[j] \neq \hat{X}_u^{(t-2)}[j] \mid \mathcal{R}_{<t} \right].
\]

Then (12) follows by averaging over all \(\mathcal{R}_{<t}\) using the law of total probability. \(\square\)

Using Lemma 3.14, we can prove Lemma 3.13 which provides upper bounds on the fast convergence of Algorithm 1 by bounding the expected total weights for all chains of fixed size.

Proofs of Lemma 3.13 Let \(\mathcal{I} = (T_i)_{v \in V, \rho \leq i \leq M_v}\) be a random update schedule generated by \(n\) independent rate-1 Poisson clocks up to time \(T\). Consider the binary relation \(\rightarrow\) defined in (5) and the dependency graph \(D_{\rightarrow}\), defined in (6), both derived from this \(\mathcal{I}\).

Due to Lemma 3.14 by applying a union bound over all updates \((v, i)\) and averaging over \(\mathcal{I}\), we have

\[
\Pr \left[ \hat{X}^{(t)} \neq \hat{X}^{(t-1)} \right] \leq \mathbb{E}_\mathcal{I} \left[ \sum_{\text{chain } \pi \text{ of size } \ell} w(\pi) \right],
\]

(15)

where the weight function \(w(\cdot)\) is defined in (10).

It remains to upper bound the expected total weight referred to in (15). Let \(M = \sum_{v \in V} M_v\) denote the total number of updates in the schedule \(\mathcal{I} = (T_i)_{v \in V, \rho \leq i \leq M_v}\). It follows the Poisson distribution with mean \(nT\), which is given by:

\[
\forall m \geq 0, \quad \Pr[M = m] = e^{-nT} \frac{(nT)^m}{m!}.
\]

Conditioning on \(M = m\) for a fixed integer \(m \geq 1\), we can sort all \(m\) updates according to their respective times as: \(0 < T_{U_1} < T_{U_2} < \cdots < T_{U_m} < T\). This defines a random sequence of updates:

\[(U_1, I_1), (U_2, I_2), \ldots, (U_m, I_m) \in V \times \{1, 2, \ldots, m\},\]

(16)
where \((U_i, I_i)\) represents the \(i\)-th update in global time order. Due to the independence and symmetry of the Poisson clocks, the sequence \(U = (U_1, \ldots, U_m) \in V^m\) is uniformly distributed.

Given a sequence \(U = (U_1, \ldots, U_m) \in V^m\) of fixed length \(m\), for any increasing subsequence \(1 \leq j_1 < \cdots < j_l \leq m\) with \(1 \leq l \leq m\), define:

\[
W(j_1, \ldots, j_l) = W_U(j_1, \ldots, j_l) \triangleq \prod_{i=1}^{l} S(U_{j_i}, U_{j_i+1}).
\]  

(16)

For a random schedule \(\mathcal{T} = (T^0_i)_{\nu \in V, 0 \leq i \leq M}\) conditioned on \(M = m\), and the sequence \(U = (U_1, \ldots, U_m)\) of updated sites determined by \(\mathcal{T}\), it is straightforward to verify that for every \(1 \leq l \leq m\),

\[
\sum_{\text{chain } \pi \text{ of size } l} w(\pi) \leq \sum_{1 \leq j_1 < \cdots < j_l \leq m} W_U(j_1, \ldots, j_l).
\]  

(17)

Let \(U = (U_1, U_2, \ldots, U_m) \in V^m\) be chosen uniformly at random. For any \(1 \leq j_1 < \cdots < j_l \leq m\) and any \(v_1, \ldots, v_l \in V\), the probability that \((U_{j_1}, \ldots, U_{j_l}) = (v_1, \ldots, v_l)\) is \(\frac{1}{n^l}\). Thus, for any \(\frac{1}{p} + \frac{1}{q} = 1\),

\[
\mathbb{E}[W_U(j_1, \ldots, j_l)] = \frac{1}{n^l} \sum_{v_1, \ldots, v_l \in V} \prod_{i=1}^{l-1} S(v_i, v_{i+1}) = \frac{1}{n^l} \|S^{\ell-1}1\|_p
\]

(Hölder’s inequality)

\[
\leq \frac{n^{1/q}}{n^l} \|S^{\ell-1}1\|_p
\]

\[
\leq \frac{n^{1/q}}{n^l} \|S\|_p^{\ell-1} \|1\|_p
\]

\[
= \frac{1}{n^{\ell-1}} \|S\|_p^{\ell-1}.
\]  

(18)

By Condition 1, it holds that \(\|S\|_p \leq \theta\) for some \(p \in [1, \infty]\). Hence \(\mathbb{E}[W_U(j_1, \ldots, j_l)] \leq \left(\frac{\theta}{n}\right)^{\ell-1}\). Combining with (17), for every \(1 \leq l \leq m\),

\[
\mathbb{E}_{\mathcal{T}} \left[ \sum_{\text{chain } \pi \text{ of size } l} w(\pi) \bigg| M = m \right] \leq \mathbb{E}_{U \in V^m} \left[ \sum_{1 \leq j_1 < \cdots < j_l \leq m} W_U(j_1, \ldots, j_l) \right]
\]

\[
\leq \left(\frac{m}{\ell}\right) \left(\frac{\theta}{n}\right)^{\ell-1}.
\]  

(19)

Combined with (15), for any \(\ell \geq 1\), the probability that Algorithm 1 does not terminate within \(\ell \geq 1\) iterations is:

\[
\Pr \left[ \hat{X}(\ell) \neq \hat{X}(\ell-1) \right] \leq \sum_{m \geq 0} \Pr[M = m] \cdot \mathbb{E}_{\mathcal{T}} \left[ \sum_{\text{chain } \pi \text{ of size } l} w(\pi) \bigg| M = m \right]
\]

\[
\leq \sum_{m \geq \ell} \frac{e^{-\pi T(nT)^m}}{m!} \left(\frac{m}{\ell}\right) \left(\frac{\theta}{n}\right)^{\ell-1}
\]

(take \(k = m - \ell\))

\[
= \frac{nT}{\ell!} (\theta T)^{\ell-1} \sum_{k \geq 0} \frac{e^{-\pi T(nT)^k}}{k!}
\]

\[
= \frac{nT}{\ell!} \left(\frac{e\theta T}{\ell - 1}\right)^{\ell-1}
\]

\[
((\ell - 1)! \geq \left(\frac{\ell-1}{e}\right)^{\ell-1})
\]

\[
\leq \frac{nT}{\ell} \left(\frac{e\theta T}{\ell - 1}\right)^{\ell-1}.
\]
To ensure this is at most $\epsilon$, set $\ell = \lceil (2e\theta + 1) T + \log_2 \left( \frac{m}{n} \right) + 1 \rceil$. This proves Item 1 of Lemma 3.13.

Next, to prove Item 2 of Lemma 3.13, consider a more refined upper bound than (17):

$$
\sum_{\text{chain } \pi \text{ of size } \ell} w(\pi) \leq \sum_{1 \leq j_1 < \ldots < j_\ell \leq m} I \left[ (U_{j_1}, I_{j_1}) \to \cdots \to (U_{j_\ell}, I_{j_\ell}) \right] \cdot W_U(j_1, \ldots, j_\ell). \tag{20}
$$

Thus, for every $1 \leq \ell \leq m$, we have

$$
\mathbb{E}_\pi \left[ \sum_{\text{chain } \pi \text{ of size } \ell} w(\pi) \bigg| M = m \right] \leq \sum_{1 \leq j_1 < \ldots < j_\ell \leq m} \sum_{(v_1, \ldots, v_\ell) \in V^\ell} \Pr \left[ (U_{j_1}, I_{j_1}) \to \cdots \to (U_{j_\ell}, I_{j_\ell}) \right] \cdot \mathbb{E}_U \left[ W_U(j_1, \ldots, j_\ell) \right] \cdot W_{(v_1, \ldots, v_\ell)}(j_1, \ldots, j_\ell).
$$

Here, the inequality follows from (20), and the equations are due to the total expectation and total probability.

Let $U = (U_1, U_2, \ldots, U_m) \in V^m$ be chosen uniformly at random. For any $1 \leq i < j \leq m$, we have $(U_i, I_i) \to (U_j, I_j)$ only if $U_k \neq U_i$ for all $i < k < j$. Therefore, for any $1 \leq j_1 < \cdots < j_\ell \leq m$ and $(v_1, v_2, \ldots, v_\ell) \in V^\ell$, we have

$$
\Pr \left[ (U_{j_1}, I_{j_1}) \to \cdots \to (U_{j_\ell}, I_{j_\ell}) \right] \leq \prod_{i=1}^{\ell-1} \left( 1 - \frac{1}{n} \right)^{j_i - j_{i-1} - 1} = \left( 1 - \frac{1}{n} \right)^{j_\ell - j_1 - (\ell - 1)}.
$$

By Condition 1 combined with (18), (21) and (22), we obtain that for every $1 \leq \ell \leq m$,

$$
\mathbb{E}_\pi \left[ \sum_{\text{chain } \pi \text{ of size } \ell} w(\pi) \bigg| M = m \right] \leq \sum_{1 \leq j_1 < \ldots < j_\ell \leq m} \sum_{(v_1, \ldots, v_\ell) \in V^\ell} \Pr \left[ (U_{j_1}, I_{j_1}) \to \cdots \to (U_{j_\ell}, I_{j_\ell}) \right] \cdot \mathbb{E}_U \left[ W_U(j_1, \ldots, j_\ell) \right] \cdot W_{(v_1, \ldots, v_\ell)}(j_1, \ldots, j_\ell) \leq \sum_{1 \leq j_1 < \ldots < j_\ell \leq m} \left( 1 - \frac{1}{n} \right)^{j_\ell - j_1 - (\ell - 1)} \sum_{(v_1, \ldots, v_\ell) \in V^\ell} \Pr \left[ (U_{j_1}, \ldots, U_{j_\ell}) = (v_1, \ldots, v_\ell) \right] \cdot W_{(v_1, \ldots, v_\ell)}(j_1, \ldots, j_\ell) \leq \sum_{1 \leq j_1 < \ldots < j_\ell \leq m} \left( 1 - \frac{1}{n} \right)^{j_\ell - j_1 - (\ell - 1)} \mathbb{E}_U \left[ W_U(j_1, \ldots, j_\ell) \right] \leq \sum_{1 \leq j_1 < \ldots < j_\ell \leq m} \left( 1 - \frac{1}{n} \right)^{j_\ell - j_1 - (\ell - 1)} \left( \frac{\|S\|_F^\ell}{n} \right) \leq \sum_{1 \leq j_1 < \ldots < j_\ell \leq m} \left( 1 - \frac{1}{n} \right)^{j_\ell - j_1 - (\ell - 1)} \left( \frac{\theta}{n} \right)^{\ell - 1}.
$$
If $\theta < 1$, then combined with (15), we have
\[
\Pr \left[ \hat{X}^{(\ell)} \neq \hat{X}^{(\ell-1)} \right] \leq \sum_{m \geq 0} \Pr[M = m] \cdot \mathbb{E}_{\pi} \left[ \sum_{\text{chain } \pi \text{ of size } \ell} w(\pi) \mid M = m \right]
\]
\[
\leq \sum_{m \geq 0} \Pr[M = m] \cdot \sum_{1 \leq j_1 < \ldots < j_\ell \leq m} \left( 1 - \frac{1}{n} \right)^{j_\ell - j_1 - (\ell-1)} \left( \frac{\theta}{n} \right)^{\ell-1}
\]
(take $k_i = j_{i+1} - j_i - 1$)
\[
\leq \sum_{m \geq 0} \Pr[M = m] \cdot \sum_{1 \leq j_1 \leq m} \sum_{k_1, k_2, \ldots, k_{\ell-1} \geq 0} \left( 1 - \frac{1}{n} \right)^{\sum_{i=1}^{\ell-1} k_i} \left( \frac{\theta}{n} \right)^{\ell-1}
\]
\[
\leq \sum_{m \geq 0} \Pr[M = m] \cdot mn^{\ell-1} \left( \frac{\theta}{n} \right)^{\ell-1}
\]
\[
= \mathbb{E}[m]n^{\ell-1}
\]
\[
= nT\theta^{\ell-1}.
\]
Taking $\ell = \left\lceil \frac{1}{\theta \ln (\frac{nT}{\epsilon})} \right\rceil + 1$, the probability that Algorithm 1 does not terminate within $\ell \geq 1$ iterations is at most $\epsilon$. This proves Item 2 of Lemma 3.13.

## 4 Universal Coupling via Correlated Sampling

In this section, we present the universal coupling claimed in Theorem 3.8.

Let $\Delta(\Omega)$ denote the probability simplex on the sample space $\Omega$. Formally,
\[
\Delta(\Omega) \triangleq \left\{ p \in [0,1]^{\Omega} \mid \sum_{x \in \Omega} p(x) = 1 \right\}.
\]
Each $p \in \Delta(\Omega)$ represents a probability distribution over $\Omega$, where the probability of $x \in \Omega$ is $p(x)$.

**Definition 4.1** (universal coupling). A deterministic function $\text{Sample} : \Delta(\Omega) \times [0,1] \rightarrow \Omega$ is a universal coupling on sample space $\Omega$ if, when $\mathcal{R} \in [0,1]$ is chosen uniformly at random, for any distribution $p \in \Delta(\Omega)$ and $x \in \Omega$,
\[
\Pr_{\mathcal{R}} \left[ \text{Sample}(p, \mathcal{R}) = x \right] = p(x).
\]
A universal coupling simultaneously couples all distributions over $\Omega$. Our objective is to achieve a universal coupling with small competitiveness (as defined in Definition 3.3), meaning that for any pair of distributions $p, q \in \Delta(\Omega)$, the probability $\Pr[\text{Sample}(p, \mathcal{R}) \neq \text{Sample}(q, \mathcal{R})]$ should be as close as possible to the best achievable value $d_{TV}(p,q)$.

Such a universal coupling with small competitiveness can be achieved by the following sampling procedure. Note that the uniform random $\mathcal{R} \in [0,1]$ is mapped to an infinite sequence $(X_1, Y_1), (X_2, Y_2), \ldots$, where each $(X_i, Y_i) \in \Omega \times [0,1]$ is chosen uniformly and independently at random.

**Algorithm 2: Sample($p, \mathcal{R}$)**

**Input**: $p \in \Delta(\Omega)$; $\mathcal{R} = ((X_1, Y_1), (X_2, Y_2), \ldots)$, where each $(X_i, Y_i) \in \Omega \times [0,1]$.
**Output**: a sample in $\Omega$.
1. let $i^*$ be the smallest $i \geq 1$ such that $Y_i < p(X_i)$;
2. return $X_{i^*}$.
Remark 4.2. This sampling procedure has been independently discovered in various contexts to solve different problems. Notably, it was identified by Kleinberg and Tardos \cite{KT02} for rounding linear programs, and by Holenstein \cite{Hol07} to simplify Raz’s proof of parallel repetition theorem. As Charikar \cite{Cha02} pointed out, this sampling strategy generalizes Broder’s MinHash strategy \cite{Bro97} for universal coupling of uniform distributions. The optimality of such sampling strategies was further studied in \cite{BGH+20}. In that work and its follow-up studies, such as \cite{BGH+23, GKM21, STW19, KVYZ23, KKMV23, AS19}, sampling strategies satisfying Definition 4.1 were referred to as correlated sampling strategies.

The correctness, efficiency, and competitiveness of this sampling procedure are ensured by the following lemmas:

Lemma 4.3 (correctness and efficiency). For any \( p \in \Delta(\Omega) \), with random \( R = ((X_1, Y_1), (X_2, Y_2), \ldots) \) where each \( (X_i, Y_i) \in \Omega \times [0, 1] \) is chosen uniformly and independently at random:

1. For any \( x \in \Omega \), the sampling procedure returns \( x \) with probability \( p(x) \), i.e., \( \Pr[\text{Sample}(p, R) = x] = p(x) \);

2. The index \( i^* \) chosen in Algorithm 2 follows a geometric distribution a success probability of \( 1/|\Omega| \).

Lemma 4.4 (coupling performance). For any \( p, q \in \Delta(\Omega) \), with random \( R = ((X_1, Y_1), (X_2, Y_2), \ldots) \) where each \( (X_i, Y_i) \in \Omega \times [0, 1] \) is chosen uniformly and independently at random:

\[
\Pr[\text{Sample}(p, R) = \text{Sample}(q, R)] \geq \frac{1 - d_{TV}(p, q)}{1 + d_{TV}(p, q)}.
\]  

(23)

Theorem 3.8 follows immediately from Lemma 4.3 and Lemma 4.4 because it holds that

\[
\Pr[\text{Sample}(p, R) \neq \text{Sample}(q, R)] \leq \frac{2d_{TV}(p, q)}{1 + d_{TV}(p, q)} \leq 2d_{TV}(p, q).
\]

Remark 4.5. The coupled probability in (23) corresponds to the following Jaccard similarity:

\[
1 - d_{TV}(p, q) = \frac{\sum_{x \in \Omega} \min(p(x), q(x))}{\sum_{x \in \Omega} \max(p(x), q(x))} = \frac{|P \cap Q|}{|P \cup Q|}.
\]

This measures the similarity between two points \( P, Q \subseteq \Omega \times [0, 1] \) naturally constructed from the distributions \( p, q \in \Delta(\Omega) \), respectively, as: \( P \triangleq \{(x, y) \mid x \in \Omega \land y < p(x)\} \) and \( Q \triangleq \{(x, y) \mid x \in \Omega \land y < q(x)\} \). An illustration of these point sets \( P \) and \( Q \) is provided in the following figure.

A recent study \cite{BGH+20} has proven that this is essentially optimal for universal coupling.
Lemmas 4.3 and 4.4 have been established in multiple independent works [KT02, Hol07] within their respective contexts. For completeness, we provide the formal proofs of these lemmas here.

**proof of Lemma 4.3** Let \((X, Y) \in \Omega \times [0, 1]\) be chosen uniformly at random. For any function \(w : \Omega \rightarrow [0, 1]\),

\[
\Pr[Y < w(X)] = \sum_{x \in \Omega} \Pr[X = x \land Y < w(x)] = \frac{1}{|\Omega|} \sum_{x \in \Omega} w(x).
\]

(24)

In particular, for \(p \in \Delta(\Omega)\), we have \(\Pr[Y < p(X)] = 1/|\Omega|\) since \(\sum_{x \in \Omega} p(x) = 1\). Since the \((X_i, Y_i)\) pairs are independent, this proves Item 2 of the lemma: \(i^*\) follows a geometric distribution with a success probability of \(1/|\Omega|\).

For any \(p \in \Delta(\Omega)\) and \(x \in \Omega\),

\[
\Pr[X = x \mid Y < p(X)] = \frac{\Pr[X = x \land Y < p(x)]}{\Pr[Y < p(X)]} = \frac{p(x)/|\Omega|}{1/|\Omega|} = p(x),
\]

where the second equation follows from the fact that \(\Pr[Y < p(X)] = 1/|\Omega|\). Since the output of \(\text{Sample}(p, \mathcal{R})\) follows the distribution of \(X\) conditional on \(Y < p(X)\), this proves Item 1 of the lemma: \(\text{Sample}(p, \mathcal{R})\) always returns a correct sample from \(p\) for every \(p \in \Delta(\Omega)\).

**proof of Lemma 4.4** Let \((p \land q) : \Omega \rightarrow [0, 1]\) and \((p \lor q) : \Omega \rightarrow [0, 1]\) be two functions defined as:

\[
\forall x \in \Omega : \ (p \land q)(x) = \min(p(x), q(x)) \quad \text{and} \quad (p \lor q)(x) = \max(p(x), q(x)).
\]

Clearly, \((p \land q)(x) \leq (p \lor q)(x)\), and it is straightforward to verify that

\[
\sum_{x \in \Omega} (p \land q)(x) = 1 - d_{TV}(p, q) \quad \text{and} \quad \sum_{x \in \Omega} (p \lor q)(x) = 1 + d_{TV}(p, q).
\]

By (24), for \((X, Y)\) chosen uniformly at random from \(\Omega \times [0, 1]\), we have

\[
\Pr[Y < (p \land q)(X)] = \frac{1 - d_{TV}(p, q)}{|\Omega|} \quad \text{and} \quad \Pr[Y < (p \lor q)(X)] = \frac{1 + d_{TV}(p, q)}{|\Omega|},
\]

and thus,

\[
\Pr[Y < (p \land q)(X) \mid Y < (p \lor q)(X)] = \frac{\Pr[Y < (p \land q)(X)]}{\Pr[Y < (p \lor q)(X)]} = \frac{1 - d_{TV}(p, q)}{1 + d_{TV}(p, q)}. \tag{25}
\]

Let \(\mathcal{R} = ((X_1, Y_1), (X_2, Y_2), \ldots)\) be an infinite sequence where each \((X_i, Y_i) \in \Omega \times [0, 1]\) is chosen uniformly and independently at random. Define:

\[
I^*_p = \min\{i \mid Y_i < p(X_i)\} \quad \text{and} \quad I^*_q = \min\{i \mid Y_i < q(X_i)\}.
\]

Observe that \(\min\left(I^*_p, I^*_q\right) = \min\{i \mid Y_i < (p \lor q)(X_i)\}\), and

\[
I^*_p = I^*_q \implies \text{Sample}(p, \mathcal{R}) = \text{Sample}(q, \mathcal{R}). \tag{26}
\]

Then, for every \(i \geq 1\),

\[
\Pr[I^*_p = I^*_q = i \mid \min\left(I^*_p, I^*_q\right) = i] = \Pr[Y_i < (p \land q)(X_i) \land \left(\bigwedge_{j<i} Y_j \geq (p \lor q)(X_j)\right) \land Y_i < (p \lor q)(X_i) \land \left(\bigwedge_{j<i} Y_j \geq (p \lor q)(X_j)\right)]
\]

\[
= \Pr[Y_i < (p \land q)(X_i) \land Y_i < (p \lor q)(X_i)] = 1 - \frac{d_{TV}(p, q)}{1 + d_{TV}(p, q)}. \tag{27}
\]
where (27) follows from the independence of different \((X_i, Y_i)\) pairs and the fact that:

\[ Y_i \geq (p \cup q)(X_j) \implies Y_i \geq (p \cap q)(X_j), \]

and (28) follows from (25). Therefore,

\[
\begin{align*}
\Pr[I'_p = I'_q] &= \sum_{i \geq 1} \Pr[\min(I'_p, I'_q) = i] \cdot \Pr[I'_p = I'_q = i | \min(I'_p, I'_q) = i] \\
&= \frac{1 - d_{TV}(p, q)}{1 + d_{TV}(p, q)} \cdot \sum_{i \geq 1} \Pr[\min(I'_p, I'_q) = i] \\
&= \frac{1 - d_{TV}(p, q)}{1 + d_{TV}(p, q)}.
\end{align*}
\]

(29)

where (29) follows from (28). Finally, by the observation in (26),

\[
\Pr[\text{Sample}(p, R) = \text{Sample}(q, R)] \geq \Pr[I'_p = I'_q] = \frac{1 - d_{TV}(p, q)}{1 + d_{TV}(p, q)}.
\]

5 Parallel and Distributed Implementations

In this section, we present implementations of the algorithm for simulating single-site dynamics.

The following theorem provides a formal restatement of Theorem 1.2 and Theorem 1.3.

**Theorem 5.1.** ([Theorem 1.2] and [1.3] formally stated). Assume the existence of oracles for evaluating \(\{P^*_\}\), such that for any \(v \in V\), \(\tau \in Q^{N^d}\) and \(x \in Q\), the oracle returns the probability \(P^*_\{x\}\). There exists a CRCW-PRAM algorithm such that, given any \(X_0 \in Q^V\) on graph \(G = (V, E)\) and \(0 < T \leq n^0\), where \(n = |V|, m = |E|\) and \(q = |Q|\), the followings hold:

1. The algorithm returns a random configuration \(X \in Q^V\) that is identically distributed as \(X_T\) in the continuous-time single-site dynamics \((X_t)_{t \in R_{\geq 0}}\) specified by \(\{P^*_\}\) on \(G\) conditioned on \(X_0\).

2. (linear speedup) Assume **Condition 1** with parameter \(\rho\). With probability \(1 - n^{-c_1}\), the algorithm terminates within \(O_{c_0,c_1}(\rho \cdot (T + \log n))\) depth on \(O_{c_1}\left(\left(m + nq^2 \log^2 n\right) \log n\right)\) processors.

3. (exponential speedup) Assume \(|Q| = 2\) and **Condition 1** with parameter \(\rho < 1\). With probability \(1 - n^{-c_1}\), the algorithm terminates within \(O_{c_1}\left(\left(\frac{1}{1 - \rho} \log T + \log n\right)\right)\) depth on \(O_{c_1}(mT)\) processors.

We first describe the algorithm that works for general finite domain \(Q\) and achieves the linear speedup stated in **Item 2** of **Theorem 5.1**. For this case, we use **Algorithm 2** on sample space \(Q\) as the Sample\((\cdot, \cdot)\) subroutine used in **Line 7** of **Algorithm 1**. **Algorithm 2** can be parallelized straightforwardly. The sample space is the set \(Q\) of all spins. In each round, for given width \(w \geq 1\), the algorithm checks in parallel the next \(w\) pairs \((X_i, Y_i) \in Q \times [0, 1]\) in the sequence \((X_1, Y_1), (X_2, Y_2), \ldots\), to find the pair \((X_i, Y_i)\) with the smallest \(i\) that satisfies \(Y_i \leq p(X_i)\) (which, in CRCW-PRAM, uses \(O(1)\) depth and \(O(w^2)\) processors [KR91]) and returns the \(X_i\). The algorithm proceeds to the next round if no such pair \((X_i, Y_i)\) is found.

To avoid generating an infinite sequence of random numbers at the beginning of the algorithm, we may apply the **principle of deferred decisions**: a random number is drawn only when it is accessed by the algorithm. However, it is crucial to ensure that the randomness used by the algorithm is consistent. Specifically, the resolution of the same update \((v, i)\) in **Line 7** of **Algorithm 1** must use the same random bits \(R_{(v,i)}\) every time. The Sample subroutine is implemented using a dynamic data
while not increasing the number of parallel rounds. The final algorithm, which uses Algorithm 4 as a subroutine, is described in Algorithm 5.

It is straightforward to verify that Algorithm 3 satisfies the correctness and coupling performance stated in Lemma 4.3 and Lemma 4.4. Additionally, according to (24) and the independence between the pairs \((X_i, Y_i)\), the number of iterations of the while loop in Algorithm 3 before termination follows a geometric distribution with success probability \(1 - (1 - 1/q)^w \geq 1 - e^{-w/q}\), where \(q = |Q|\).

We now describe our parallel implementation of the algorithm that faithfully simulates a continuous-time single-site dynamics \((X_t)_{t \in \mathbb{R}_+}\) up to an arbitrarily specified time \(T > 0\), given the initial configuration \(X_0 \in Q^V\). The implementation is based on Algorithm 1 and uses Algorithm 3 as the Sample subroutine. The random choices \((\xi, \mathcal{R})\) used by the chain are generated and accessed internally by the algorithm. The detailed implementation is formally described in Algorithm 4.

### 5.1 Further Reducing the Total Work

The number of processors used in Algorithm 4 grows at least linearly with \(T\). To reduce this cost, we apply a simple optimization: divide the time interval \([0, T]\) into smaller time intervals, each of length \(O(\log n)\), and simulate the chain over these intervals sequentially. This approach allows us to effectively treat \(T\) as \(O(\log n)\) when analyzing the processor and communication costs of the algorithm, while not increasing the number of parallel rounds. The final algorithm, which uses Algorithm 4 as a subroutine, is described in Algorithm 5.

**Proof of Theorem 5.1** It is straightforward to verify that Algorithm 4 simulates the process of Algorithm 1 on the same initial configuration \(X_0\) and the same random choices of \((\xi, \mathcal{R})\) used to generate the chain \((X_t)_{0 \leq t \leq T}\). By Lemma 3.2, Algorithm 1 returns the correct sample of \(X_T\) upon termination, and so does Algorithm 4. The correctness of Algorithm 5 follows from the Markov property, thus proving Item 1 of Theorem 5.1.
Algorithm 4: Simulation \((X_0, T)\)

\[
\text{Input : initial configuration } X_0 \in Q^V \text{ and time } T > 0.\\
\text{Output: a random } X \in Q^V \text{ identically distributed as the } X_T \text{ in the continuous-time single-site dynamics } (X_t)_{t \in \mathbb{R} \geq 0} \text{ specified by } \{P^T_v\} \text{ conditioning on } X_0.\\
\]

1. \textbf{forall } \(v \in V\) \textbf{ in parallel do} 
2. \hspace{1em} \(T_0^v \leftarrow 0;\)
3. \hspace{1em} generate \(0 < T_1^v < \cdots < T_{M_v}^v < T\) by independently simulating a rate-1 Poisson process;
4. \textbf{end}
5. \textbf{forall updates} \((v, i)\) \textbf{ (where } \(v \in V, 1 \leq i \leq M_v)\) \textbf{ in parallel do}
6. \hspace{1em} compute \(\text{pred}_u(v, i) \triangleq \max\{j \geq 0 \mid T_j^u < T_i^v\};\)
7. \textbf{end}
8. \textbf{forall updates} \((v, i)\) \textbf{ (where } \(v \in V, 1 \leq i \leq M_v)\) \textbf{ in parallel do}
9. \hspace{1em} \(\hat{X}_0^{(0)}[i] \leftarrow X_0(v);\)
10. \hspace{1em} create a new instance of ConsistSampler \(S_{(v,i)}\) and call \(S_{(v,i)}.\text{Initialize}(||Q|\ln|V||);\)
11. \textbf{end}
12. \(\ell \leftarrow 0;\)
13. \textbf{repeat}
14. \hspace{1em} \(\ell \leftarrow \ell + 1;\)
15. \hspace{1em} \textbf{forall } \(v \in V\) \textbf{ in parallel do} \(\hat{X}_0^{(\ell)}[0] \leftarrow X_0(v);\)
16. \hspace{1em} \textbf{forall updates} \((v, i)\) \textbf{ (where } \(v \in V, 1 \leq i \leq M_v)\) \textbf{ in parallel do}
17. \hspace{2em} let \(\tau \in Q_{M_v}^v\) be constructed as: \(\forall u \in N_v^+, \tau_u \leftarrow \hat{X}_u^{(\ell-1)}[\text{pred}_u(v, i)];\)
18. \hspace{2em} \(\hat{X}_v^{(\ell)}[i] \leftarrow S_{(v,i)}.\text{Draw}(P^T_v);\)
19. \textbf{end}
20. \textbf{until } \(\hat{X}^{(\ell)} = \hat{X}^{(\ell-1)};\)
21. \textbf{return } X = \left(\hat{X}_v^{(\ell)}[M_v]\right)_{v \in V};

The output of Algorithm 3 is identical to that of Algorithm 2 on the same input distribution and random bits, ensuring that Algorithm 3 is 2-competitive. Under the assumption of Condition 1 with parameter \(\rho\), and choosing \(I_0 \leq \ln(n)\), Lemma 3.4 shows that Simulation \((X_0, T_0)\) terminates within \(O(\rho \cdot T_0 + \log \frac{n}{\epsilon}) = O(\rho \cdot \log \frac{n}{\epsilon})\) iterations of the repeat loop in Line 13 of Algorithm 4 with probability at least \(1 - \epsilon\). Within each iteration, the algorithm makes parallel calls to Algorithm 3. With probability at least \(1 - \exp(-n)\) there are \(O(nT_0) = O(n \ln(n))\) parallel calls to Algorithm 3 in each iteration, because the total number of updates follows a Poisson distribution with mean \(nT_0 \leq n \ln(n)\). For each individual call to Algorithm 3, the number of rounds required follows a geometric distribution with a success probability of at least \(1 - \exp(-n) = 1 - 1/n\). This is bounded by \(O(1 + \log \frac{n}{\epsilon})\) with probability at least \(1 - \epsilon\), while each round takes \(O(1)\) depth to compute. Apart from the repeat loop, the depth of Simulation \((X_0, T_0)\) is dominated by the sequential computations in Line 3 and Line 6 which are both bounded by \(\max_{v \in V} M_v\), the maximum number of times a Poisson clock rings at a site. This is bounded by \(O(\log \frac{n}{\epsilon})\) with probability at least \(1 - \epsilon\) due to the concentration of the Poisson random variable with mean \(T_0 \leq \ln(n)\). Overall, the depth of Simulation \((X_0, T_0)\) for a \(T_0 \leq \ln(n)\) is bounded by \(O(\rho \cdot \log \frac{n}{\epsilon}(1 + \log \ln \frac{1}{\epsilon})\) with probability at least \(1 - \epsilon - \exp(-\Omega(n))\). Now consider Algorithm 5 which runs for \(\ln(n) < T \leq n^{c_0}\). It divides \(T\) into \(T/\ln(n)\) parts. Choosing the error as \(\epsilon = n^{-c_1}/T\) and applying the union bound, with probability at least \(1 - n^{-c_1}\), the depth of Algorithm 5 is bounded by \(O_{c_1}(\frac{T}{\ln(n)} \cdot \rho \cdot \log(nT) \cdot \log(nT)) = O_{c_0,c_1}(\rho \cdot T)\) because \(T \leq n^{c_0}\). Taking the maximum of the \(0 < T \leq \ln(n)\) and \(T > \ln(n)\) cases, the depth of Algorithm 5 is always upper bounded by
Algorithm 5: Parallel simulation of single-site dynamics with reduced total work

**Input:** initial configuration $X_0 \in Q^V$ and time $T > 0$.

**Output:** a random $X \in Q^V$ identically distributed as the $X_T$ in the continuous-time single-site dynamics $(X_t)_{t \in \mathbb{R}_{\geq 0}}$ specified by $\{P_{v}^T\}$ conditioning on $X_0$.

1. while $T > \ln(n)$ do
2.   $X_0 \leftarrow \text{Simulation} \left( X_0, \ln(n) \right)$;
3.   $T \leftarrow T - \ln(n)$;
4. end
5. return $\text{Simulation} \left( X_0, T \right)$;

$O_{\omega,c_1}(\rho \cdot (T + \log n))$ with probability at least $1 - n^{-c_1}$.

Algorithm 5 uses the same number of processors as the Simulation $(X_0, T_0)$ in Algorithm 4 for a $T_0 \leq \ln(n)$. It suffices to bound the number of processors used for the latter. The number of processors used by Simulation $(X_0, T_0)$ is dominated by the calls to Algorithm 3 in Line 6 and the calls to Algorithm 3 in Line 18. The former uses $\sum_{v \in V} M_v \deg(v) = \sum_{(u,v) \in E} (M_u + M_v)$ processors, where $M_v$, the number of times the Poisson clocks rings at site $v$, follows a Poisson distribution with mean $T_0 \leq \ln(n)$. Hence, the number of processors used in each iteration, the number of parallel calls to Algorithm 3 in Line 18, is given by the number of updates $M = \sum_{v \in V} M_v$, which follows a Poisson distribution with mean $nT_0 \leq n\ln(n)$, and is thus bounded by $O(n T_0) = O(n \log n)$ with probability $1 - \exp(-\Omega(n))$. Each call to Algorithm 3 takes $(q \ln(n))^2$ processors for an $O(1)$ depth computation of the min index in the CRCW-PRAM model. Overall, the total number of processors is bounded by $O_{c_1}((m + n q^2 \log^2 n) \log n)$ with probability at least $1 - n^{-c_1}$.

This proves Item 2.

Finally, to prove Item 3 of Theorem 5.1 consider Algorithm 4 where the Sample($\cdot$, $\cdot$) subroutine is realized by inverse transform sampling on the Boolean domain, as described in Definition 3.6. Specifically, in Algorithm 4 [Line 10] is replaced by drawing $R_{(v,i)} \in [0, 1]$ uniformly and independently at random; and [Line 18] is replaced by:

$$\hat{X}^{(i)}[i] \leftarrow \text{if } R_{(v,i)} \geq P_{v}^T(0),$$

By Corollary 3.7 Simulation $(X_0, T)$ terminates within $O \left( \frac{1}{1 - \epsilon} \log \left( \frac{nT}{\epsilon} \right) \right)$ iterations of the repeat loop in Line 2 of Algorithm 4 with probability at least $1 - \epsilon$. The total number of updates follows a Poisson distribution with mean $nT$, so with probability at least $1 - \exp(-n)$, there are $O(nT)$ updates in total. Each iteration of the repeat loop in Line 2 of Algorithm 1 uses $O(1)$ depth, taking $O_{c_1}(mT)$ processors. The $O_{c_1}(mT)$ processors can be reused between iterations. Before entering the repeat loop, the calculations of the predecessors of all updates in Line 6 of Algorithm 4 require $O_{c_1}(mT)$ processors. Overall, with probability at least $1 - n^{-c_1}$, the depth of Simulation $(X_0, T)$ is $O_{c_1} \left( \frac{1}{1 - \epsilon} (\log T + \log n) \right)$ and it uses $O_{c_1}(mT)$ processors. This proves Item 3.

**Remark 5.2** (bounded precision). The random choices in our algorithms—specifically, the times generated by the Poisson clocks and the real number drawn uniformly from $[0, 1]$ in the sampling subroutines Algorithm 2 and Algorithm 3—are assumed to be real numbers of unbounded precision for expositional simplicity. However, this assumption can be relaxed in practical implementations. First, the random update times $T_{v}^o$‘s generated by Poisson clocks do not need to be real numbers. They are used solely to determine the relation $\rightarrow$ between updates, as defined in (5). The relation $\rightarrow$ is uniquely determined as long as all times generated by the Poisson clocks are distinct. This uniqueness is assured with high probability by generating only the first $O(\log n)$ bits for the times using standard methods for simulating Poisson point processes [SKM95].
Furthermore, if all probabilities in the local update distributions \( \{P^*_v\} \) of the single-site dynamics are expressed with \( k \)-bit precision (i.e. they are rational numbers with a denominator of \( 2^k \)), we can construct the universal coupling (correlated sampling) only for those distributions \( p \in \Delta(Q) \) specifically for those distributions with \( k \)-bit precision. In Algorithm 2, Algorithm 3 and Definition 3.6 we can replace the real numbers drawn uniformly from \([0, 1]\) with uniform \( k \)-bit rational numbers from \([0, 1]\). The analyses in Section 4 remain valid with this approach, as (24) holds for any function with \( k \)-bit precision.

### 5.2 Distributed Implementation

The following theorem is a formal restatement of Theorem 1.5.

**Theorem 5.3** (Theorem 1.5, formally stated). Assume Condition 1 with parameter \( \rho \). There is a CONGEST algorithm on the network \( G = (V, E) \) such that, given any \( X_0 \in \mathcal{Q}^V \) and \( 0 < T \leq n^c_1 \), where \( n = |V| \) and \( q = |Q| \), the followings hold:

1. The algorithm terminates within \( O_{c_1}((\rho \cdot (T + \log n)) \) rounds of communication, with each message containing \( O_{c_1}(\log n \log q) \) bits.

2. The algorithm returns a random \( X \in \mathcal{Q}^V \) whose distribution is within \( n^{-c_1} \) total variation distance from the distribution of \( X_T \) in the continuous-time single-site dynamics \( (X_t)_{t \in \mathbb{R} \geq 0} \) specified by \( \{P^*_v\} \) conditioned on \( X_0 \).

**Proof.** A CONGEST algorithm on the network \( G = (V, E) \) is described in Algorithm 6. The algorithm consists of two phases. In **Phase I**, each site \( v \in V \) locally generates all its update times \( 0 < T_1^v < \cdots < T_M^v < T \) up to time \( T \) and exchange them with all its neighbors. After receiving all update times from its neighbors, \( v \) can locally determine the “predecessors” of all its updates \((v, i)\) in the relation \( \rightarrow \) based on the update times within its neighborhood \( N^+_v \). This phase can be completed in \( O(T + \log \frac{n}{q}) \) rounds with probability at least \( 1 - \epsilon \), as the number of updates \( M_v \) for each node \( v \) follows a Poisson distributions with a mean of \( T \). Each message contains an update time \( T_j^v \), which does not need to be a real number with infinite precision. As long as the predecessors \( \text{pred}_i(v, i) \) in Line 4 are correctly computed, the algorithm will run correctly. And this occurs with probability at least \( 1 - \epsilon \) using a message size of \( O(\log \frac{n}{q}) \) bits. In **Phase II**, Algorithm 6 simulates the process of Algorithm 1, with each round in Algorithm 6 corresponding to an iteration of the repeat loop in Algorithm 1. The only communication required is to retrieve the current neighborhood configuration, as specified in Line 11. For each pair of neighbors, this involves at most \( O(T + \log \frac{n}{q}) \) spins and can be represented using \( O((T + \log \frac{n}{q}) \log q) \) bits, with probability at least \( 1 - \epsilon \), due to the concentration of Poisson random variable. According to Lemma 3.4 with probability at least \( 1 - \epsilon \), **Phase II** requires at most \( O(\rho \cdot T + \log \frac{n}{q}) \) rounds to reach a fixpoint. At this point, the algorithm achieves the \( O_{c_1}(\rho \cdot T + \log n) \) round complexity with probability at least \( 1 - n^{-c_1} \), as stated in the theorem, although each message consists of \( O_{c_1}((T + \log n) \log q) \) bits.

To further reduce the message size to \( O(\log n \log q) \) bits, we apply the same trick as in Algorithm 5. For \( \ln(n) < T \leq n^{c_1} \), we divide \( T \) into \( T / \ln(n) \) many subintervals of length \( T_0 = \ln(n) \) and run Algorithm 6 consecutively for these time subintervals, such that Algorithm 6 is executed within each time subinterval for a properly fixed number of rounds without global coordination. This approach works as long as no errors occur. By setting the error probability to \( \epsilon = n^{-c_1} / T \) and applying the union bound, with probability at least \( 1 - n^{-c_1} \), no error occurs, and the output is correct. In the event of an error, the output is treated as arbitrary, introducing a total variation distance error of \( n^{-c_1} \) to the final result. With probability at least \( 1 - n^{-c_1} \), the algorithm that divides \( T \) into subintervals and runs Algorithm 6 as subroutines terminates in \( O_{c_1}(\frac{T}{\ln(n)} \cdot (\rho \cdot \ln n + \log(nT))) = O_{c_1, c_1}(\rho \cdot T) \) rounds.
Algorithm 6: A CONGEST algorithm for simulating single-site dynamics

Input at $v \in V$ : initial spin $X_0(v)$ of site $v \in V$ and $T > 0$.
Output at $v \in V$ : the spin $X(v)$ of site $v$.

1 Phase I:
   2 generate $0 < T_1 < \cdots < T_{M_0} < T$ by independently simulating a rate-1 Poisson process;
   3 forall neighbors $u \in N_v$ do send $(T^u_i)_{1 \leq i \leq M_0}$ to $u$ and receive $(T^v_i)_{1 \leq i \leq M_0}$ from $u$;
      /* in $O(T + \log n)$ rounds of communications w.h.p. */
   4 forall $1 \leq i \leq M_0$ and $u \in N^+_v$ do compute $\text{pred}_u(v, i) \leftarrow \max \{ j \mid T^u_j < T^v_i \} \cup \{ 0 \}$;
      /* only needs the first $O(\log n)$ bits of each $T^u_i$ w.h.p. */
   5 end-of-phase

2 Phase II:
   6 generate independent random $R^{(v, i)}$ for all $1 \leq i \leq M_0$,
      where $R^{(v, i)} = \left( (X^{(v, i)}_1, Y^{(v, i)}_1), (X^{(v, i)}_2, Y^{(v, i)}_2), \ldots \right)$;
   7 /* for some properly fixed $L = O(\rho \cdot T + \log n)$ */
   8 $X^{(0)}_v[i] \leftarrow X_0(v)$ for all $0 \leq i \leq M_0$;
   9 for rounds $\ell = 1$ to $L$ do
      10 $X^{(\ell)}_v[0] \leftarrow X_0(v)$;
      11 retrieve $X^{(\ell-1)}_v[i]$ for all $1 \leq j \leq M_0$ from all neighbors $u \in N_v$;
      12 /* $O((T + \log n) \log q)$-bits message from each neighbor w.h.p. */
      13 forall $1 \leq i \leq M_0$ do
         14 let $\tau \in Q^{N_v}$ be constructed as: $\forall u \in N^+_v$, $\tau_u \leftarrow X^{(\ell-1)}_v[\text{pred}_u(v, i)]$;
         15 $X^{(\ell)}_v[i] \leftarrow \text{Sample} \left( P^v_\tau, R^{(v, i)} \right)$;
      16 /* use the Sample in Algorithm 2 */
   17 end
   18 return $X(v) = X^{(L)}_v[M_0]$;
   19 end-of-phase

since $T \leq n^{c_0}$, with each message consisting of $O(c_1 \log n \log q)$ bits. In the case where $T < \ln(n)$, with probability $\geq 1 - n^{-c_1}$, Algorithm 6 terminates in $O(c_1 (\rho \cdot T + \log n))$ rounds, with each message consisting of $O(c_1 ((T + \log n) \log q))$ bits.

Overall, with probability at least $1 - n^{-c_1}$, the final algorithm (which applies Algorithm 6 with the trick as in Algorithm 5) terminates within $O(c_{0, c_1} (\rho \cdot (T + \log n)))$ rounds, with each message consisting of $O(c_1 \log n \log q)$ bits. The algorithm faithfully simulates Algorithm 1 except for an error probability of at most $n^{-c_1}$.

6 Applications

6.1 Application for Hardcore and Ising Model

Proof of Corollary 1.8 For the hardcore models with $\lambda \leq (1 - \delta) \lambda_c(\Delta) = (1 - \delta) \frac{(\Delta - 1)^{\Delta - 1}}{(\Delta - 2)^{\Delta - 1}}$, and Ising models with $\beta \in \left( \frac{\Delta - 2 + \delta}{\Delta - 3}, \frac{\Delta - \delta}{\Delta - 2 + \delta} \right)$, the modified log-Sobolev inequalities (MSLI) established in [CFYZ22]
show that the continuous-time Glauber dynamics has mixing time $t_{\text{mix}}^C(e) = O_\delta \left( \log \frac{n}{\beta} \right)$. For the hardcore models with $\lambda = O \left( \frac{1}{\Delta} \right)$ and Ising models with $\beta \in 1 \pm O \left( \frac{1}{\Delta} \right)$, the Dobrushin’s influence matrix for the Glauber dynamics satisfies $\rho(u,v) = 0$ for non-adjacent $u, v \in V$, and satisfies the following for adjacent $\{u, v\} \in E$, where the degree of $v$ is denoted as $d_v \triangleq |N_v| \leq \Delta$:

(hardcore model) \[ \rho(u,v) = \frac{\lambda}{1 + \lambda} = O \left( \frac{1}{\Delta} \right), \]

(Ising model) \[ \rho(u,v) = \min_{0 \leq k \leq d_v - 1} \left\{ \frac{1}{1 + \lambda \beta^{d_v - 2k}} - \frac{1}{1 + \lambda \beta^{d_v - 2(k+1)}} \right\} = O \left( \frac{1}{\Delta} \right). \]

Thus, we have $\|\rho\|_\infty = O(1)$, meaning Condition 1 holds with parameter $\rho = O(1)$.

Applying Theorem 5.1 with $T = t_{\text{mix}}^C \left( \frac{1}{\text{poly}(n)} \right) = O_\delta \left( \log n \right)$, Algorithm 5 returns an approximate sample that is $1/\text{poly}(n)$-close in total variation distance to the Gibbs distribution. This algorithm uses $O_\delta \left( \log n \right)$ depth on $\tilde{O}(m + n) = \tilde{O}(m)$ processors with high probability, assuming the availability of oracles to evaluate the marginal probabilities:

(hardcore model) \[ \forall \tau \in \{0, 1\}^N, \quad \mu^T_v(1) = \begin{cases} 0 & \|\tau\|_1 \geq 1, \\ \frac{\lambda}{1 + \lambda} & \|\tau\|_1 = 0 \end{cases}, \]

(Ising model) \[ \forall \tau \in \{0, 1\}^N, \quad \mu^T_v(1) = \frac{\lambda \beta^{-d_v + 2\|\tau\|_1}}{1 + \lambda \beta^{\|\tau\|_1/(d_v - \|\tau\|_1)}}. \]

These oracles can be implemented with $O(\log d_v) = O(\log \Delta)$ depth using $O(d_v)$ processors. The overhead in processor usage contributes only to the $\tilde{O}(n)$ term in the $\tilde{O}(m + n)$ total processor bound. This results in an overall bound of $O_\delta \left( \log n \cdot \log \Delta \right)$ depth and $\tilde{O}(m + \sum_{v \in V} d_v) = \tilde{O}(m)$ processors.

### 6.2 Application for SAT Solutions

**Proof of Corollary 1.9** Let $\mu$ denote the uniform distribution over all satisfying assignments of $\Phi$.

The sampling algorithm proposed in [FGYZ21] is composed of three key steps:

1. Constructing the set $M \subseteq V$ of “marked” variables: A set $M \subseteq V$ that satisfies [FGYZ21] Condition 3.1 is constructed using the Moser-Tardos algorithm. This can be parallelized via the parallel Moser-Tardos algorithm [MT10]. The local lemma condition (2) is sufficiently strong to allow the construction of $M \subseteq V$ in parallel while ensuring [FGYZ21] Condition 3.1 is met.

2. Simulate the Glauber dynamics for $\mu_M$: The Glauber dynamics is then simulated for the distribution $\mu_M$, which is the projection of $\mu$ onto $M$. This involves a single-site dynamics on the complete graph induced by $M$, with local update distributions $\{P_v^T\}$ defined as $P_v^T = \mu_v^{T_M[n]}$, where $T$ is the current assignment on $M$. According to [FGYZ21] Lemma 4.2, all configurations on $M$ have positive measure under $\mu_M$, ensuring that $\{P_v^T\}$ are well-defined. By [FGYZ21] Lemma 4.1, this chain mixes within $t_{\text{mix}}^D(e) = O(n \log \frac{n}{\beta})$ steps, which translates to a continuous-time mixing time $t_{\text{mix}}^C(e) = O(\log \frac{n}{\beta})$ via (3). This mixing time bound arises from a path coupling argument [FGYZ21] Equation (11)], which also ensures Condition 1 holds with parameter $\rho = O(1)$ in the $\ell_\infty$ norm. Applying Theorem 5.1 with $T = t_{\text{mix}}^C \left( \frac{1}{\text{poly}(n)} \right) = O(\log n)$, Algorithm 5 simulates this chain within $O(\log n)$ depth using $\tilde{O}(|M|^2) = \tilde{O}(n^2)$ processors, given access to oracles for evaluating $\mu_v^{T_M[n]}$.

To draw from the marginal distribution $\mu_v^{T_M[n]}$, [FGYZ21] Algorithm 3 is used. During each update of the chain, with high probability, the clauses already satisfied by the current assignment
on $\mathcal{M} \setminus \{v\}$ disconnect the CNF into components of small sizes. Rejection sampling can be applied to the component containing $v$, with each trial succeeding with a probability of at least $n^{-2^{-20}}$.

3. Extending the partial assignment of “marked” variables to all variables: After the Glauber dynamics on $\mu_M$ has sufficiently mixed, the random assignment on $\mathcal{M}$ is extended to all variables. This extension is also performed using [FGYZ21, Algorithm 3], which samples from $\mu^\tau_{\mathcal{T}_1 \setminus \mathcal{M}}$.

It remains to ensure the parallelizability of: (1) The marginal sampler [FGYZ21, Algorithm 3], and (2) the oracles to evaluate the marginal distribution $\mu^\tau_{\mathcal{T}_1 \setminus \mathcal{M}}$. For (1), the marginal sampler [FGYZ21, Algorithm 3] draws from $\mu^\tau_{\mathcal{T}_1 \setminus \mathcal{M}}$, where $S, T \subseteq V$ are disjoint. It first constructs the connected components that intersect $5$ in the formula of clauses not yet satisfied by $\tau_T$, and then draws from $\mu^\tau_{\mathcal{T}_1 \setminus \mathcal{M}}$ via rejection sampling on these components. The first step can be parallelized using a connected component algorithm such as [SV80] and the latter is trivially parallelizable, and the subsequent rejection sampling is trivially parallelizable. For (2), due to the Chernoff-Hoeffding bound, the marginal probabilities $\mu^\tau_{\mathcal{T}_1 \setminus \mathcal{M}}$ can be estimated with an additive error of $1/poly(n)$ with $1 - 1/poly(n)$ confidence by independently (and thus in parallel) repeating [FGYZ21, Algorithm 3] $poly(n)$ times. This introduces only $1/poly(n)$ total variation error into the final output.

7 Conclusion and Open Problems

In this work, we present a generic parallel algorithm for faithful simulation of single-site dynamics. Assuming a substantially weakened asymptotic variant of $\ell_p$-Dobrushin’s condition, our algorithm achieves linear speedup in $n$ when parallelizing single-site dynamics across $n$ sites. If the strict $\ell_p$-Dobrushin’s condition is assumed for Boolean random variables, the parallelization achieves an exponential speedup.

The asymptotic Dobrushin’s condition required for linear speedup is relatively easy to satisfy. It essentially requires that the discrepancy in the system does not propagate at a super-exponential rate, which is quite modest compared to the usual requirement for fast mixing, where discrepancy should decay.

Under this mild condition, our parallel simulation algorithm can convert single-site dynamics with near-linear mixing time into RNC algorithms for sampling, provided the marginal distributions for single-site updates are RNC-computable. Additionally, through non-adaptive simulated annealing, we can also obtain RNC algorithms for approximate counting.

Open problems. The study of efficient parallel algorithms for sampling and counting is highly motivated by practical applications. In this paper, we address these challenges through the faithful parallel simulation of single-site dynamics and non-adaptive simulated annealing. Our approach is generic and well-suited for leveraging existing results on single-site Markov chains with well-studied mixing properties. Despite the advancements provided by our approach, the problem of giving RNC counterparts for well-known sampling and counting algorithms remains widely open. We conclude by presenting a few concrete open problems:

- We apply our result to the chain in [FGYZ21], which provides a parallel sampler for satisfying solutions of the CNFs in the local lemma regime. However, the total work of this sampler is a large polynomial. A key open question is to design a parallel sampler with polylogarithmic depth while maintaining total work comparable to the sequential sampler in [FGYZ21], which is close to linear in $n$, the number of variables.
• In this paper, we utilize correlated sampling to ensure both the efficiency and faithfulness of parallel simulations for single-site dynamics. To our knowledge, this marks the first application of correlated sampling in parallelizing stochastic processes. An intriguing direction for future research is how this approach can be generalized to parallelize randomized algorithms.

• A significant challenge is to parallelize Markov chains with super-linear polynomial mixing times into RNC algorithms. Notable examples include the Jerrum-Sinclair chains for matchings and the ferromagnetic Ising model with no external field \cite{Jer03}. Finding RNC counterparts for these algorithms remains a major open problem.

• An even greater challenge is to give an NC (deterministic) algorithm for approximating counting, especially for graphical models that have unbounded maximum degree.

References

\[\text{[AAG}^+\text{12]}\] Amr Ahmed, Moahmed Aly, Joseph Gonzalez, Shravan Narayanamurthy, and Alexander J Smola. Scalable inference in latent variable models. In Proceedings of the 5th ACM international conference on Web search and data mining (WSDM), pages 123–132, 2012.

\[\text{[ABTV23]}\] Nima Anari, Callum Burgess, Kevin Tian, and Thuy-Duong Vuong. Quadratic speedups in parallel sampling from determinantal distributions. In Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pages 367–377, 2023.

\[\text{[AGC}^+\text{22]}\] Navid Anjum Aadit, Andrea Grimaldi, Mario Carpentieri, Luke Theogarajan, John M Martinis, Giovanni Finocchio, and Kerem Y Camsari. Massively parallel probabilistic computing with sparse ising machines. Nature Electronics, 5(7):460–468, 2022.

\[\text{[AGR24]}\] Nima Anari, Ruiquan Gao, and Aviad Rubinstein. Parallel sampling via counting. In Proceedings of the 56th Annual ACM Symposium on Theory of Computing, pages 537–548, 2024.

\[\text{[AHL}^+\text{23]}\] Nima Anari, Yizhi Huang, Tianyu Liu, Thuy-Duong Vuong, Brian Xu, and Katherine Yu. Parallel discrete sampling via continuous walks. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC), pages 103–116, 2023.

\[\text{[AHSS21]}\] Nima Anari, Nathan Hu, Amin Saberi, and Aaron Schild. Sampling arborescences in parallel. In Proceedings of the 12th Innovations in Theoretical Computer Science Conference (ITCS), volume 185, page 18, 2021.

\[\text{[AJK}^+\text{22]}\] Nima Anari, Vishesh Jain, Frederic Koehler, Huy Tuan Pham, and Thuy-Duong Vuong. Entropic independence: optimal mixing of down-up random walks. In STOC, pages 1418–1430. ACM, 2022.

\[\text{[ALO20]}\] Nima Anari, Kuikui Liu, and Shayan Oveis Gharan. Spectral independence in high-dimensional expanders and applications to the hardcore model. In FOCS, pages 1319–1330. IEEE, 2020.

\[\text{[AS19]}\] Omer Angel and Yinon Spinka. Pairwise optimal coupling of multiple random variables. arXiv, abs/1903.00632, 2019.
[BCC+22] Antonio Blanca, Pietro Caputo, Zongchen Chen, Daniel Parisi, Daniel Štefankovič, and Eric Vigoda. On mixing of markov chains: Coupling, spectral independence, and entropy factorization. In Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 3670–3692, 2022.

[BGH+20] Mohammad Bavarian, Badih Ghazi, Elad Haramaty, Pritish Kamath, Ronald L Rivest, and Madhu Sudan. Optimality of correlated sampling strategies. Theory of Computing, 16(1), 2020.

[BGH+23] Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Satchit Sivakumar, and Jessica Sorrell. Stability is stable: Connections between replicability, privacy, and adaptive generalization. In Proceedings of the 55th Annual ACM SIGACT Symposium on Theory of Computing (STOC), pages 520–527, 2023.

[BHH+08] Boaz Barak, Moritz Hardt, Ishay Haviv, Anup Rao, Oded Regev, and David Steurer. Rounding parallel repetitions of unique games. In Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS ’08, page 374–383, 2008.

[Bro97] Andrei Z Broder. On the resemblance and containment of documents. In Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171), pages 21–29. IEEE, 1997.

[BRY20] Amartya Shankha Biswas, Ronitt Rubinfeld, and Anak Yodpinyanee. Local access to huge random objects through partial sampling. In Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS), volume 151, page 27, 2020.

[CE22] Yuansi Chen and Ronen Eldan. Localization schemes: A framework for proving mixing bounds for markov chains (extended abstract). In FOCS, pages 110–122. IEEE, 2022.

[CFV24] Charlie Carlson, Daniel Frishberg, and Eric Vigoda. Improved Distributed Algorithms for Random Colorings. In 27th International Conference on Principles of Distributed Systems (OPODIS 2023), volume 286, pages 13:1–13:18, 2024.

[CFYZ21] Xiaoyu Chen, Weiming Feng, Yitong Yin, and Xinyuan Zhang. Rapid mixing of glauber dynamics via spectral independence for all degrees. In FOCS, pages 137–148. IEEE, 2021.

[CFYZ22] Xiaoyu Chen, Weiming Feng, Yitong Yin, and Xinyuan Zhang. Optimal mixing for two-state anti-ferromagnetic spin systems. In FOCS, pages 588–599. IEEE, 2022.

[Cha02] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing, STOC ’02, page 380–388, 2002.

[CLV20] Zongchen Chen, Kuikui Liu, and Eric Vigoda. Rapid mixing of glauber dynamics up to uniqueness via contraction. In FOCS, pages 1307–1318. IEEE, 2020.

[CLV21] Zongchen Chen, Kuikui Liu, and Eric Vigoda. Optimal mixing of glauber dynamics: entropy factorization via high-dimensional expansion. In STOC, pages 1537–1550. ACM, 2021.

[DDJ18] Constantinos Daskalakis, Nishanth Dikkala, and Siddhartha Jayanti. Hogwild!-Gibbs can be panaccurate. In Proceedings of the 31st Advances in Neural Information Processing Systems (NIPS), pages 32–41, 2018.

31
[DFK91] Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM (JACM)*, 38(1):1–17, 1991.

[DGJ09] Martin Dyer, Leslie Ann Goldberg, and Mark Jerrum. Matrix norms and rapid mixing for spin systems. *The Annals of Applied Probability*, 19(1):71–107, 2009.

[Dob70] Roland L. Dobrushin. Prescribing a system of random variables by conditional distributions. *Theory of Probability & Its Applications*, 15(3):458–486, 1970.

[DS85a] Roland L Dobrushin and Senya B Shlosman. Completely analytical gibbs fields. In *Statistical physics and dynamical systems*, pages 371–403. Springer, 1985.

[DS85b] Roland Lvovich Dobrushin and Senya B Shlosman. Constructive criterion for the uniqueness of gibbs field. In *Statistical physics and dynamical systems*, pages 347–370. Springer, 1985.

[DSOR16] Christopher De Sa, Kunle Olukotun, and Christopher Ré. Ensuring rapid mixing and low bias for asynchronous Gibbs sampling. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1567–1576, 2016.

[EKZ21] Ronen Eldan, Frederic Koehler, and Ofer Zeitouni. A spectral condition for spectral gap: fast mixing in high-temperature ising models. *Probability Theory and Related Fields*, pages 1–17, 2021.

[FG18] Manuela Fischer and Mohsen Ghaffari. A simple parallel and distributed sampling technique: Local glauber dynamics. In *32nd International Symposium on Distributed Computing (DISC)*, volume 121, pages 26–1, 2018.

[FGYZ21] Weiming Feng, Heng Guo, Yitong Yin, and Chihao Zhang. Fast sampling and counting k-sat solutions in the local lemma regime. *Journal of the ACM (JACM)*, 68(6):1–42, 2021.

[FHY21a] Weiming Feng, Thomas P Hayes, and Yitong Yin. Distributed metropolis sampler with optimal parallelism. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2121–2140, 2021.

[FHY21b] Weiming Feng, Kun He, and Yitong Yin. Sampling constraint satisfaction solutions in the local lemma regime. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1565–1578. ACM, 2021.

[FSY17] Weiming Feng, Yuxin Sun, and Yitong Yin. What can be sampled locally? In *Proceedings of the 36th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 121–130, 2017.

[FVY21] Weiming Feng, Nisheeth K Vishnoi, and Yitong Yin. Dynamic sampling from graphical models. *SIAM Journal on Computing*, 50(2):350–381, 2021.

[FY18] Weiming Feng and Yitong Yin. On local distributed sampling and counting. In *Proceedings of the 37th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 189–198, 2018.

[GJL19] Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the lovász local lemma. *Journal of the ACM (JACM)*, 66(3):1–31, 2019.
[GKM21] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. User-level differentially private learning via correlated sampling. *Advances in Neural Information Processing Systems*, 34:20172–20184, 2021.

[Gla63] Roy J. Glauber. Time-dependent statistics of the Ising model. *Journal of mathematical physics*, 4(2):294–307, 1963.

[GLGG11] Joseph E Gonzalez, Yucheng Low, Arthur Gretton, and Carlos Guestrin. Parallel Gibbs sampling: From colored fields to thin junction trees. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 324–332, 2011.

[GM07] A Gerschcnfeld and A Monianari. Reconstruction for models on random graphs. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 194–204, 2007.

[GŠV16] Andreas Galanis, Daniel Štefankovič, and Eric Vigoda. Inapproximability of the partition function for the antiferromagnetic Ising and hard-core models. *Combin. Probab. Comput.*, 25(4):500–559, 2016.

[Hay06] Thomas P. Hayes. A simple condition implying rapid mixing of single-site dynamics on spin systems. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 39–46, 2006.

[Hol07] Thomas Holenstein. Parallel repetition: simplifications and the no-signaling case. In *Proceedings of the 39th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 411–419, 2007.

[HS07] Thomas P. Hayes and Alistair Sinclair. A general lower bound for mixing of single-site dynamics on graphs. *The Annals of Applied Probability*, pages 931–952, 2007.

[HSV21] Kun He, Xiaoming Sun, and Kewen Wu. Perfect sampling for (atomic) Lovász local lemma. *arXiv*, abs/2107.03932, 2021.

[Jer03] Mark Jerrum. *Counting, sampling and integrating: algorithms and complexity*. Springer Science & Business Media, 2003.

[JLY19] Michael I Jordan, Jason D Lee, and Yun Yang. Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 114(526):668–681, 2019.

[JPV21] Vishesh Jain, Huy Tuan Pham, and Thuy Duong Vuong. On the sampling Lovász local lemma for atomic constraint satisfaction problems. *arXiv*, abs/2102.08342, 2021.

[JS93] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM Journal on Computing*, 22(5):1087–1116, 1993.

[JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM (JACM)*, 51(4):671–697, 2004.

[KKMV23] Alkis Kalavasis, Amin Karbasi, Shay Moran, and Grigoris Velegkas. Statistical indistinguishability of learning algorithms. In *International Conference on Machine Learning*, pages 15586–15622. PMLR, 2023.
[KR91] Richard M Karp and Vijaya Ramachandran. Parallel algorithms for shared-memory machines. In *Handbook of theoretical computer science (vol. A) algorithms and complexity*, pages 869–941. 1991.

[KT02] Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002.

[KVYZ23] Amin Karbasi, Grigoris Velegkas, Lin Yang, and Felix Zhou. Replicability in reinforcement learning. *Advances in Neural Information Processing Systems*, 36:74702–74735, 2023.

[LPW17] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, Providence, RI, 2017.

[M+87] Nicholas Metropolis et al. The beginning of the monte carlo method. *Los Alamos Science*, 15(584):125–130, 1987.

[Mar19] Katalin Marton. Logarithmic sobolev inequalities in discrete product spaces. *Combinatorics, Probability and Computing*, 28(6):919–935, 2019.

[MM09] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.

[Moi19] Ankur Moitra. Approximate counting, the lovász local lemma, and inference in graphical models. *Journal of the ACM (JACM)*, 66(2):1–25, 2019.

[MS15] Alireza S Mahani and Mansour TA Sharabiani. Simd parallel mcmc sampling with applications for big-data bayesian analytics. *Computational Statistics & Data Analysis*, 88:75–99, 2015.

[MT10] Robin A Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. *Journal of the ACM (JACM)*, 57(2):11, 2010.

[MVV87] Ketan Mulmuley, Umesh V Vazirani, and Vijay V Vazirani. Matching is as easy as matrix inversion. In *Proceedings of the 19th annual ACM Symposium on Theory of Computing (STOC)*, pages 345–354, 1987.

[NRRW11] Feng Niu, Benjamin Recht, Christopher Re, and Stephen J Wright. Hogwild! a lock-free approach to parallelizing stochastic gradient descent. In *Proceedings of the 24th Advances in Neural Information Processing Systems (NIPS)*, pages 693–701, 2011.

[NWX14] Willie Neiswanger, Chong Wang, and Eric P. Xing. Asymptotically exact, embarrassingly parallel mcmc. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI)*, page 623–632, 2014.

[Pea82] Judea Pearl. Reverend bayes on inference engines: a distributed hierarchical approach. In *Proceedings of the Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 133–136, 1982.

[Rao08] Anup Rao. Parallel repetition in projection games and a concentration bound. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC ’08, page 1–10, 2008.

[Riv16] Ronald L Rivest. Symmetric encryption via keyrings and ecc. In *Northernmost Crypto Workshop, Longyearbyen, Norway, Midnight Lect*, 2016.
[SBB+22] Steven L Scott, Alexander W Blocker, Fernando V Bonassi, Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayes and big data: The consensus monte carlo algorithm. In Big Data and Information Theory, pages 8–18. Routledge, 2022.

[SKM95] D. Stoyan, W.S. Kendall, and J. Mecke. Stochastic Geometry and Its Applications. Inorganic Chemistry. Wiley, 1995.

[SN10] Alexander Smola and Shravan Narayanamurthy. An architecture for parallel topic models. Proc. VLDB Endow., 3(1–2):703–710, sep 2010.

[SS14] Allan Sly and Nike Sun. Counting in two-spin models on d-regular graphs. The Annals of Probability, 42(6):2383–2416, 2014.

[STW19] Madhu Sudan, Himanshu Tyagi, and Shun Watanabe. Communication for generating correlation: A unifying survey. IEEE Transactions on Information Theory, 66(1):5–37, 2019.

[SV80] Yossi Shiloach and Uzi Vishkin. An o (log n) parallel connectivity algorithm. Technical report, Computer Science Department, Technion, 1980.

[ŠVV09] Daniel Štefankovič, Santosh Vempala, and Eric Vigoda. Adaptive simulated annealing: A near-optimal connection between sampling and counting. Journal of the ACM (JACM), 56(3):1–36, 2009.

[Ten95] Shang-Hua Teng. Independent sets versus perfect matchings. Theoretical Computer Science, 145(1-2):381–390, 1995.

[TSD20] Alexander Terenin, Daniel Simpson, and David Draper. Asynchronous gibbs sampling. In Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), pages 144–154, 2020.