FAST SAMPLING AND COUNTING k-SAT SOLUTIONS IN THE LOCAL LEMMA REGIME

WEIMING FENG, HENG GUO, YITONG YIN, AND CHIHAO ZHANG

Abstract. We give new algorithms based on Markov chains to sample and approximately count satisfying assignments to k-uniform CNF formulas where each variable appears at most d times. For any k and d satisfying $kd < n^{o(1)}$ and $k \geq 20 \log k + 20 \log d + 60$, the new sampling algorithm runs in close to linear time, and the counting algorithm runs in close to quadratic time.

Our approach is inspired by Moitra (JACM, 2019) which remarkably utilizes the Lovász local lemma in approximate counting. Our main technical contribution is to use the local lemma to bypass the connectivity barrier in traditional Markov chain approaches, which makes the well developed MCMC method applicable on disconnected state spaces such as SAT solutions. The benefit of our approach is to avoid the enumeration of local structures and obtain fixed polynomial running times, even if $k = \omega(1)$ or $d = \omega(1)$.

1. Introduction

Sampling from an exponential-sized solution space and estimating the number of feasible solutions are two very related fundamental computation problems. The Markov chain Monte Carlo (MCMC) method is the most successful technique due to its generic nature and the fast running time, with many famous applications such as [DFK91, JSV04]. A basic requirement for the method to apply is that the state space has to be connected via moves of the Markov chain to let the chain converge to the desired distribution. This requirement prevents us from applying the method to the problems where the solution space is not connected via local moves. Unfortunately, this barrier holds for perhaps the most important solution space in Computer Science: the satisfying assignments of *conjunctive normal form* (CNF) formulas [Wig19].

Recently, a number of new methods based on the variable framework of the Lovász local lemma were proposed to tackle the problem [Moi19, GJL19]. Most notably, the breakthrough of [Moi19] introduced a novel approach for estimating the number of solutions of k-SAT in a local lemma regime. By far, it is still the only tractable result for sampling and approximately counting k-SAT solutions in the local lemma regime without additional structural assumptions on the formulas. However, since this new algorithm relies on local enumeration, its time cost is in the form of $n^{O(d^2k^2)}$, where d is the variable degree in the local lemma. Although a polynomial time for constant d and k, this time cost is not fixed-parameter tractable with parameters d and k. Indeed, for $d = \omega(1)$ or $k = \omega(1)$, the running time becomes super-polynomial.

In this paper, we develop a new approach to overcome the connectivity barrier for Markov chain methods. The main idea is to sample from the marginal probability of an algorithmically chosen subset of variables, so that the standard Glauber dynamics is now ergodic. However, this distribution is not a Gibbs distribution nor satisfies any kind of conditional independence properties. New challenges arise as both analyzing and implementing the Glauber dynamics require new ideas. We give a high-level overview of the techniques in Section 1.1.

(Weiming Feng, Yitong Yin) State Key Laboratory for Novel Software Technology, Nanjing University, 163 Xianlin Avenue, Nanjing, Jiangsu Province, China. E-mail: fengwm@smail.nju.edu.cn, yinyt@nju.edu.cn

(Heng Guo) School of Informatics, University of Edinburgh, Informatics Forum, Edinburgh, EH8 9AB, United Kingdom. E-mail: hguo@inf.ed.ac.uk

(Chihao Zhang) John Hopcroft Center for Computer Science, Shanghai Jiao Tong University, 800 Dongchuan Road, Minhang District, Shanghai, China. E-mail: chihao@sjtu.edu.cn

This research was supported by the National Key R&D Program of China 2018YFB1003202 and the NSFC Nos. 61722207, 61672275 and 61902241.

Part of the research was done when Weiming Feng was visiting the University of Edinburgh.

To illustrate the new technique, we choose a canonical #**P**-complete problem, namely counting the number of satisfying assignments of CNF formulas (#Sat) as our main application. We call a CNF formula Φ a (k,d)-formula if all of its clauses have size k and each variable appears in at most d clauses.

Theorem 1.1 (simplified). *The following holds for any sufficiently small $\zeta > 0$.*

There is an algorithm such that given any $0 < \varepsilon < 1$ and (k,d)-formula Φ with n variables where $k \geq 20\log k + 20\log d + 3\log\left(\frac{1}{\zeta}\right)$, it terminates in time $\widetilde{O}\left(d^2k^3n\left(\frac{n}{\varepsilon}\right)^{\zeta}\right)$ and outputs a random assignment X that is ε close in total variation distance to the uniform distribution over satisfying assignments of Φ.

Moreover, there is an algorithm that given any $0 < \varepsilon < 1$ and (k,d)-formula Φ, under the same assumption, terminates in time $\widetilde{O}\left(d^3k^3\left(\frac{n}{\varepsilon}\right)^{2+\zeta}\right)$ and outputs an e^{ε}-approximation to the number of satisfying assignments of Φ. In the above, $\widetilde{O}(\cdot)$ hides a factor of $\mathrm{polylog}(n, d, \frac{1}{\varepsilon})$.

The formal statements, with explicit running time bounds, are given in Theorem 6.1 (for sampling) and in Theorem 7.1 (for counting).

Remark. Our algorithms in Theorem 1.1 have unusual running time bounds that are controlled by a parameter ζ. The parameter ζ cannot be too large. In fact, it must be no greater than 2^{-20}, which implies that k is at least 60. As ζ gets smaller, the condition we require becomes stronger, but the sampling and counting algorithms run closer to linear and quadratic time, respectively. This is somewhat similar to algorithms for the Lovász local lemma, where the running time increases as the slack of the condition goes to 0.

In particular, if we set $\zeta = 2^{-20}$, the condition becomes $k \geq 20\log k + 20\log d + 60$. The running time of our algorithm is a fixed polynomial in n, $\frac{1}{\varepsilon}$, d, and k. Besides, for example, the exponent of n is $1+\zeta$ for sampling, which is very close to 1. In contrast, Moitra's algorithms [Moi19] for both counting and sampling require a stronger condition $k \geq 60\log d + 60\log k + 300$, and run in time $\left(\frac{n}{\varepsilon}\right)^{O(d^2k^2)}$. Our algorithms are much faster and remain in polynomial-time even if k or d is as large as $\Omega(n)$. Nonetheless, for approximate counting, Moitra's algorithm remains the only efficient *deterministic* algorithm for #Sat under conditions of this type.

Theorem 6.1 and Theorem 7.1 are in fact slightly stronger than Theorem 1.1, because in Theorem 1.1 we have simplified the condition between the exponent ζ and (k,d). For example, for $\varepsilon = 1/\mathrm{poly}(n)$, and for $\omega(1) < kd < n^{o(1)}$ in the regime above, our algorithms run in $n^{1+o(1)}$ time for sampling, and $(n/\varepsilon)^{2+o(1)}$ time for e^{ε}-approximate counting.

1.1. **Algorithm overview.** The first step of our algorithm is to mark variables. We ensure that every clause has a certain amount of marked and unmarked variables. Because every clause has sufficiently many unmarked variables, using the local lemma, we show that each individual marked variable is close to the uniform distribution. We call this local uniformity. This step so far is very similar to [Moi19].

Our goal is to sample from the marginal distribution on the marked variables. To do this, we simulate an idealized Glauber dynamics P_{Glauber} which converges to this distribution. However, this distribution is not a Gibbs distribution, and to calculate the transition probabilities becomes #**P**-hard. Our main effort is to show the following two things:

(1) P_{Glauber} mixes in $O(n\log n)$ time (Section 4);
(2) P_{Glauber} can be approximately efficiently implemented (Section 5).

To show Item 1, we use the path coupling technique by Bubley and Dyer [BD97], which requires that for two assignments X_t and Y_t that differ on only one variable v_0, the expected difference of X_{t+1} and Y_{t+1} after one step of P_{Glauber} is less than 1. For a marked variable $v \neq v_0$, let μ_v^X be the Gibbs distribution conditioned on X_t minus the assignment of v. In other words, μ_v^X is defined over assignments to all unmarked variables and v. Define μ_v^Y similarly. Consider a disagreement coupling C_v between μ_v^X and μ_v^Y, constructed greedily starting from v_0. The crucial observation is that, the probability that v cannot be coupled is upper bounded by the probability that v is in the discrepancy set of C_v. Similar couplings have been defined by Moitra [Moi19]. (To get a better condition on our parameters, we actually follow the adaptive version in [GLLZ19].) We then define a different disagreement coupling C over all variables other than v_0, marked and unmarked alike, so that the expected difference of X_{t+1}

and Y_{t+1} is upper bounded by the expected size of the discrepancy set of C. This upper bound is shown by yet another coupling between the two couplings C_v and C.

Finally, we show that the expected size of the discrepancy set of C (not including v_0) is less than 1. Here we need a new argument based on counting induced paths to analyze these greedy disagreement couplings. This is because the old analysis based on the so-called $\{2,3\}$-trees [Moi19, GLLZ19], which was used to show these couplings terminate in $O(\log n)$ steps with high probability, can only get a constant bound in the form of $O(dk)$ on this expectation, and thus is no longer strong enough.

To show Item 2, we first observe that due to local uniformity, at any step of P_{Glauber}, unmarked variables are scattered into small connected components. This has been observed before [Moi19, GLLZ19]. However, these components can have size as large as $\Omega(dk \log n)$. Thus, a brute force enumeration would take time $n^{\Omega(dk)}$, which is too slow to our need. Instead, we employ the local lemma again to show that a random assignment on these components satisfy all relevant clauses with probability roughly $\Omega(n^{-\zeta})$. Thus, a naive rejection sampling has expected running time $O(n^{\zeta})$, which results in the small overhead in Theorem 1.1. Moreover, at the end of the algorithm, we need to sample all unmarked variables, this is done by the same rejection sampling method.

So far we have explained our sampling algorithm. For counting, we use the simulated annealing method [BŠVV08, ŠVV09, Hub15, Kol18]. First we define a suitable Gibbs distribution, which can be viewed as a product distribution conditioned on a new formula Φ' being satisfied. Then our sampling algorithm can be adapted with minimal changes. With the Gibbs distribution and its sampling algorithm, adaptive annealing can be applied to yield fast algorithms already. Instead, we show that a simpler non-adaptive annealing procedure provides similar time bounds. Note that in general non-adaptive annealing is provably slower than the adaptive version [ŠVV09]. The local lemma once again plays an important role to obtain necessary properties for a fast non-adaptive annealing procedure.

In [GLLZ19], a notion called "pre-Gibbs distribution" was introduced. Its samples are pairs (S, σ_S) where S is a random subset of variables and σ_S is an assignment of S. The main requirement is that if we sample from the Gibbs distribution conditioned on σ_S, the resulting sample follows the desired Gibbs distribution. Our algorithm here is a realization of sampling from the pre-Gibbs distribution, where S is fixed a priori. It remains interesting to explore this idea of "pre-Gibbs sampling", where we should allow a dynamic S. With a dynamic S, we may get rid of the marking process by incorporating the adaptive coupling idea of [GLLZ19], which can greatly improve our assumption in Theorem 1.1.

1.2. **Related work.** The most relevant work is the algorithm by Moitra [Moi19], which we have discussed and compared with in detail above. Moitra's work is subsequently refined and adapted to hypergraph colorings [GLLZ19], but it still suffers from the same slow running time. The partial rejection sampling (PRS) method [GJL19] also works in the local lemma setting. However, for CNF formulas, PRS requires more complicated structural conditions in addition to the ones relating k and d.

Prior to our work, no Markov chain algorithm is known to work in the local lemma parameter regimes for #Sat, mainly because of the connectivity barrier. For monotone k-CNF formulas, where connectivity is not an issue, Hermon et al. [HSZ19] showed that the (straightforward) Glauber dynamics mixes in $O(n \log n)$ time if $k \geq 2 \log d + C$ for some constant C, which is tight up to the constant C due to complementing hardness results [BGG+19]. For proper colorings over simple hypergraphs, Frieze and Anastos [FA17] showed that a slight variant of the straightforward Glauber dynamics mixes rapidly under conditions almost match the local lemma. However, their work also requires that the vertex degrees are at least $\Omega(\log n)$ to ensure that the giant connected component occupies a $1 - n^{-c}$ fraction of the whole state space. In comparison, although our algorithm is also based on Markov chains, we completely bypassed the connectivity issue.

Deterministic approximate counting algorithms often run in time $n^{f(\Delta)}$ where Δ is some parameter, such as the maximum degree of vertices in a graph, and $f(\Delta) \to \infty$ as $\Delta \to \infty$. This is not desirable and is not polynomial-time if $\Delta = \omega(1)$. Recently, there has been some effort to bring down such running times (often using randomized techniques like Markov chains) to achieve polynomial running time with fixed exponents for all Δ. Examples include the work of Efthymiou et al. [EHS+19] for counting independent sets [Wei06], and the work of Chen et al. [CGG+19] for the algorithmic Pirogov-Sinai theory [JKP19, HPR19].

2. Preliminaries

2.1. **Notations.** Let $\Phi = (V, C)$ be a CNF formula, where V is the set of Boolean variables and C is the set of clauses. For each clause $c \in C$, we use

$$\mathsf{vbl}(c) \triangleq \{y \in V \mid y \text{ or } \neg y \text{ appears in } c\} \tag{1}$$

to denote the set of variables that appear in c. We say a CNF formula Φ is k-uniform if each clause contains exactly k literals on distinct variables, i.e. $|\mathsf{vbl}(c)| = k$ for all $c \in C$. For any $c \in C$ and $x \in \mathsf{vbl}(c)$, we assume only one of the literal in $\{x, \neg x\}$ appears in c. Otherwise, the clause c can always be satisfied. We also assume that each variable belongs to at most d distinct clauses. Let μ denote the uniform distribution over all satisfying assignments for Φ. Our goal is to draw from a distribution close enough to μ.

We often model the CNF formula $\Phi = (V, C)$ as a hypergraph

$$H_\Phi \triangleq (V, \mathcal{E}), \tag{2}$$

where the vertices in H_Φ are variables in Φ and the hyperedges are defined as $\mathcal{E} \triangleq \{\mathsf{vbl}(c) \mid c \in C\}$.

We write log to denote $\log_2$ and ln to denote $\log_e$. We also write $\exp(s)$ to denote e^s, especially when s is a complicated expression. We use **Pr** without subscript to denote the probability space generated by the algorithm in the context, and use subscript to clarify other probability spaces.

2.2. **Lovász local lemma.** Let $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ be a collection of mutually independent random variables. For any event E, denote by $\mathsf{vbl}(E) \subseteq \mathcal{R}$ the set of variables determining E. In other words, changing the values of variables outside of $\mathsf{vbl}(E)$ does not change the truth value of E. Let $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$ be a collection of "bad" events. For each event $B \in \mathcal{B}$, we define $\Gamma(B) \triangleq \{B' \in \mathcal{B} \mid B' \neq B \text{ and } \mathsf{vbl}(B') \cap \mathsf{vbl}(B) \neq \varnothing\}$. For any event $A \notin \mathcal{B}$ and its determining variables $\mathsf{vbl}(A) \subseteq \mathcal{R}$, we define $\Gamma(A) \triangleq \{B \in \mathcal{B} \mid \mathsf{vbl}(A) \cap \mathsf{vbl}(B) \neq \varnothing\}$. Let $\mathbf{Pr}_{\mathcal{P}}[\cdot]$ denote the product distribution of variables in $\mathcal{R}$. The following version of the Lovász local lemma is from [HSS11].

Theorem 2.1. *If there is a function $x : \mathcal{B} \to (0, 1)$ such that for any $B \in \mathcal{B}$,*

$$\mathbf{Pr}_{\mathcal{P}}[B] \leq x(B) \prod_{B' \in \Gamma(B)} (1 - x(B')), \tag{3}$$

then it holds that

$$\mathbf{Pr}_{\mathcal{P}}\left[\bigwedge_{B \in \mathcal{B}} \overline{B}\right] \geq \prod_{B \in \mathcal{B}} (1 - x(B)) > 0.$$

Thus, there exists an assignment of all variables that avoids all the bad events.

Moreover, for any event A, it holds that

$$\mathbf{Pr}_{\mathcal{P}}\left[A \,\middle|\, \bigwedge_{B \in \mathcal{B}} \overline{B}\right] \leq \mathbf{Pr}_{\mathcal{P}}[A] \prod_{B \in \Gamma(A)} (1 - x(B))^{-1}.$$

The next corollary follows from Theorem 2.1.

Corollary 2.2. *Let $\Phi = (V, C)$ be a CNF formula. Assume each clause contains at least k_1 variables and at most k_2 variables, and each variable belongs to at most d clauses. For any $s \geq k_2$, if $2^{k_1} \geq 2eds$, then there exists a satisfying assignment for Φ and for any $v \in V$,*

$$\max\left\{\mathbf{Pr}_{X \sim \mu}[X(v) = 0], \mathbf{Pr}_{X \sim \mu}[X(v) = 1]\right\} \leq \frac{1}{2} \exp\left(\frac{1}{s}\right),$$

where μ is the uniform distribution of all satisfying assignments for Φ.

Proof. Let $\mathbf{Pr}_{\mathcal{P}}[\cdot]$ denote the product distribution that every variable in V takes a value from $\{0, 1\}$ uniformly and independently. We define a collection of bad events B_c for each $c \in C$, where B_c represents the clause c is not satisfied. For each $c \in C$, we take $x(B_c) = \frac{1}{2ds}$. Thus, for any clause $c \in C$, we have

$$\mathbf{Pr}_{\mathcal{P}}[B_c] \leq \left(\frac{1}{2}\right)^{k_1} \leq \frac{1}{2eds}.$$

To verify (3), note that for any $y > 1$, it holds that $\left(1 - \frac{1}{y}\right)^{y-1} \geq \frac{1}{\mathrm{e}}$. Since $s \geq k_2$ and $|\Gamma(B_c)| \leq (d-1)k_2 \leq 2ds - 1$ for all $c \in C$, We have

$$\mathbf{Pr}_{\mathcal{P}}[B_c] \leq \frac{1}{2ds}\left(1 - \frac{1}{2ds}\right)^{2ds-1} \leq \frac{1}{2ds}\left(1 - \frac{1}{2ds}\right)^{|\Gamma(B_c)|} = x(B_c)\prod_{b\in\Gamma(B_c)}(1 - x(B_b)).$$

Hence, there exists a satisfying assignment for CNF formula Φ. For any variable $v \in V$, let B_v denote the event that v takes the value 0. Note that $|\Gamma(B_v)| = d$. By Theorem 2.1, we have

$$\mathbf{Pr}_{X\sim\mu}[X(v) = 1] \leq \frac{1}{2}\left(1 - \frac{1}{2ds}\right)^{-d} \leq \frac{1}{2}\exp\left(\frac{1}{s}\right).$$

Similarly, we have $\mathbf{Pr}_{X\sim\mu}[X(v) = 0] \leq \frac{1}{2}\exp\left(\frac{1}{s}\right)$. □

The Moser-Tardos algorithm [MT10] constructs an assignment of all random variables in $\mathcal{P}$ that avoids all the bad events in $\mathcal{B}$. The Moser-Tardos algorithm is given in Algorithm 1.

Algorithm 1: The Moser-Tardos algorithm

```
for each R ∈ ℛ, sample v_R independently according to the distribution of R;
while there exists a bad event B ∈ ℬ s.t. B occurs do
    pick an arbitrary B ∈ ℬ s.t. B occurs;
    resample the value of v_R for all variables R ∈ vbl(B);
return (v_R)_{R∈ℛ}
```

Proposition 2.3 (Moser and Tardos [MT10])**.** *Suppose the asymmetric local lemma condition (3) in Theorem 2.1 holds with the function $x : \mathcal{B} \to (0, 1)$. Upon termination, the Moser-Tardos algorithm returns an assignment that avoids all the bad events. The expected total resampling steps for Moser-Tardos algorithm is at most $\sum_{B\in\mathcal{B}} \frac{x(B)}{1-x(B)}$.*

2.3. **Coupling and mixing times for Markov chains.** Let μ and ν be two probability distributions over the same space Ω. The total variation distance is defined by

$$d_{\mathrm{TV}}(\mu, \nu) \triangleq \frac{1}{2}\sum_{x\in\Omega}|\mu(x) - \nu(x)|.$$

If we have a random variable X whose law is ν, we may write $d_{\mathrm{TV}}(\mu, X)$ instead of $d_{\mathrm{TV}}(\mu, \nu)$ to simplify the notation.

A coupling C of μ and ν is a joint distribution over $\Omega\times\Omega$ such that projecting on the first (or second) coordinate is μ (or ν). A well known inequality regarding coupling is the following.

Proposition 2.4. *Let C be an arbitrary coupling of μ and ν. Then*

$$d_{\mathrm{TV}}(\mu, \nu) \leq \mathbf{Pr}_{(x,y)\sim C}[x \neq y].$$

Moreover, there exists an optimal coupling that achieves equality.

A Markov chain $(X_t)_{t\geq 0}$ over a state space Ω is given by its transition matrix $P : \Omega \times \Omega \to \mathbb{R}_{\geq 0}$. A Markov chain P is called *irreducible* if for any $X, Y \in \Omega$, there exists an integer t such that $P^t(X, Y) > 0$. A Markov chain P is called *aperiodic* if for any $X \in \Omega$, it holds that $\gcd\{t \mid P^t(X, X) > 0\} = 1$. We say the distribution μ over Ω is the *stationary distribution* of a Markov chain P if $\mu = \mu P$. A Markov chain P is *reversible* with respect to μ if it satisfies the detailed balance condition

$$\mu(X)P(X, Y) = \mu(Y)P(Y, X),$$

which implies that μ is a stationary distribution of P. If a Markov chain P is irreducible and aperiodic, then it converges to the unique stationary distribution μ. The *mixing time* of a Markov chain P with stationary distribution μ is defined by

$$T_{\mathrm{mix}}(P, \delta) \triangleq \max_{X_0\in\Omega}\min\left\{t : d_{\mathrm{TV}}\left(P^t(X_0, \cdot), \mu\right) \leq \delta\right\}. \tag{4}$$

See the textbook [LP17] for more details and backgrounds on Markov chains and mixing times.

Consider an irreducible and aperiodic Markov chain specified by the transition matrix P. A coupling of the Markov chain is a joint process $(X_t, Y_t)_{t\geq 0}$ such that both $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$ individually follow the transition rule of P, and if $X_t = Y_t$ then $X_s = Y_s$ for all $s \geq t$. The total variation distance between $P^t(X_0, \cdot)$ and μ can be bounded by $\max_{X_0\in\Omega} d_{\mathrm{TV}}\left(P^t(X_0,\cdot), \mu\right) \leq \max_{X_0,Y_0\in\Omega^2} \mathbf{Pr}\left[X_t \neq Y_t\right]$.

Path coupling [BD97] is a powerful technique to construct couplings of Markov chains. In this paper, we use the following path coupling lemma, which is simplified for the Boolean hypercube. Let the state space $\Omega = \{0,1\}^N$ for some integer $N \geq 1$. For any $X, Y \in \Omega$, define the *Hamming distance* between X, Y as

$$d_{\mathrm{Ham}}(X, Y) \triangleq \left|\{1 \leq i \leq N \mid X(i) \neq Y(i)\}\right|.$$

Proposition 2.5 ([BD97]). *Let $\Omega = \{0,1\}^N$ some integer $N \geq 1$. Let $P : \Omega\times\Omega \to \mathbb{R}_{\geq 0}$ be the transition matrix of an irreducible and aperiodic Markov chain. Suppose there is a coupling $(X, Y) \to (X', Y')$ of the Markov chain defined for all $X, Y \in \Omega$ with $d_{Ham}(X, Y) = 1$, which satisfies*

$$\mathbf{E}\left[d_{Ham}(X', Y') \mid X, Y\right] \leq 1 - \lambda,$$

for some $0 < \lambda < 1$. Then the mixing time of the Markov chain is bounded by

$$T_{mix}(P, \delta) \leq \frac{1}{\lambda}\log\left(\frac{N}{\delta}\right).$$

3. The sampling algorithm

Let $\Phi = (V, C)$ be a k-uniform CNF formula, in which each variable belongs to at most d clauses. In this section we give our Markov chain based algorithm to sample satisfying assignments almost uniformly at random.

3.1. Marking variables.

Our algorithm first marks a set of marked variables $\mathcal{M} \subseteq V$. We say a variable $v \in V$ is *marked* if $v \in \mathcal{M}$, or v is *unmarked* if $v \notin \mathcal{M}$. We will ensure the following condition for the set of marked variables $\mathcal{M}$, where $k_\alpha \geq 1$ and $k_\beta \geq 1$ are two integer parameters to be specified later satisfying $k_\alpha + k_\beta \leq k$.

Condition 3.1. *Each clause has at least k_α marked variables and at least k_β unmarked variables.*

We use the Moser-Tardos algorithm, Algorithm 1, to find $\mathcal{M}$. Define $0 \leq \alpha, \beta \leq 1$ as

$$\alpha \triangleq \frac{k_\alpha}{k}, \qquad \beta \triangleq \frac{k_\beta}{k}.$$

Suppose we mark each variable independently with probability $\frac{1+\alpha-\beta}{2}$. For each clause $c \in C$, let M_c be the bad event that "c has less than k_α marked variables or less than k_β unmarked variables". The lemma below follows from Proposition 2.3 and verifying (3).

Lemma 3.2. *Assume $2^k \geq (2edk)^{\frac{6\ln 2\cdot(1+\alpha-\beta)}{(1-\alpha-\beta)^2}}$. There is an algorithm such that for any $\delta > 0$, with probability at least $1-\delta$, it returns a set of marked variables satisfying Condition 3.1 with time complexity $O\left(dkn\log\frac{1}{\delta}\right)$, where $n = |V|$ is the number of variables.*

Proof. To apply Algorithm 1, Let $\mathbf{Pr}_{\mathcal{P}}[\cdot]$ denote the product distribution that every variable is marked independently with probability $\frac{1+\alpha-\beta}{2}$. By Chernoff bound [MU17, Corollary 4.6], we have

$$\forall c \in C: \quad \mathbf{Pr}_{\mathcal{P}}\left[M_c\right] \leq 2\exp\left(-\frac{(1-\alpha-\beta)^2}{6(1+\alpha-\beta)}\cdot k\right) = 2\left(\frac{1}{2}\right)^{\frac{(1-\alpha-\beta)^2}{6\ln 2\cdot(1+\alpha-\beta)}\cdot k}.$$

We define a function x as $x(M_c) \triangleq \frac{1}{dk}$ for all $c \in C$. We have for all $c \in C$,

$$\mathbf{Pr}_{\mathcal{P}}\left[M_c\right] \leq \frac{1}{edk} \leq x(M_c)\prod_{M_b\in\Gamma(M_c)}(1 - x(M_b)).$$

Since the total number of clauses is at most dn, by Proposition 2.3, the expected number of resampling steps is at most

$$\sum_{c \in C} \frac{x(M_c)}{1 - x(M_c)} \leq \frac{2n}{k}. \tag{5}$$

By Markov inequality, if we run Algorithm 1 for at most $\frac{4n}{k}$ resampling steps, the algorithm returns the set $\mathcal{M}$ with probability at least $\frac{1}{2}$. If we run $\lceil \log \frac{1}{\delta} \rceil$ Moser-Tardos algorithms independently, then with probability at least $1 - \delta$, one of them finds the set $\mathcal{M}$ within $\frac{4n}{k}$ resampling steps.

Note that in each resampling step, we need to resample k variables and check whether dk bad event occurs, and the cost of checking one event is at most k. Hence, the total time complexity is $O\left(ndk \log \frac{1}{\delta}\right)$. □

We note that much better concentration bound is known to the Moser-Tardos algorithm [HH17]. However, Lemma 3.2 is sufficient to our need.

We use the algorithm in Lemma 3.2 with $\delta = \frac{\varepsilon}{4}$ to construct the set of marked variables $\mathcal{M} \subseteq V$. If the algorithm fails to construct $\mathcal{M}$, then our algorithm terminates immediately and outputs an arbitrary assignment $X \in \{0,1\}^V$. This bad event occurs with probability at most $\frac{\varepsilon}{4}$. In the rest of this section, we assume that the set of marked variables $\mathcal{M} \subseteq V$ is already found.

3.2. **The main algorithm.** In this section we present our algorithm for sampling satisfying assignments of CNFs. We will need some notations first. For an arbitrary set of variables $S \subseteq V$, let μ_S be the marginal distribution on S induced from μ. Formally,

$$\forall \sigma \in \{0,1\}^S : \quad \mu_S(\sigma) = \sum_{\tau \in \{0,1\}^V,\ \tau(S) = \sigma} \mu(\tau).$$

When $S = \{v\}$ for some $v \in V$, we also write μ_v instead of $\mu_{\{v\}}$. Moreover, for a partial assignment $X \in \{0,1\}^\Lambda$ where $\Lambda \subset V$ and $S \cap \Lambda = \varnothing$, let $\mu_S^X(\cdot) := \mu_S(\cdot \mid X)$ be the marginal distribution on S conditioned on the partial assignment on Λ is X.

The main idea of our sampling algorithm is to simulate a Markov chain whose stationary distribution is the marginal distribution $\mu_{\mathcal{M}}$ on $\mathcal{M}$. Let P_{Glauber} be the idealized Glauber dynamics for the marked variables. Namely, we start with an initial assignment $X_0 \in \{0,1\}^{\mathcal{M}}$ where $X_0(v)$ is uniformly at random for all $v \in \mathcal{M}$. In the t-th step, the chain evolves as follows:

- pick $v \in \mathcal{M}$ uniformly at random and set $X_t(u) \leftarrow X_{t-1}(u)$ for all $u \in \mathcal{M} \setminus \{v\}$;
- sample $X_t(v) \in \{0,1\}$ from the distribution $\mu_v(\cdot \mid X_{t-1}(\mathcal{M} \setminus \{v\}))$.

This chain is reversible with respect to $\mu_{\mathcal{M}}$, as for any $X, Y \in \{0,1\}^{\mathcal{M}}$ that differ on only v,

$$\begin{aligned} \mu_{\mathcal{M}}(X) P_{\text{Glauber}}(X,Y) &= \frac{1}{|M|} \cdot \mu_{\mathcal{M}}(X) \mu_v(Y(v) \mid X(\mathcal{M} \setminus \{v\})) = \frac{1}{|M|} \cdot \frac{\mu_{\mathcal{M}}(X)\mu_{\mathcal{M}}(Y)}{\mu_{\mathcal{M}\setminus\{v\}}(X(\mathcal{M} \setminus \{v\}))} \\ &= \frac{1}{|M|} \cdot \mu_{\mathcal{M}}(Y) \mu_v(X(v) \mid Y(\mathcal{M} \setminus \{v\})) = \mu_{\mathcal{M}}(Y) P_{\text{Glauber}}(Y,X). \end{aligned} \tag{6}$$

We will show that P_{Glauber} is both irreducible and aperiodic in our parameter regimes. We simulate this chain to obtain a random assignment $X_{\mathcal{M}} \in \{0,1\}^{\mathcal{M}}$ whose distribution is close enough to $\mu_{\mathcal{M}}$. Then the algorithm samples a random assignment $X_{V \setminus \mathcal{M}} \in \{0,1\}^{V \setminus \mathcal{M}}$ for unmarked variables from the distribution $\mu_{V \setminus \mathcal{M}}(\cdot \mid X_{\mathcal{M}})$. The final sample is $X_{\mathsf{alg}} \triangleq X_{\mathcal{M}} \cup X_{V \setminus \mathcal{M}}$.

This chain P_{Glauber} is an idealized process because the transitions of the chain rely on evaluating some nontrivial marginal probabilities, which in general can be as hard as the problem of counting the number of satisfying assignments itself. To efficiently simulate one step of the Markov chain and to complete the random assignments for unmarked variables, we need to sample from the marginal distributions $\mu_v(\cdot \mid X_{t-1}(\mathcal{M} \setminus \{v\}))$ and $\mu_{V \setminus \mathcal{M}}(\cdot \mid X_T)$, where $t \leq T$ and T is an upper bound of the mixing time of P_{Glauber}. We will use a subroutine Sample$(\cdot)$ for this. Given an assignment $X \in \{0,1\}^\Lambda$ on the subset $\Lambda \subseteq \mathcal{M}$ and a subset $S \subseteq V \setminus \Lambda$ of variables, the subroutine Sample(Φ, δ, X, S) returns a random assignment $Y \in \{0,1\}^S$ from the distribution $\mu_S(\cdot \mid X)$ upon success. We will ensure that Sample(Φ, δ, X, S) is efficient and when we call it in Algorithm 2, it returns a sample within total

variation distance δ to the desired distribution with probability at least $1-\delta$ for a small $\delta > 0$. This is because due to Corollary 2.2 and its variants, the marked variables are almost uniform, and conditioned on any almost uniform assignment of (almost all) marked variables, the remaining formula splits into many disjoint small connected components.

The whole sampling algorithm is formally described in Algorithm 2.

Algorithm 2: The sampling algorithm

input : a CNF formula $\Phi = (V, C)$, a parameter $\varepsilon > 0$, and a set of marked variables $\mathcal{M}$.
output: a random assignment $X_{\mathsf{alg}} \in \{0,1\}^V$.

1 sample $X_0(v) \in \{0,1\}$ uniformly and independently for each $v \in \mathcal{M}$;
2 **for** *each* t *from* 1 *to* $T := \left\lceil 2n \log \frac{4n}{\varepsilon} \right\rceil$ **do**
3 choose variable $v \in \mathcal{M}$ uniformly at random;
/* resample $X_{\mathcal{M}}(v)$ from the distribution $\mu_v(\cdot \mid X_{t-1}(\mathcal{M} \setminus \{v\}))$ */
4 $X_t(v) \leftarrow \mathsf{Sample}(\Phi, \frac{\varepsilon}{4(T+1)}, X_{t-1}(\mathcal{M} \setminus \{v\}), \{v\})$;
5 $\forall u \in \mathcal{M}$ and $u \neq v$, $X_t(u) \leftarrow X_{t-1}(u)$;
/* sample $X_{V \setminus \mathcal{M}}$ from the distribution $\mu_{V \setminus \mathcal{M}}(\cdot \mid X_T)$ */
6 $X_{V \setminus \mathcal{M}} \leftarrow \mathsf{Sample}(\Phi, \frac{\varepsilon}{4(T+1)}, X_T, V \setminus \mathcal{M})$;
7 **return** $X_{\mathsf{alg}} = X_T \cup X_{V \setminus \mathcal{M}}$;

In Algorithm 2, Sample$(\cdot)$ appears in Line 4 and Line 6 and returns random assignments on $\{v\}$ and $V \setminus \mathcal{M}$ respectively. In our implementation, we allow their distributions to be slightly biased (controlled by the parameter $\delta = \frac{\varepsilon}{4(T+1)}$).

The correctness and the efficiency of Algorithm 2 rely on three facts:

(1) the Glauber dynamics for marked vertices is rapidly mixing;
(2) the Sample$(\cdot)$ subroutine for unmarked vertices is efficient;
(3) the small bias in the distribution caused by Sample$(\cdot)$ does not affect the final distribution much.

The rapid mixing property of the Glauber dynamics is analyzed in Section 4. Details of Sample$(\cdot)$ will be given in Section 3.3 and its analysis in Section 5.

We will ensure that, with high probability, Sample$(\cdot)$ returns samples whose distributions are close to the desired ones in both Line 4 and Line 6. Using this, we will show that Algorithm 2 couples with high probability with the idealized chain P_{Glauber}. As a result, the distribution of the random assignment X_{alg} returned by Algorithm 2 is close to $\mu(\cdot)$.

Lemma 3.3. *Suppose* $2^{k_\alpha} \geq 4\mathrm{e}^2 d^2 k^2$, $2^{k_\beta} \geq 2^{16} d^9 k^9$, *and* $\mathcal{M}$ *satisfying Condition 3.1 has been found. The random assignment* $X_{\mathsf{alg}} \in \{0,1\}^V$ *returned by Algorithm 2 satisfies*

$$d_{\mathrm{TV}}\left(X_{\mathsf{alg}}, \mu\right) \leq \frac{3\varepsilon}{4}. \tag{7}$$

Lemma 3.3 is proved in Section 6.

3.3. **The** Sample **subroutine.** Here we give the subroutine Sample(Φ, δ, X, S), where $X \in \{0,1\}^\Lambda$ is an assignment on subset $\Lambda \subseteq \mathcal{M}$ and $S \subseteq V \setminus \Lambda$ is a subset of variables. The output of the subroutine is a random assignment $Y \in \{0,1\}^S$, which ideally should follow the conditional marginal distribution $\mu_S(\cdot \mid X)$. However, in order for the efficiency of the subroutine, some small error is tolerated.

Our basic idea is to find all connected components of a new formula Φ^X. We will show that in the execution of Algorithm 2, these components are sufficiently small. Then we will use rejection sampling on them independently for each component.

Let us first define Φ^X and its connected components. Given a CNF formula $\Phi = (V, C)$ and a partial assignment $X \in \{0,1\}^\Lambda$ for some $\Lambda \subseteq V$, we simplify Φ under X to obtain $\Phi^X = (V^X, C^X)$. Formally, we have

- $V^X = V \setminus \Lambda$, and

- C^X is obtained from C by removing all clauses that has been satisfied under X[1] and removing the appearance of x or $\neg x$ from the remaining unsatisfied clauses for every $x \in \Lambda$.

Recall that

$$\forall \sigma \in \{0,1\}^{V^X} = \{0,1\}^{V\setminus\Lambda}: \quad \mu^X_{V\setminus\Lambda}(\sigma) = \mu_{V\setminus\Lambda}(\sigma \mid X). \tag{8}$$

It is straightforward to check that $\mu^X_{V\setminus\Lambda}$ is the uniform distribution over all satisfying assignments of Φ^X. Let $H_{\Phi^X} = (V^X, \mathcal{E}^X)$ be the hypergraph defined in (2) for the CNF formula Φ^X. Let $H^X_i = (V^X_i, \mathcal{E}^X_i)$ for $1 \le i \le \ell$ denote all the connected components in the hypergraph H_{Φ^X}, where ℓ is the number of connected components. Each $H^X_i = (V^X_i, \mathcal{E}^X_i)$ represents a CNF formula $\Phi^X_i = (V^X_i, C^X_i)$, where

$$C^X_i \triangleq \left\{ c \in C^X \mid \text{clause } c \text{ is represented by a hyperedge in } \mathcal{E}^X_i \right\}.$$

We have $\Phi^X = \Phi^X_1 \wedge \Phi^X_2 \wedge \cdots \wedge \Phi^X_\ell$, and all the V^X_i are disjoint. Let μ^X_i be the uniform distribution on all satisfying assignments of Φ^X_i for every $i = 1, \ldots, \ell$, then $\mu^X_{V\setminus\Lambda}(\cdot)$ is the product distribution of all μ^X_i.

Obviously, the distribution $\mu_S(\cdot \mid X)$ is determined by only those connected components intersecting S. Without loss of generality, we assume that $S \cap V^X_i \neq \emptyset$ for $1 \le i \le m$ and $S \cap V^X_i = \emptyset$ for $m < i \le \ell$. To draw a random assignment $Y \in \{0,1\}^S$ from the distribution $\mu_S(\cdot \mid X)$, we independently draw a random assignment Y_i from $\mu^X_i(\cdot)$ for each $1 \le i \le m$. Let

$$Y' \triangleq \bigcup_{i=1}^{m} Y_i.$$

Note that $S \subseteq \bigcup_{i=1}^m V_i$. Our sample Y is the projection of Y' on S, namely $Y = Y'(S)$. It is easy to verify that Y follows the marginal distribution on S induced by $\mu^X_{V\setminus\Lambda}$. By (8), the random assignment Y follows the distribution $\mu_S(\cdot \mid X)$.

To draw from individual $\mu^X_i(\cdot)$ for each $1 \le i \le m$, we can simply use the naive rejection sampling: repeatedly draw uniform assignments on $\{0,1\}^{V^X_i}$ and return the first one that satisfies Φ^X_i. This should terminate fast if the connected component $\mathcal{E}^X_i$ is small.

Our implementation of Sample(Φ, δ, X, S) is then clear: it tries for each Φ^X_i, $1 \le i \le m$, to repeatedly draw uniform assignments for at most R (to be suitably fixed) times and return the first satisfying one. Bad events happen if for one of the components, say Φ^X_i, the size of Φ^X_i is too large or all R trials fail to satisfy Φ^X_i, in which case an arbitrary assignment on S is returned.

Formally, let $0 < \eta < 1$ satisfy

$$2^{k_\beta} \ge \frac{20}{\eta} edk, \tag{9}$$

and define

$$R \triangleq \left\lceil \left(\frac{n}{\delta}\right)^{\frac{\eta}{10}} \log \frac{n}{\delta} \right\rceil.$$

In the subroutine Sample(Φ, δ, X, S), we

- check the size $|\mathcal{E}^X_i|$ for all $1 \le i \le m$, if there exists $|\mathcal{E}^X_i| > dk \log \frac{n}{\delta}$, then the subroutine terminates and returns a $Y \in \{0,1\}^S$ uniformly at random;
- for each $1 \le i \le m$, use the naive rejection sampling for at most R times to draw a random assignment Y^X_i from the distribution μ^X_i; if there exists $1 \le i \le m$ such that the subroutine fails to draw a Y^X_i from μ^X_i after R rejection sampling trials, then the subroutine terminates and returns a $Y \in \{0,1\}^S$ uniformly at random.

The subroutine Sample(Φ, δ, X, S) is described in Algorithm 3.

The following proposition is a basic property of rejection sampling.

[1]Let $c \in C$ be a clause in Φ. We say c is satisfied under the (partial) assignment X if any literal of c is already assigned true.

Algorithm 3: Sample(Φ, δ, X, S)

Input : a CNF formula $\Phi = (V, C)$, a parameter $0 < \delta, \eta < 1$, an assignment $X \in \{0,1\}^{\Lambda}$ for some $\Lambda \subseteq V$, a set of variables $S \subseteq V \setminus \Lambda$, and $n = |V|$.

Output : a random assignment $Y \in \{0,1\}^{S}$.

1 simplify Φ under X and obtain a new formula Φ^X;

2 find all the connected components $\left\{H_i^X = (V_i^X, \mathcal{E}_i^X) \mid 1 \le i \le m\right\}$ in H_{Φ^X} s.t. each $V_i^X \cap S \neq \emptyset$;

3 **if** *there exists* $1 \le i \le m$ *s.t.* $|\mathcal{E}_i^X| > dk \log \frac{n}{\delta}$ **then**

4 **return** *an assignment* $Y \in \{0,1\}^S$ *uniformly at random;*

5 **for** *each* i *from 1 to* m **do**

6 let $\Phi_i^X = (V_i^X, C_i^X)$ be the CNF formula represented by $H_i^X = (V_i^X, \mathcal{E}_i^X)$;

7 $Y_i^X \leftarrow \mathsf{RejectionSampling}\left(\Phi_i^X, R\right)$, where $R = \left\lceil \left(\frac{n}{\delta}\right)^{\frac{\eta}{10}} \log \frac{n}{\delta} \right\rceil$;

8 **if** $Y_i^X = \perp$ **then**

9 **return** *an assignment* $Y \in \{0,1\}^S$ *uniformly at random;*

10 **return** $Y = Y'(S)$, *where* $Y' = \bigcup_{i=1}^{m} Y_i^X$;

Algorithm 4: RejectionSampling(Φ, R)

Input : a CNF formula $\Phi = (V, C)$, a parameter $R > 0$.

Output : a random assignment $Y \in \{0,1\}^V$ or a special symbol $\perp$.

1 **for** *each* i *from 1 to* R **do**

2 sample $Y \in \{0,1\}^V$ uniformly and independently;

3 **if** *all the clauses in* C *are satisfied by* Y **then**

4 **return** Y;

5 **return** $\perp$;

Proposition 3.4. *In the subroutine* Sample(Φ, δ, X, S)*, conditioned on that the random assignment* $Y \in \{0,1\}^S$ *is returned in Line 10,* Y *follows the law* $\mu_S(\cdot \mid X)$.

With the CNF formula represented by a standard data structure, the running time of Sample(Φ, δ, X, S) is easily bounded by $\tilde{O}(|S| \cdot R \cdot \mathrm{poly}(d, k))$. This is rigorously analyzed in Lemma 5.1 in Section 5. In the same lemma we also prove that conditioning on that every component is small (i.e. Line 4 in Algorithm 3 is not executed), the Sample subroutine fails (i.e. Line 9 in Algorithm 3 happens) with probability at most δ. Such failure is due to the randomness of the rejection sampling. In another key lemma, Lemma 5.2 in Section 5, we prove that for any call of Sample in Algorithm 2, Line 4 in Algorithm 3 is indeed executed with probability at most δ. Such failure is due to the randomness of the input X to Sample. Overall, with probability at least $1 - \delta$, the distribution of the assignments returned by Sample(Φ, δ, X, S) is within total variation distance at most δ from $\mu_S(\cdot \mid X)$.

4. Rapid mixing of the idealized dynamics

Let $\Phi = (V, C)$ be a CNF formula. Let $\mathcal{M} \subseteq V$ be the set of marked variables satisfying Condition 3.1 and $\Omega \triangleq \{0,1\}^{\mathcal{M}}$. Let P_{Glauber} be the Glauber dynamics for marked variables, and use $(X_t)_{t \ge 0}$ to denote the state at time t where $X_t \in \{0,1\}^{\mathcal{M}}$. In this section, we show that the idealized Glauber dynamics P_{Glauber} is rapidly mixing.

Lemma 4.1. *Let* $\Phi = (V, C)$ *be a* k*-uniform CNF formula such that each variable belongs to at most* d *clauses. Suppose* $\mathcal{M} \subseteq V$ *satisfies Condition 3.1 with parameters* k_α *and* k_β*. Let* $P_{Glauber}$ *be the Glauber dynamics for marked variables. If* $2^{k_\beta} \ge 2^{16} d^9 k^9$*, then for any* $\delta > 0$*, it holds that*

$$T_{mix}(P_{Glauber}, \delta) \le \left\lceil 2n \log \frac{n}{\delta} \right\rceil,$$

where $n = |V|$ *and the mixing time* T_{mix} *is defined in (4).*

4.1. **The stationary distribution.** We first prove that the Glauber dynamics P_{Glauber} has the unique stationary distribution $\mu_{\mathcal{M}}$.

Lemma 4.2. *If $2^{k_\beta} \geq 4edk$, then the support of $\mu_{\mathcal{M}}$ is all of $\Omega = \{0,1\}^{\mathcal{M}}$, and the Glauber dynamics P_{Glauber} for marked variables has the unique stationary distribution $\mu_{\mathcal{M}}$.*

Proof. For any $v \in \mathcal{M}$ and any assignment $X' \in \{0,1\}^{\mathcal{M}\setminus\{v\}}$, we claim that

$$\forall c \in \{0,1\}: \quad \mu_v(c \mid X') > 0. \tag{10}$$

This implies that for any $X, Y \in \Omega$ with Hamming distance $d_{\text{Ham}}(X,Y) = |\{v \in \mathcal{M} \mid X(v) \neq Y(v)\}|$, it is possible to transform X to Y in $d_{\text{Ham}}(X,Y)$ steps. Hence, P_{Glauber} is irreducible. It also implies that the support of $\mu_{\mathcal{M}}$ is Ω. Besides, for any $X \in \Omega$, we have $P_{\text{Glauber}}(X,X) > 0$. Hence, this chain is aperiodic.

We now prove (10). Let $\Phi^{X'}$ be the CNF formula obtained from Φ by deleting all the clauses that are satisfied by X' and all the variables in $\mathcal{M} \setminus \{v\}$. Let μ' denote the uniform distribution of all solutions of Φ'. Then we have

$$\forall c \in \{0,1\}: \quad \mu_v(c \mid X') = \mu'_v(c).$$

In CNF formula Φ', each clause has at least k_β variables and at most k variables and each variable belongs to at most d clauses. Since $2^{k_\beta} \geq 4edk$, by Corollary 2.2, we have

$$\forall c \in \{0,1\}: \quad \mu_v(c \mid X') = \mu'_v(c) \leq \frac{1}{2}\exp\left(\frac{1}{2k}\right) \leq \frac{\sqrt{e}}{2} < 1.$$

This implies $\mu_v(c \mid X') > 0$ for all $c \in \{0,1\}$.

By the update rule of the Glauber dynamics chain, it is easy to verify the following detailed balance condition as in (6):

$$\forall X, Y \in \Omega: \quad \mu_{\mathcal{M}}(X) P_{\text{Glauber}}(X,Y) = \mu_{\mathcal{M}}(Y) P_{\text{Glauber}}(Y,X).$$

Since the Markov chain is irreducible and aperiodic, this proves that the Markov chain $(X_t)_{t\geq 0}$ has the unique stationary distribution $\mu_{\mathcal{M}}$. □

Hence, under the condition in Lemma 4.1, P_{Glauber} has unique stationary distribution $\mu_{\mathcal{M}}$.

4.2. **The mixing time.** We next prove that P_{Glauber} is rapidly mixing provided that $2^{k_\beta} \geq 2^{16} d^9 k^9$. The mixing time in Lemma 4.1 is proved by the path coupling argument [BD97]. For any $X, Y \in \Omega$, recall their Hamming distance as

$$d_{\text{Ham}}(X,Y) \triangleq |\{v \in \mathcal{M} \mid X(v) \neq Y(v)\}|.$$

Let $X, Y \in \Omega$ be two assignments that disagree only on a single variable, namely, $d_{\text{Ham}}(X,Y) = 1$. We construct a coupling of Markov chains $(X,Y) \to (X',Y')$ satisfying

$$\mathbf{E}\left[d_{\text{Ham}}(X',Y') \mid X,Y\right] \leq 1 - \frac{1}{2n}. \tag{11}$$

Note that $d_{\text{Ham}}(X,Y) \leq n$ for all $X, Y \in \Omega$. Then Lemma 4.1 is proved by the path coupling lemma (Proposition 2.5) together with Lemma 4.2.

The coupling $(X,Y) \to (X',Y')$ is defined as follows.

Definition 4.3. Let $X, Y \in \Omega$ be two assignments that disagree only on a single variable, say $X(v_0) = 0$ and $Y(v_0) = 1$ where $v_0 \in \mathcal{M}$. Let $\mathcal{M}_v \triangleq \mathcal{M} \setminus \{v\}$ for any $v \in \mathcal{M}$. The coupling $(X,Y) \to (X',Y')$ is defined as:

- pick the same variable $v \in \mathcal{M}$ uniformly at random, and set $X'(u) = X(u)$ and $Y'(u) = Y(u)$ for all variables $u \in \mathcal{M}_v$;
- sample $(X'(v), Y'(v))$ jointly from the optimal coupling of two conditional marginal distributions $\mu_v(\cdot \mid X(\mathcal{M}_v))$ and $\mu_v(\cdot \mid Y(\mathcal{M}_v))$.

It is easy to verify that this is a valid coupling of two Markov chains. Two transitions $X \to X'$ and $Y \to Y'$ are both faithful copies of the Glauber dynamics chain. We remark that none of the couplings in this section is efficiently computable. They only serve as tools for the analysis of the Markov chain.

For each marked variable $v \in \mathcal{M}$, we define D_v as

$$D_v \triangleq d_{\mathrm{TV}}\left(\mu_v(\cdot \mid X(\mathcal{M}_v)), \mu_v(\cdot \mid Y(\mathcal{M}_v))\right). \tag{12}$$

which is the total variation distance between $\mu_v(\cdot \mid X(\mathcal{M}_v))$ and $\mu_v(\cdot \mid Y(\mathcal{M}_v))$. Moreover, since $X(\mathcal{M}_{v_0}) = Y(\mathcal{M}_{v_0})$, by (12),

$$D_{v_0} = 0.$$

By Proposition 2.4, under our coupling,

$$\mathbf{Pr}\left[X'(v) \neq Y'(v) \mid v \in \mathcal{M} \text{ is picked}\right] = D_v.$$

Hence, the expected Hamming distance between X' and Y' is at most

$$\begin{aligned}\mathbf{E}\left[d_{\mathrm{Ham}}(X', Y') \mid X, Y\right] &= 1 + \frac{1}{|\mathcal{M}|}\sum_{v\in\mathcal{M}} D_v - \frac{1}{|\mathcal{M}|} \\ &= 1 - \frac{1}{|\mathcal{M}|}\left(1 - \sum_{v\in\mathcal{M}} D_v\right). \end{aligned} \tag{13}$$

To prove the inequality in (11), it is sufficient to prove the following lemma and notice that $|\mathcal{M}| \leq n$.

Lemma 4.4. *Given two assignments $X, Y \in \Omega$ such that X and Y disagree only on a single variable $v_0 \in \mathcal{M}$, if $2^{k_\beta} \geq 2^{16} d^9 k^9$, it holds that*

$$\sum_{v\in\mathcal{M}} D_v \leq \frac{1}{2},$$

where D_v is the total variation distance defined in (12).

Combining inequality (13) and Lemma 4.4 proves inequality (11). This proves Lemma 4.1. Lemma 4.4 is shown in the next subsection.

4.3. **Analysis of the path coupling.** Let us first sketch the proof idea of Lemma 4.4. Recall that we have two assignments X and Y which differ on only v_0. In order to bound D_v for any $v \in \mathcal{M}$ and $v \neq v_0$, we construct a coupling C_v of two distributions $\mu(\cdot \mid X(\mathcal{M}_v))$ and $\mu(\cdot \mid Y(\mathcal{M}_v))$, where $\mathcal{M}_v = \mathcal{M} \setminus \{v\}$. Since C_v projected on v is a coupling between $\mu_v(\cdot \mid X(\mathcal{M}_v))$ and $\mu_v(\cdot \mid Y(\mathcal{M}_v))$, by Proposition 2.4, we have

$$D_v \leq \mathbf{Pr}_{(\sigma_X,\sigma_Y)\sim C_v}\left[\sigma_X(v) \neq \sigma_Y(v)\right].$$

A high-level description of our construction of C_v is as follows: we start from two partial assignments X and Y such that initially only the value on v_0 is set, say $X(v_0) = 0$ and $Y(v_0) = 1$. In each step, in a Breadth-First Search way, we extend the partial assignments using the optimal coupling between two marginal distributions to a new variable. At last, we obtain a set of variables $V_1^{C_v}$ which is a superset of all variables on which X and Y disagree. Therefore,

$$\mathbf{Pr}_{(\sigma_X,\sigma_Y)\sim C_v}\left[\sigma_X(v) \neq \sigma_Y(v)\right] \leq \mathbf{Pr}_{C_v}\left[v \in V_1^{C_v}\right].$$

We then construct another coupling C of distributions $\mu(\cdot \mid X(v_0))$ and $\mu(\cdot \mid Y(v_0))$ in a similar way, where $v_0 \in \mathcal{M}$ is the unique vertex on which X and Y differ. The coupling also produces a set V_1 which is a superset of all variables with different values. We carefully define the coupling C so that for every $v \in \mathcal{M} \setminus \{v_0\}$, it holds that

$$\mathbf{Pr}_{C_v}\left[v \in V_1^{C_v}\right] = \mathbf{Pr}_{C}\left[v \in V_1^{C}\right]. \tag{14}$$

Recall that $D_{v_0} = 0$. Therefore, we only need to bound D_v for those $v \in \mathcal{M} \setminus \{v_0\}$. Hence,

$$\sum_{v \in \mathcal{M}} D_v = \sum_{v \in \mathcal{M} \setminus \{v_0\}} D_v \le \sum_{v \in \mathcal{M} \setminus \{v_0\}} \mathbf{Pr}_{C_v}\left[v \in V_1^{C_v}\right]$$

$$\text{(by (14))} \qquad = \sum_{v \in \mathcal{M} \setminus \{v_0\}} \mathbf{Pr}_{C}\left[v \in V_1^{C}\right] = \mathbf{E}_C\left[\left|V_1^C\right|\right] - 1.$$

Finally, we bound $\mathbf{E}_C\left[\left|V_1^C\right|\right]$ by enumerating all induced paths in the square of the line graph of H_Φ (Definition 4.12) rooted at v_0.

We describe the coupling C_v in Section 4.3.1 and the coupling C in Section 4.3.2. And finally in Section 4.3.3, Lemma 4.4 is proved by a coupling between the two couplings C_v and C.

4.3.1. *The coupling C_v.* First we define the following distribution $\nu = \nu^{(v)}$ over all assignments in $\{0,1\}^V$.

Definition 4.5. Fix a variable $v \in \mathcal{M} \setminus \{v_0\}$. Let $\nu = \nu^{(v)}$ be the distribution μ conditional on the assignment of the set $\Lambda = \mathcal{M} \setminus \{v_0, v\}$ is specified as $X(\Lambda) = Y(\Lambda)$, where $X, Y \in \{0,1\}^{\mathcal{M}}$ differ at only v_0. Formally,

$$\forall \sigma \in \{0,1\}^V : \quad \nu(\sigma) = \frac{\mathbf{1}\left[\sigma(\Lambda) = X(\Lambda)\right] \cdot \mu(\sigma)}{\sum_{\tau \in \{0,1\}^V} \mathbf{1}\left[\tau(\Lambda) = X(\Lambda)\right] \cdot \mu(\tau)}. \tag{15}$$

Note that if $2^{k_\beta} \ge 2edk$, then by Lemma 4.2, the distribution ν is well-defined.

For every $v \in \mathcal{M} \setminus \{v_0\}$, the coupling C_v generates a pair of random assignments $X^{C_v}, Y^{C_v} \in \{0,1\}^V$. The projection X^{C_v} (or Y^{C_v}) has the law ν conditioned on $X^{C_v}(v_0) = X(v_0) = 0$ (or on $Y^{C_v}(v_0) = Y(v_0) = 1$). Let $k_\gamma \ge 1$ be an integer parameter to be specified later satisfying $k_\gamma < k_\beta$ and

$$2^{k_\beta - k_\gamma} \ge 2eds, \text{ where } s \triangleq 36d^4k^5. \tag{16}$$

We then define two parameters p_{low} and p_{up} as follows:

$$p_{\text{low}} \triangleq \frac{1}{2} - \frac{1}{s}, \qquad p_{\text{up}} \triangleq \frac{1}{2} + \frac{1}{s}. \tag{17}$$

We will see later that $[p_{\text{low}}, p_{\text{up}}]$ is the interval in which the marginal probability on a single variable can locate during the process of the coupling.

Recall that $H_\Phi = (V, \mathcal{E})$ is the hypergraph for Φ defined in (2). The coupling procedure C_v is similar to the one used in [GLLZ19], which is an *adaptive* version of the coupling appeared in [Moi19].

The coupling procedure C_v is described in Algorithm 5, where we fix an arbitrary ordering of all clauses and all variables. The meanings of some variables appear in the algorithm are

- V_1 - a superset of all *discrepancy* variables. It contains all variables on which X^{C_v} and Y^{C_v} disagree. It may contain some additional variables to ease our analysis later.
- V_{set} - the variables whose values have been determined in the BFS process. X^{C_v} and Y^{C_v} can either agree or disagree on them.
- $\mathcal{S}$ - a subset of V_{set} on which X^{C_v} and Y^{C_v} agree. The coupling guarantees that $\mathcal{S} \cap \mathcal{M} = \varnothing$. Intuitively $\mathcal{S}$ together with $\mathcal{M}$ separates discrepancy variables from the rest.

The algorithm keeps growing the set V_1 in a BFS manner until there is no unassigned variable on the boundary of V_1. We remark that some of the choices in Algorithm 5 may seem confusing at first. They are because we need to later compare it with C to show (14). For example, we may choose $u \in \mathcal{M}_{v_0}$ in Line 4. Since we are coupling ν conditioned on v_0 being 0 and 1 respectively, any $u \in \mathcal{M}_{v_0}$ is guaranteed to be coupled successfully according to $X(u) = Y(u)$. However, we may still put u into V_1. This is a vacuous step that merely serves the purpose of comparing with C later, because we want to guarantee that under a suitable coupling, the set V_1 generated by C_v is the same as C.

Lemma 4.6. *The following properties hold for the coupling procedure C_v in Algorithm 5.*

Algorithm 5: The coupling procedure C_v

Input : a CNF formula Φ, a hypergraph $H_\Phi = (V, \mathcal{E})$, a set of marked variables $\mathcal{M}$, a variable $v_0 \in \mathcal{M}$, the distribution ν in (15), the parameters $p_{\text{low}}, p_{\text{up}}$ in (17), a parameter $k_\gamma > 0$ such that $k_\gamma < k_\beta$;

Output : a pair of assignments $X^{C_v}, Y^{C_v} \in \{0,1\}^V$.

1 $X^{C_v}(v_0) = 0$ and $Y^{C_v}(v_0) = 1$;
2 $V_1 \leftarrow \{v_0\}$, $V_2 \leftarrow V \setminus V_1$, $V_{\text{set}} \leftarrow \{v_0\}$ and $\mathcal{S} \leftarrow \emptyset$;
3 **while** $\exists e \in \mathcal{E}$ *s.t.* $e \cap V_1 \neq \emptyset, (e \cap V_2) \setminus V_{\text{set}} \neq \emptyset$ **do**
4 let e be the first such hyperedge and u be the first variable in $(e \cap V_2) \setminus V_{\text{set}}$;
5 sample a real number $r_u \in [0,1]$ uniformly at random;
6 let $p_u^X = \nu_u(0 \mid X^{C_v})$ and $p_u^Y = \nu_u(0 \mid Y^{C_v})$;
7 extend X^{C_v} to variable u s.t. $X^{C_v}(u) = 0$ if $r_u \le p_u^X$, o.w. $X^{C_v}(u) = 1$;
8 extend Y^{C_v} to variable u s.t. $Y^{C_v}(u) = 0$ if $r_u \le p_u^Y$, o.w. $Y^{C_v}(u) = 1$;
9 $V_{\text{set}} \leftarrow V_{\text{set}} \cup \{u\}$;
10 **if** $p_{\text{low}} < r_u \le p_{\text{up}}$ **then**
11 $V_1 \leftarrow V_1 \cup \{u\}$, $V_2 \leftarrow V \setminus V_1$;
12 **if** $(u \notin \mathcal{M}) \wedge (r_u \le p_{\text{low}} \vee r_u > p_{\text{up}})$ **then**
13 $\mathcal{S} \leftarrow \mathcal{S} \cup \{u\}$;
14 **for** $e \in \mathcal{E}$ *s.t. e is satisfied by both* $X^{C_v}(\mathcal{S})$ *and* $Y^{C_v}(\mathcal{S})$ **do**
15 $\mathcal{E} \leftarrow \mathcal{E} \setminus \{e\}$;
16 **for** $e \in \mathcal{E}$ *s.t.* $|e \cap (V_{\text{set}} \setminus \mathcal{M})| = k_\gamma$ **do**
17 $V_1 \leftarrow V_1 \cup (e \setminus V_{\text{set}})$, $V_2 \leftarrow V \setminus V_1$;
18 extend X^{C_v} and Y^{C_v} further on the set $V_2 \setminus V_{\text{set}}$ using the optimal coupling between $\nu_{V_2 \setminus V_{\text{set}}}(\cdot \mid X^{C_v}(V_{\text{set}}))$ and $\nu_{V_2 \setminus V_{\text{set}}}(\cdot \mid Y^{C_v}(V_{\text{set}}))$;
19 extend X^{C_v} and Y^{C_v} further on the set $V_1 \setminus V_{\text{set}}$ using the optimal coupling between $\nu_{V_1 \setminus V_{\text{set}}}(\cdot \mid X^{C_v}(V_{\text{set}} \cup V_2))$ and $\nu_{V_1 \setminus V_{\text{set}}}(\cdot \mid Y^{C_v}(V_{\text{set}} \cup V_2))$;
20 **return** (X^{C_v}, Y^{C_v});

- *The coupling procedure C_v terminates eventually and returns a pair $X^{C_v}, Y^{C_v} \in \{0,1\}^V$ such that X^{C_v} and Y^{C_v} have the law ν conditioned on $X^{C_v}(v_0) = 0$ and on $Y^{C_v}(v_0) = 1$, respectively.*
- *If $2^{k_\beta - k_\gamma} \ge 2eds$ where $s = 36d^4k^5$, then $X^{C_v}(V_2) = Y^{C_v}(V_2)$.*

We need the following lemma to prove Lemma 4.6.

Lemma 4.7. *In the coupling procedure C_v, if $2^{k_\beta - k_\gamma} \ge 2eds$ where $s = 36d^4k^5$, then for each p_u^X and p_u^Y computed Line 6, if $u \in \mathcal{M} \setminus \{v_0, v\}$, then $p_u^X = p_u^Y$; if $u \notin \mathcal{M} \setminus \{v\}$, then it holds that*

$$p_{\text{low}} \le p_u^X, p_u^Y \le p_{\text{up}}.$$

Proof. We prove the lemma by considering the two cases.

Case 1: $u \in \mathcal{M} \setminus \{v_0, v\}$. Due to the definition of ν (Definition 4.5), it must hold that $p_u^X = p_u^Y = 0$ or $p_u^X = p_u^Y = 1$, which implies $p_u^X = p_u^Y$.

Case 2: $u \notin \mathcal{M} \setminus \{v\}$. We prove the lemma for p_u^X. For p_u^Y it holds similarly. In each step, we have $X^{C_v} \in \{0,1\}^{V_{\text{set}}}$. Due to Definition 4.5, the distributions $\nu_u(\cdot \mid X^{C_v})$ is the distribution μ conditional on the values of variables in $\mathcal{M}_v \cup V_{\text{set}}$ are fixed. We use $\mathcal{E}_H$ to denote the set of all hyperedges in hypergraph H_Φ. We claim that for each execution of Line 6, the following property holds

$$\forall e \in \mathcal{E}_H : |e \cap (V_{\text{set}} \setminus \mathcal{M})| \le k_\gamma \vee \text{the clause represented by } e \text{ is satisfied by } X^{C_v}. \tag{18}$$

By Condition 3.1, each hyperedge contains at least k_β unmarked variables. By (18), for each hyperedge that is not satisfied by the current X^{C_v}, it contains at least $k_\beta - k_\gamma$ unmarked variables whose value are not fixed by the current X^{C_v}. By the definition of the distribution ν and Corollary 2.2, if $2^{k_\beta - k_\gamma} \ge 2eds$

where $s = 36d^4k^5$, then

$$p_u^X = \nu_u(0 \mid X^{C_v}) \le \frac{1}{2}\exp\left(\frac{1}{s}\right) \le \frac{1}{2}\left(1+\frac{2}{s}\right) \le \frac{1}{2}+\frac{1}{s},$$
$$1-p_u^X = \nu_u(1 \mid X^{C_v}) \le \frac{1}{2}\exp\left(\frac{1}{s}\right) \le \frac{1}{2}\left(1+\frac{2}{s}\right) \le \frac{1}{2}+\frac{1}{s}.$$

We now prove (18). Note that at the beginning of the coupling procedure C_v, the set $V_{\text{set}} = \{v_0\} \subseteq \mathcal{M}$, and thus for all hyperedges $e \in \mathcal{E}$, it holds that $|e \cap (V_{\text{set}} \setminus \mathcal{M})| = 0$. Hence, the property in (18) holds at the beginning.

Suppose in some execution of Line 6, there is a hyperedge e that violates the property in (18). Formally, the clause represented by e is not satisfied by X^{C_v} and $|e \cap (V_{\text{set}} \setminus \mathcal{M})| > k_\gamma$. Then we can find the first round of the while-loop after which the clause represented by e is not satisfied by X^{C_v} and $|e \cap (V_{\text{set}} \setminus \mathcal{M})| = k_\gamma$. Denote this round by R. In round R and any previous round of R, the clause represented by e cannot be satisfied by X^{C_v}. Hence e cannot be deleted in Line 15 up to round R. Since $|e \cap (V_{\text{set}} \setminus \mathcal{M})| = k_\gamma$, e satisfies the condition in Line 16. After Line 17, we have $e \subseteq V_1 \cup V_{\text{set}}$, which means that, after the round R, any vertex $u \in e$ cannot be pick in Line 4. Hence, it holds that $|e \cap (V_{\text{set}} \setminus \mathcal{M})| = k_\gamma$ after the round R, which contradicts to the assumption that $|e \cap (V_{\text{set}} \setminus \mathcal{M})| > k_\gamma$. □

Proof of Lemma 4.6. Firstly, we prove that the coupling procedure must terminate. This is because the size of the set V_{set} is increased by one in each while-loop.

Secondly, we prove that the final X^{C_v} follows the distribution ν conditional on $X^{C_v}(v_0) = X(v_0) = 0$. The same argument applies to Y^{C_v}. At the beginning, we set $X^{C_v}(v_0) = 0$. Note that, in each step, it holds that $X^{C_v} \in \{0,1\}^{V_{\text{set}}}$ and the algorithm always extends X^{C_v} according to the distribution ν conditional on the current assignment on V_{set}. By the chain rule, it is easy to verify the final X^{C_v} follows the distribution ν conditional on $X^{C_v}(v_0) = X(v_0) = 1$.

Finally, consider the final sets $V_1, V_2, \mathcal{S}, V_{\text{set}}$ and the final assignments X^{C_v} and Y^{C_v}. We prove the following two properties.

(i) The two distributions $\nu_{V_2 \setminus V_{\text{set}}}(\cdot \mid X^{C_v}(V_{\text{set}}))$ and $\nu_{V_2 \setminus V_{\text{set}}}(\cdot \mid X^{C_v}(V_{\text{set}} \cap V_2))$ are identical; and the two distributions $\nu_{V_2 \setminus V_{\text{set}}}(\cdot \mid Y^{C_v}(V_{\text{set}}))$ and $\nu_{V_2 \setminus V_{\text{set}}}(\cdot \mid Y^{C_v}(V_{\text{set}} \cap V_2))$ are identical.

(ii) $X^{C_v}(V_{\text{set}} \cap V_2) = Y^{C_v}(V_{\text{set}} \cap V_2)$.

If the above two properties (i) and (ii) hold, then $\nu_{V_2 \setminus V_{\text{set}}}(\cdot \mid X^{C_v}(V_{\text{set}}))$ and $\nu_{V_2 \setminus V_{\text{set}}}(\cdot \mid Y^{C_v}(V_{\text{set}}))$ can be perfectly coupled, which implies $X^{C_v}(V_2 \setminus V_{\text{set}}) = Y^{C_v}(V_2 \setminus V_{\text{set}})$. Combining with Property (ii), it proves that $X^{C_v}(V_2) = Y^{C_v}(V_2)$.

We now prove Property (i). We show that two distributions $\nu_{V_2 \setminus V_{\text{set}}}(\cdot \mid X^{C_v}(V_{\text{set}}))$ and $\nu_{V_2 \setminus V_{\text{set}}}(\cdot \mid X^{C_v}(V_{\text{set}} \cap V_2))$ are identical. For Y^{C_v} it holds similarly. First observe that $\mathcal{S} \subseteq V_2$. This is because a variable u is added to V_1 either because the condition in Line 10 holds or because of Line 17. In the first case, the condition in Line 12 does not hold and u will never be added to $\mathcal{S}$. In the second case, $u \notin V_{\text{set}}$ and thus $u \notin \mathcal{S}$ as well. Once a variable u is added into V_1, u cannot be picked in Line 4, and thus u cannot be added in $\mathcal{S}$ for the rest of the coupling.

For any clause c in the original CNF formula Φ such that $\mathsf{vbl}(c) \cap V_1 \neq \emptyset$ and $\mathsf{vbl}(c) \cap V_2 \neq \emptyset$, we claim that one of the following properties must hold:

- The clause c is satisfied by the assignment $X^{C_v}(\mathcal{S})$;
- The clause c satisfies $\mathsf{vbl}(c) \cap V_2 \subseteq V_{\text{set}}$.

All clauses spanning both V_1 and $V_2 \setminus V_{\text{set}}$ are in the first case, and they are satisfied by $X^{C_v}(V_{\text{set}} \cap V_2)$ as $\mathcal{S} \subseteq V_{\text{set}} \cap V_2$. This implies Property (i).

We show the claim next. Suppose there exists a clause c with $\mathsf{vbl}(c) \cap V_1 \neq \emptyset$ and $\mathsf{vbl}(c) \cap V_2 \neq \emptyset$ such that c is not satisfied by $X^{C_v}(\mathcal{S})$ and $\mathsf{vbl}(c) \cap V_2 \nsubseteq V_{\text{set}}$. Let e denote the hyperedge that represents c in H_Φ. Since the coupling procedure terminates, the hyperedge e must be deleted in Line 15 during the coupling procedure C_v. Otherwise, e satisfies the condition in Line 3, and the coupling procedure cannot terminate. However, since c is not satisfied by $X^{C_v}(\mathcal{S})$ after the whole coupling procedure, c cannot be satisfied by $X^{C_v}(\mathcal{S})$ during the coupling procedure. This implies that e cannot be deleted in Lines 15.

We then prove Property (ii). Suppose $X^{C_v}(V_{\text{set}} \cap V_2) \neq Y^{C_v}(V_{\text{set}} \cap V_2)$. Let $u \in V_{\text{set}} \cap V_2$ be a variable such that $X^{C_v}(u) \neq Y^{C_v}(u)$. Since $u \in V_{\text{set}}$ and $u \neq v_0$, the coupling have computed p_u^X, p_u^Y in Line 6. Since $X^{C_v}(u) \neq Y^{C_v}(u)$, it must be that $p_u^X \neq p_u^Y$. By Lemma 4.7, we know that $p_{\text{low}} \leq p_u^X, p_u^Y \leq p_{\text{up}}$. By Lines 7 and 8, since $X^{C_v}(u) \neq Y^{C_v}(u)$,

$$p_{\text{low}} < r_u \leq p_{\text{up}},$$

where $r_u \in [0,1]$ is drawn in Line 5. In this case, the variable u must be added into V_1 in Line 11 and u stays in V_1 for the rest of the coupling. However, by assumption, $u \in V_2 = V \setminus V_1$. Contradiction. □

By Lemma 4.6, we know that the marginal distribution of $X^{C_v}(v)$ is identical to $\nu_v(\cdot \mid X^{C_v}(v_0) = 0)$. By Definition 4.5, we know that $X^{C_v}(v)$ follows the law $\mu_v(\cdot \mid X(\mathcal{M}_v))$. Similarly, we know that $Y^{C_v}(v)$ follows the law $\mu_v(\cdot \mid Y(\mathcal{M}_v))$. By Proposition 2.4, we have that $\forall v \in \mathcal{M} \setminus \{v_0\}$,

$$\begin{aligned} D_v &= d_{\text{TV}}\left(\mu_v(\cdot \mid X(\mathcal{M}_v)), \mu_v(\cdot \mid Y(\mathcal{M}_v))\right) \\ &\leq \mathbf{Pr}_{C_v}\left[X^{C_v}(v) \neq Y^{C_v}(v)\right] \\ &\leq \mathbf{Pr}_{C_v}\left[v \in V_1\right]. \end{aligned}$$

The last inequality holds because by Lemma 4.6, if $X^{C_v}(v) \neq Y^{C_v}(v)$, then $v \notin V_2$ and thus $v \in V_1$. Note that $D_{v_0} = 0$. The sum of all D_v can be bounded as follows

$$\sum_{v \in \mathcal{M}} D_v = \sum_{v \in \mathcal{M} \setminus \{v_0\}} D_v \leq \sum_{v \in \mathcal{M} \setminus \{v_0\}} \mathbf{Pr}_{C_v}\left[v \in V_1^{C_v}\right], \tag{19}$$

where we use $V_1^{C_v}$ to denote the set V_1 generated by the coupling procedure C_v.

4.3.2. *The coupling C.* To bound the sum of all $\mathbf{Pr}_{C_v}\left[v \in V_1^{C_v}\right]$, we introduce the coupling procedure C in Algorithm 6. The coupling C is basically the same as C_v except that *it treats all variables in $\mathcal{M}_{v_0}$ as free variables.* This difference is reflected in Line 6 of Algorithm 6, where we use conditional distribution of μ instead of ν in Line 6 of Algorithm 5. However, as p_{low} and p_{up} stay the same, we can construct a coupling of two couplings C and C_v such that the final set V_1 does not change. In this way, we obtain a uniform treatment for $\mathbf{Pr}_{C_v}\left[v \in V_1^{C_v}\right]$ for all v, which leads to a better bound comparing to analysing $\mathbf{Pr}_{C_v}\left[v \in V_1^{C_v}\right]$ individually.

To be more precise, we have the following lemma.

Lemma 4.8. *The following properties hold for the coupling procedure C in Algorithm 6.*

- *The coupling procedure C terminates eventually and returns a pair $X^C, Y^C \in \{0,1\}^{V_{\text{set}}}$ for a random set $V_{\text{set}} \subseteq V$ such that $v_0 \in V_{\text{set}}$.*
- *If $2^{k_\beta - k_\gamma} \geq 2eds$ where $s = 36d^4k^5$, then for any variable $v \in \mathcal{M} \setminus \{v_0\}$,*

$$\mathbf{Pr}_{C_v}\left[v \in V_1^{C_v}\right] = \mathbf{Pr}_C\left[v \in V_1^C\right],$$

where $V_1^{C_v}$ is the set V_1 generated by the coupling procedure C_v and V_1^C is the set V_1 generated by the coupling procedure C.

We need the following lemma, which is the analogue of Lemma 4.7. It follows from the same proof of the second case of Lemma 4.7.

Lemma 4.9. *In the coupling procedure C, if $2^{k_\beta - k_\gamma} \geq 2eds$ where $s = 36d^4k^5$, then for each p_u^X and p_u^Y computed Line 6, it holds that*

$$p_{\text{low}} \leq p_u^X, p_u^Y \leq p_{\text{up}}.$$

Proof of Lemma 4.8. We first show that the coupling procedure must terminate. This is because the size of the set V_{set} is increased by one in each while-loop.

Fix a variable $v \in \mathcal{M} \setminus \{v_0\}$. Consider the coupling procedure C_v (Algorithm 5) and the coupling procedure C (Algorithm 6). We couple the two procedures by sampling the same random real number

Algorithm 6: The coupling procedure C

Input : a CNF formula Φ, a hypergraph $H_\Phi = (V, \mathcal{E})$, a set of marked variables $\mathcal{M}$, a variable $v_0 \in \mathcal{M}$, the parameters $p_{\text{low}}, p_{\text{up}}$ in (17), a parameter $k_\gamma > 0$ such that $k_\gamma < k_\beta$;

Output : a pair of assignments $X^C, Y^C \in \{0,1\}^{V_{\text{set}}}$ for some random set $V_{\text{set}} \subseteq V$.

1 $X^C(v_0) \leftarrow 0$ and $Y^C(v_0) \leftarrow 1$;
2 $V_1 \leftarrow \{v_0\}$, $V_2 \leftarrow V \setminus V_1$, $V_{\text{set}} \leftarrow \{v_0\}$ and $\mathcal{S} \leftarrow \varnothing$;
3 **while** $\exists e \in \mathcal{E}$ *s.t.* $e \cap V_1 \neq \varnothing, (e \cap V_2) \setminus V_{\text{set}} \neq \varnothing$ **do**
4 let e be the first such hyperedge and u be the first variable in $(e \cap V_2) \setminus V_{\text{set}}$;
5 sample a random real number $r_u \in [0,1]$ uniformly at random;
6 let $p_u^X = \mu_u(0 \mid X^C)$ and $p_u^Y = \mu_u(0 \mid Y^C)$;
7 extend X^C further on variable u s.t. $X^C(u) = 0$ if $r_u \le p_u^X$, o.w. $X^C(u) = 1$;
8 extend Y^C further on variable u s.t. $Y^C(u) = 0$ if $r_u \le p_u^Y$, o.w. $Y^C(u) = 1$;
9 $V_{\text{set}} \leftarrow V_{\text{set}} \cup \{u\}$;
10 **if** $p_{\text{low}} < r_u \le p_{\text{up}}$ **then**
11 $V_1 \leftarrow V_1 \cup \{u\}$, $V_2 \leftarrow V \setminus V_1$;
12 **if** $(u \notin \mathcal{M}) \wedge (r_u \le p_{\text{low}} \vee r_u > p_{\text{up}})$ **then**
13 $\mathcal{S} \leftarrow \mathcal{S} \cup \{u\}$;
14 **for** $e \in \mathcal{E}$ *s.t.* e *is satisfied by both* $X^C(\mathcal{S})$ *and* $Y^C(\mathcal{S})$ **do**
15 $\mathcal{E} \leftarrow \mathcal{E} \setminus \{e\}$;
16 **for** $e \in \mathcal{E}$ *s.t.* $|e \cap (V_{\text{set}} \setminus \mathcal{M})| = k_\gamma$ **do**
17 $V_1 \leftarrow V_1 \cup (e \setminus V_{\text{set}})$; $V_2 \leftarrow V \setminus V_1$;
18 **return** (X^C, Y^C);

$r_u \in [0,1]$ for each variable u. We claim that the following invariant holds for the two coupling procedures:

$$
\begin{aligned}
&V_1^{C_v} = V_1^C, \quad V_2^{C_v} = V_2^C, \quad V_{\text{set}}^{C_v} = V_{\text{set}}^C, \quad \mathcal{E}^{C_v} = \mathcal{E}^C, \\
&\mathcal{S}^{C_v} = \mathcal{S}^C, \quad X^{C_v}(\mathcal{S}^{C_v}) = X^C(\mathcal{S}^C), \quad Y^{C_v}(\mathcal{S}^{C_v}) = Y^C(\mathcal{S}^C)
\end{aligned} \tag{20}
$$

This implies that $V_1^{C_v} = V_1^C$ in the end, which is the second item of the lemma. We show (20) by induction.

Initially, it holds that $V_1^{C_v} = V_1^C = \{v_0\}$, $V_2^{C_v} = V_2^C = V \setminus \{v_0\}$, $V_{\text{set}}^{C_v} = V_{\text{set}}^C = \{v_0\}$ and $\mathcal{S}^{C_v} = \mathcal{S}^C = \varnothing$.

For each step of the while-loop, suppose (20) holds, then two coupling procedure pick the same hyperedge e and the same vertex $u \in e$. The two coupling procedures sample the same random number r_u and use the same parameters p_{low} and p_{up} in (17). Hence, after the Line 13 of either coupling, $V_1^{C_v} = V_1^C$, $V_2^{C_v} = V_2^C$, $V_{\text{set}}^{C_v} = V_{\text{set}}^C$, $\mathcal{E}^{C_v} = \mathcal{E}^C$, and $\mathcal{S}^{C_v} = \mathcal{S}^C$. Note that if the variable u is added into $\mathcal{S}$ in Line 13, then it must be that $(u \notin \mathcal{M}) \wedge (r_u \le p_{\text{low}} \vee r_u > p_{\text{up}})$. If $r_u \le p_{\text{low}}$, then by Lemma 4.7 and Lemma 4.9, in both coupling procedures $r_u \le p_u^X$ and $r_u \le p_u^Y$, which implies

$$X^{C_v}(u) = X^C(u) = Y^{C_v}(u) = Y^C(u) = 0.$$

Similarly, if $r_u > p_{\text{up}}$, then

$$X^{C_v}(u) = X^C(u) = Y^{C_v}(u) = Y^C(u) = 1.$$

Hence, the invariant in (20) holds after the Line 13. It is easy to verify that after the rest of the while-loop, the invariants in (20) still hold. □

By Lemma 4.8 and inequality (19), we have

$$\begin{aligned}\sum_{v\in\mathcal{M}} D_v &\le \sum_{v\in\mathcal{M}\setminus\{v_0\}} \mathbf{Pr}_{C_v}\left[v \in V_1^{C_v}\right]\\ &\le \sum_{v\in V\setminus\{v_0\}} \mathbf{Pr}_{C}\left[v \in V_1^{C}\right]\\ &= \mathbf{E}_C\left[|V_1^C|\right] - \mathbf{Pr}_C\left[v_0 \in V_1^C\right]\\ &= \mathbf{E}_C\left[|V_1^C|\right] - 1,\end{aligned}$$

where the last equation holds because v_0 must be in the set V_1^C. Our next step is to bound $\mathbf{E}_C\left[|V_1^C|\right]$.

4.3.3. *The proof of Lemma 4.4.* Finally, we finish the proof of Lemma 4.4 by proving the following lemma.

Lemma 4.10. *In the coupling procedure C (Algorithm 6), if $2^{k_\gamma} \ge 36d^4k^4$ and $2^{k_\beta - k_\gamma} \ge 2eds$ where $s = 36d^4k^5$, it holds that*

$$\mathbf{E}_C\left[|V_1|\right] \le \frac{3}{2}.$$

In Lemma 4.10, we can take

$$k_\gamma = \left\lceil \frac{4}{9} k_\beta \right\rceil.$$

Then, the following condition is sufficient to imply the condition of Lemma 4.10:

$$2^{k_\beta} \ge (36)^{\frac{9}{4}} d^9 k^9, \quad 2^{k_\beta} \ge (144\mathrm{e})^{\frac{9}{5}} d^9 k^9. \tag{21}$$

Note that $2^{k_\beta} \ge 2^{16} d^9 k^9$ is a sufficient condition for (21).

Consider the coupling procedure C defined in Algorithm 6. Upon termination, the coupling procedure generates assignments X^C and Y^C, and the sets of variables $V_1, V_2, V_{\mathrm{set}}, \mathcal{S} \subseteq V$. We define the failed hyperedge as follows.

Definition 4.11 (failed hyperedge)**.** We say a hyperedge $e \in \mathcal{E}$ is failed if one of the following events occurs after the coupling procedure C:

(i) there exists $v \in (e \cap V_{\mathrm{set}}) \setminus \{v_0\}$ such that $p_{\mathrm{low}} < r_v \le p_{\mathrm{up}}$;
(ii) $|e \cap (V_{\mathrm{set}} \setminus \mathcal{M})| = k_\gamma$ and e is not satisfied by both $X(\mathcal{S})$ and $Y(\mathcal{S})$.

In the following, we will use Reason (i) and Reason (ii) to denote the above two reasons of failure.

Definition 4.12 (line graph)**.** Let $H = (V, \mathcal{E})$ be a hypergraph. The line graph $\mathrm{Lin}(H) = (V_L, E_L)$ has hyperedges in $\mathcal{E}$ as its vertices and two hyperedges are adjacent if they intersect, i.e. $V_L = \mathcal{E}$ and $\{e_1, e_2\} \in E_L$ iff $e_1 \cap e_2 \neq \emptyset$.

Let $\mathrm{Lin}^2(H)$ denote the power graph of $\mathrm{Lin}(H)$. Two vertices in $\mathrm{Lin}^2(H)$ are adjacent if and only if their distance in $\mathrm{Lin}(H)$ is at most 2. For any vertex $v \in V$, we define the sets N_v, N_v^2 of hyperedges as

$$\begin{aligned}N_v &\triangleq \{e \in \mathcal{E} \mid v \in e\};\\ N_v^2 &\triangleq \{e \in \mathcal{E} \mid (v \in e) \vee (\exists e' \in \mathcal{E} \text{ s.t. } e \cap e' \neq \emptyset \wedge v \in e')\}.\end{aligned}$$

The set N_v is the set of all hyperedges that contains v. The set N_v^2 is the set of all hyperedges that either contains v or intersects with some hyperedges containing v. The following lemma asserts that for any $v \in V_1$, there are a path in $\mathrm{Lin}^2(H)$ that leads to v.

Lemma 4.13. *For any variable $v \in V \setminus \{v_0\}$, if $v \in V_1$, then there must exist a sequence of hyperedges $e_1, e_2, \ldots, e_\ell$ for some $\ell \ge 1$ such that the following properties hold:*

- *$e_1 \in N_{v_0}^2$ and $v \in e_\ell$;*
- *for all $1 \le i \le \ell$, the hyperedge e_i is failed;*
- *for all $1 \le i < \ell$, e_i and e_{i+1} are adjacent in $\mathrm{Lin}^2(H)$.*

Proof. We first show that each variable $u \in V_1 \setminus \{v_0\}$ must be incident to a failed hyperedge. For $u \in V_1 \setminus \{v_0\}$, u is either added into V_1 in Line 11 or in Line 17. Suppose u is added into V_1 in Line 11. In this case, the variable u is picked in Line 4 due to some hyperedge e. Then, it must be that $p_{\text{low}} \le r_u \le p_{\text{up}}$. This implies that u is incident to the failed hyperedge e (for Reason (i)). Next suppose u is added into V_1 in Line 17. In this case, $u \in e$ for some $e \in \mathcal{E}$ satisfying the condition in Line 16. Hence, the hyperedge e is failed for Reason (ii) and u is incident to e. If e satisfies the condition in Line 16, then $e \subseteq V_1 \cup V_{\text{set}}$ after Line 17. Hence, the condition in Reason (ii) holds for e for the rest of the coupling.

Thus we only need to show the following claim: for each failed $e \in \mathcal{E}$, there must exist a sequence of hyperedges $e_1, e_2, \ldots, e_\ell$ for some $\ell \ge 1$ such that the following properties hold:

- $e_1 \in N^2_{v_0}$ and $e = e_\ell$;
- for all $1 \le i \le \ell$, e_i is failed;
- for all $1 \le i < \ell$, e_i and e_{i+1} are adjacent in $\text{Lin}^2(H)$.

Consider the execution of the coupling procedure C. We say a hyperedge $e \in \mathcal{E}$ becomes failed once e satisfies one of the reasons in Definition 4.11. Note that once a hyperedge becomes failed, it will stay failed for the rest of the coupling. Moreover, the failed hyperedge must intersect the hyperedge satisfying the condition of the round of the while-loop in which it becomes failed. We list all failed hyperedges $e_{i_1}, e_{i_2}, \ldots, e_{i_r}$ such that e_{i_j} is the j-th hyperedge that becomes failed. Ties are broken arbitrarily. We prove the claim above by induction on the index j from 1 to r.

For the base case, we only need to show that $e_{i_1} \in N^2_{v_0}$. Notice that $v \ne v_0$ and $v \in V_1$. If some hyperedge containing v_0 is failed, then $e_{i_1} \in N_{v_0}$. Otherwise, the only possibility that $V_1 \ne \{v_0\}$ is that after setting a number of successfully coupled variables, there is a failed hyperedge satisfying Reason (ii). In the round when this happens, the current hyperedge chosen in Line 4 must contain v_0 (otherwise C terminates with $V_1 = \{v_0\}$). The first such hyperedge is e_{i_1} and thus $e_{i_1} \in N^2_{v_0}$.

Suppose the claim holds for $e_{i_1}, e_{i_2}, \ldots, e_{i_{k-1}}$. We show the claim for e_{i_k}. Consider the round of the while-loop when e_{i_k} becomes failed. In Line 4 of this round, the coupling procedure picks a hyperedge e and a variable $u \in e$ such that $e \cap V_1 \ne \emptyset$. As e_{i_k} went failed in this round, either $e_k = e$ (due to Reason (i)), or $e_{i_k} \in N_u$ (due to Reason (ii)). In both cases, $e \cap e_{i_k} \ne \emptyset$. If $v_0 \in e$, then $e_{i_k} \in N^2_{v_0}$ and the claim holds by letting $e_1 = e_{i_k}$. Otherwise, since e is picked in this round, there must exist a variable $u' \in V_1 \cap e$ and $u' \ne v_0$. Thus u' is incident to a failed hyperedge e_{i_j} for some $1 \le j \le k-1$. Since $u' \in e \cap e_{i_j}$ and $e \cap e_{i_k} \ne \emptyset$, e_{i_j} and e_{i_k} are adjacent in $\text{Lin}^2(H)$. By the induction hypothesis, there exists a failed hyperedge path in $\text{Lin}^2(H)$ that ends with e_{i_j}. This proves the claim for e_{i_k}. □

An *induced path* is a path that is also an induced subgraph. In particular, if we have an induced path $e_1, e_2, \ldots, e_\ell$, then for any $i < j$ such that $|i - j| \ge 2$, e_i and e_j are not adjacent. The following lemma follows from taking the shortest path among all paths guaranteed in Lemma 4.13.

Corollary 4.14. *For any variable $v \in V \setminus \{v_0\}$, if $v \in V_1$, then there must exist a sequence of hyperedges $e_1, e_2, \ldots, e_\ell$ for some $\ell \ge 1$ such that the following properties hold*

- *$e_1 \in N^2_{v_0}$ and $v \in e_\ell$;*
- *for all $1 \le i \le \ell$, e_i is failed;*
- *$e_1, e_2, \ldots, e_\ell$ is an induced path in $\text{Lin}^2(H)$.*

We are now ready to prove Lemma 4.10, namely

$$\mathbf{E}_C\left[|V_1|\right] \le \frac{3}{2}.$$

Fix any induced path (IP) $e_1, e_2, \ldots, e_\ell$ in $\text{Lin}^2(H)$. We bound the probability that all hyperedges in this path are failed hyperedges. Obliviously,

$$\mathbf{Pr}_C\left[\forall 1 \le i \le \ell, e_i \text{ is failed}\right] \le \mathbf{Pr}_C\left[\forall 1 \le j \le \lceil \ell/2 \rceil, e_{2j-1} \text{ is failed}\right]. \tag{22}$$

To bound the RHS of (22), we define the set of disjoint hyperedges

$$\mathcal{D} \triangleq \{e_{2j-1} \mid 1 \le j \le \lceil \ell/2 \rceil\}. \tag{23}$$

Because this is an induced path in $\text{Lin}^2(H)$, for any $e, e' \in \mathcal{D}$, it holds that $e \cap e' = \emptyset$. However, because of the subtlety of the adaptive coupling procedure C, we cannot claim that the events of e

being failed are independent from each other for $e \in \mathcal{D}$ based on this disjointness alone. Instead, we will implement the coupling procedure C in a slightly different way.

For each hyperedge $e \in \mathcal{D}$, we define two sequences of random numbers: $\mathcal{R}_{e,1}$ of length $k - k_\beta$ and $\mathcal{R}_{e,2}$ of length k_γ, where

- for each $1 \le i \le k - k_\beta$, $\mathcal{R}_{e,1}(i) \in [0,1]$ is a uniform and independent real number;
- for each $1 \le i \le k_\gamma$, $\mathcal{R}_{e,2}(i) \in [0,1]$ is a uniform and independent real number;

Suppose each hyperedge $e \in \mathcal{D}$ maintains two indices $i_{e,1}$ and $i_{e,2}$. Initially, $i_{e,1} = i_{e,2} = 1$. We run the coupling procedure C with the following modification. For each round of the while-loop in C, if the vertex u picked in Line 4 satisfies $u \in e$ for some $e \in \mathcal{D}$ (such e is unique because all hyperedges in $\mathcal{D}$ are disjoint), then we modify Line 5 as follows:

- if $u \in \mathcal{M}$, let $r_u = \mathcal{R}_{e,1}(i_{e,1})$, and let $i_{e,1} \leftarrow i_{e,1} + 1$;
- if $u \notin \mathcal{M}$, let $r_u = \mathcal{R}_{e,2}(i_{e,2})$ if the literal u appears in the clause represented by e; let $r_u = 1 - \mathcal{R}_{e,2}(i_{e,2})$ if the literal $\neg u$ appears in the clause represented by e, and let $i_{e,2} \leftarrow i_{e,2} + 1$.

Note that all numbers in $\mathcal{R}_{e,1}$ and $\mathcal{R}_{e,2}$ are uniformly distributed over $[0,1]$. In the modification above, each r_u is either r or $1 - r$ for some $r \in \mathcal{R}_{e,1} \cup \mathcal{R}_{e,2}$. Hence, each r_u is uniformly distributed over $[0,1]$. For any $e \in \mathcal{D}$, it contains at most $k - k_\beta$ marked variables, and there are at most k_γ unmarked variables $u \in e$ that need to sample r_u in C. Hence, the two sequences $\mathcal{R}_{e,1}$ and $\mathcal{R}_{e,2}$ will not exhaust during the coupling procedure C. As a result, the modification above will not affect the execution and the outcome of C.

For each $e \in \mathcal{D}$, we say the event $\mathcal{A}_e$ occurs if one of the following two events occurs:

- there exists a random number r in $\mathcal{R}_{e,1} \cup \mathcal{R}_{e,2}$ such that $p_{\text{low}} < r \le p_{\text{up}}$;
- for all $1 \le i \le k_\gamma$, $0 \le \mathcal{R}_{e,2}(i) \le p_{\text{up}}$.

Then, we have the following claim.

Claim 4.15. *For each hyperedge $e \in \mathcal{D}$, if e is a failed hyperedge after the coupling procedure C, then the event $\mathcal{A}_e$ must occur.*

Proof. Fix a hyperedge $e \in \mathcal{D}$. After the coupling procedure C, for all $u \in e \cap V_{\text{set}} \setminus \{v_0\}$, the random number r_u comes from $\mathcal{R}_{e,1} \cup \mathcal{R}_{e,2}$. Suppose e is a failed hyperedge after the coupling procedure C, by Definition 4.11, here are two cases.

Reason (i): there exists $v \in (e \cap V_{\text{set}}) \setminus \{v_0\}$ such that $p_{\text{low}} < r_v \le p_{\text{up}}$, then there must exist a random number r in $\mathcal{R}_{e,1} \cup \mathcal{R}_{e,2}$ such that $p_{\text{low}} < r \le p_{\text{up}}$;

Reason (ii): $|e \cap (V_{\text{set}} \setminus \mathcal{M})| = k_\gamma$ and e is not satisfied by both $X(\mathcal{S})$ and $Y(\mathcal{S})$. Let c_e denote the clause represented by e. We list all variables in $u_1, u_2, \ldots, u_{k_\gamma}$ in $|e \cap (V_{\text{set}} \setminus \mathcal{M})|$ such that u_i is the i-th variable processed by the while-loop in C. Fix $1 \le i \le k_\gamma$.

- Suppose the literal u_i appears in c_e. If $r_{u_i} > p_{\text{up}}$, then by Lemma 4.9 and Line 13, we have $X^C(u_i) = Y^C(u_i) = 1$ and $u_i \in \mathcal{S}$. In this case c_e is satisfied by $X(\mathcal{S})$ and $Y(\mathcal{S})$, and the event in Reason (ii) cannot occur. So we must have $r_{u_i} \le p_{\text{up}}$. Since $r_{u_i} = \mathcal{R}_{e,2}(i)$, we have $\mathcal{R}_{e,2}(i) \le p_{\text{up}}$.
- Suppose the literal $\neg u_i$ appears in c_e. If $r_{u_i} \le p_{\text{low}}$, then by Lemma 4.9 and Line 13, we have $X^C(u_i) = Y^C(u_i) = 0$ and $u_i \in \mathcal{S}$. In this case c_e is satisfied by $X(\mathcal{S})$ and $Y(\mathcal{S})$, and the event in Reason (ii) cannot occur. So we must have $r_{u_i} > p_{\text{low}}$. Since $r_{u_i} = 1 - \mathcal{R}_{e,2}(i)$, we have $\mathcal{R}_{e,2}(i) < 1 - p_{\text{low}} = p_{\text{up}}$.

This implies for all $1 \le i \le k_\gamma$, $0 \le \mathcal{R}_{e,2}(i) \le p_{\text{up}}$. □

For each $e \in \mathcal{D}$, all reals numbers in $\mathcal{R}_{e,1}$ and $\mathcal{R}_{e,2}$ are sampled uniformly and independently. We use $\mathcal{R}_e$ to denote this product distribution. And we use $\mathcal{R}$ to denote the product distribution of all $\mathcal{R}_e$

for $e \in \mathcal{D}$. By the definition of $\mathcal{D}$ in (23), we can bound the RHS of (22) as

$$\begin{aligned}
\mathbf{Pr}_C\left[\forall 1 \le i \le \ell, e_i \text{ is failed}\right] &\le \mathbf{Pr}_C\left[\forall e \in \mathcal{D}, e \text{ is failed}\right] \\
\text{(by Claim 4.15)} \qquad &\le \mathbf{Pr}_{\mathcal{R}}\left[\bigwedge_{e\in\mathcal{D}} \mathcal{A}_e\right] \\
&= \prod_{e\in\mathcal{D}} \mathbf{Pr}_{\mathcal{R}_e}\left[\mathcal{A}_e\right] \\
\text{(by the definition of } \mathcal{A}_e) \qquad &\le \prod_{e\in\mathcal{D}} \left(\frac{2k}{s} + \left(\frac{1}{2}+\frac{1}{s}\right)^{k_Y}\right).
\end{aligned}$$

We define p_{failed} as

$$p_{\mathsf{failed}} \triangleq \frac{2k}{s} + \left(\frac{1}{2}+\frac{1}{s}\right)^{k_Y}. \tag{24}$$

Note that $|\mathcal{D}| \ge \ell/2$. Thus, for any induced path (IP) $e_1, e_2, \ldots, e_\ell$, we have

$$\mathbf{Pr}_C\left[\forall 1 \le i \le \ell, e_i \text{ is failed}\right] \le p_{\mathsf{failed}}^{\ell/2}.$$

By Corollary 4.14, we have for any vertex $v \ne v_0$,

$$\begin{aligned}
\mathbf{Pr}_C\left[v \in V_1\right] &\le \sum_{\substack{\text{IP } e_1,e_2,\ldots,e_\ell \text{ in } L^2 \\ \text{satisfying } e_1 \in N^2_{v_0}, v\in e_\ell}} \mathbf{Pr}_C\left[\forall 1 \le i \le \ell, e_i \text{ is failed}\right] \\
&\le \sum_{\substack{\text{IP } e_1,e_2,\ldots,e_\ell \text{ in } L^2 \\ \text{satisfying } e_1 \in N^2_{v_0}, v\in e_\ell}} p_{\mathsf{failed}}^{\ell/2}. \qquad (25)
\end{aligned}$$

Note that $v_0 \in V_1$, then we have

$$\begin{aligned}
\mathbf{E}_C\left[|V_1|\right] - 1 &= \sum_{v\in V\setminus\{v_0\}} \mathbf{Pr}_C\left[v \in V_1\right] \\
\text{(by (25))} \qquad &\le \sum_{v\in V\setminus\{v_0\}} \sum_{\substack{\text{IP } e_1,e_2,\ldots,e_\ell \text{ in } L^2 \\ \text{satisfying } e_1 \in N^2_{v_0}, v\in e_\ell}} p_{\mathsf{failed}}^{\ell/2} \\
&\le \sum_{\substack{\text{IP } e_1,e_2,\ldots,e_\ell \text{ in } L^2 \\ \text{satisfying } e_1 \in N^2_{v_0}}} k \cdot p_{\mathsf{failed}}^{\ell/2},
\end{aligned}$$

where in the last inequality, we enumerate all the IPs starting from $N^2_{v_0}$ and use the fact that each hyperedge contains k vertices. Note that the maximum degree of $\mathrm{Lin}^2(H)$ is at most d^2k^2 and there are at most d^2k hyperedges in set $N^2_{v_0}$. Thus, we have

$$\mathbf{E}_C\left[|V_1|\right] - 1 \le \sum_{\ell=1}^{\infty} d^2k \cdot (d^2k^2)^{\ell-1} \cdot k \cdot p_{\mathsf{failed}}^{\ell/2} = \sum_{\ell=1}^{\infty} (d^2k^2)^{\ell} \cdot p_{\mathsf{failed}}^{\ell/2} = \sum_{\ell=1}^{\infty} c^{\ell} = \frac{c}{1-c},$$

where $c \triangleq d^2k^2\sqrt{p_{\mathsf{failed}}}$. Hence, to prove $\mathbf{E}\left[|V_1|\right] \le \frac{3}{2}$, it is sufficient to prove that

$$c = d^2k^2\sqrt{p_{\mathsf{failed}}} \le \frac{1}{3},$$

which, in turn, is implied by

$$p_{\mathsf{failed}} \le \frac{1}{9d^4k^4}.$$

Recall that p_{failed} is defined in (24) and $s = 36d^4k^5$. We have

$$\begin{aligned} p_{\mathsf{failed}} &= \frac{2k}{s} + \left(\frac{1}{2} + \frac{1}{s}\right)^{k_\gamma} \\ &\le \frac{1}{18d^4k^4} + \left(\frac{1}{2}\right)^{k_\gamma} \exp\left(\frac{2k_\gamma}{s}\right) \\ \text{(by } k_\gamma \le k\text{)} \qquad &\le \frac{1}{18d^4k^4} + \left(\frac{1}{2}\right)^{k_\gamma - 1}. \end{aligned}$$

Since $2^{k_\gamma} \ge 36d^4k^4$, we have that $p_{\mathsf{failed}} \le \frac{1}{9d^4k^4}$.

5. Analyze the rejection sampling subroutine

In this section, we will analyze the Sample subroutine (Algorithm 3).

Let $\Lambda \subseteq \mathcal{M}$ be a subset of marked variables, $\varepsilon > 0$, $X \in \{0,1\}^\Lambda$ and $S \subseteq V \setminus \Lambda$. We continue to use the same notations as in Section 3.3. Let $\Phi^X = (V^X, C^X)$ be the formula obtained from Φ simplified under X, and $\Phi^X = \Phi_1^X \wedge \Phi_2^X \wedge \cdots \wedge \Phi_\ell^X$ where all $\Phi_i^X = (V_i^X, C_i^X)$ are disjoint. For every $i \in [m]$, $H_i^X = (V_i^X, \mathcal{E}_i^X)$, the hypergraph representation of Φ_i^X, is connected. Assume without loss of generality that $V_i \cap S \neq \varnothing$ for $1 \le i \le m$ and $V_i \cap S = \varnothing$ for $m < i \le \ell$.

Lemma 5.1. *For any $0 < \eta < 1$, the time complexity of* $\mathsf{Sample}(\Phi, \delta, X, S)$ *is $O\left(|S| \left(\frac{n}{\delta}\right)^{\frac{\eta}{10}} d^2k^3 \log^2 \frac{n}{\delta}\right)$. Furthermore, if $2^{k_\beta} \ge \frac{20}{\eta} edk$ and $\left|\mathcal{E}_i^X\right| \le dk \log \frac{n}{\delta}$ for every $1 \le i \le m$, then* $\mathsf{Sample}(\Phi, \delta, X, S)$ *returns a random assignment $Y \in \{0,1\}^S$ satisfying $d_{\mathrm{TV}}(Y, \mu_S(\cdot \mid X)) \le \delta$.*

Proof. We first analyze the running time of $\mathsf{Sample}(\Phi, \delta, X, S)$. We need to find all the connected components $\left\{H_i^X = (V_i^X, \mathcal{E}_i^X) \mid 1 \le i \le m\right\}$ in H_{Φ^X} such that each $V_i^X \cap S \neq \varnothing$ and check whether there exists $1 \le i \le m$ such that $|\mathcal{E}_i^X| > dk \log \frac{n}{\delta}$. Suppose we store the hypergraph H_Φ as an adjacent list. For each vertex $v \in S$, we apply the deep first search starting from v in H_Φ. When visiting each hyperedge e, we can check whether e is in H_{Φ^X}. Once we find that one connected component in H_{Φ^X} contains more than $dk \log \frac{n}{\delta}$ hyperedges, we stop this process immediately. The time complexity of the deep first search step is at most

$$T_{\mathsf{DFS}} = O\left(|S| d^2k^3 \log \frac{n}{\delta}\right).$$

If $|\mathcal{E}_i^X| \le dk \log \frac{n}{\delta}$ for all $1 \le i \le m$, then we apply the rejection sampling for each Φ_i^X. Note that $m \le |S|$. The time complexity of the rejection sampling step is at most

$$T_{\mathsf{RS}} = O\left(|S| Rdk^2 \log \frac{n}{\delta}\right) = O\left(|S| \left(\frac{n}{\delta}\right)^{\frac{\eta}{10}} dk^2 \log^2 \frac{n}{\delta}\right).$$

The overall time complexity for the subroutine $\mathsf{Sample}(\Phi, \delta, X, S)$ is at most

$$T_{\mathsf{S}} = T_{\mathsf{DFS}} + T_{\mathsf{RS}} = O\left(|S| \left(\frac{n}{\delta}\right)^{\frac{\eta}{10}} d^2k^3 \log^2 \frac{n}{\delta}\right).$$

We next analyze the total variation distance between Y and $\mu_S(\cdot \mid X)$. Since $\left|\mathcal{E}_i^X\right| \le dk \log \frac{n}{\delta}$ for every $1 \le i \le m$, the random assignment Y is returned in either Line 9 or Line 10. It follows from Proposition 3.4 that we only need to show the probability that Y is returned in Line 9 is at most δ, which is equivalent to that one of the $\mathsf{RejectionSampling}(\Phi_i^X, R)$ returns $\perp$ among all $1 \le i \le m$.

Fix $1 \le i \le m$. Consider the rejection sampling for the instance Φ_i^X. Let $\mathbf{Pr}_{\mathcal{P}}[\cdot]$ be the product distribution such that each variable in C_i^X takes a value from $\{0,1\}$ uniformly and independently. For each clause $c \in C_i^X$, let B_c denote the event that c is not satisfied. Define

$$\Gamma(B_c) = \{B_b \mid b \in C_i^X \wedge b \neq c \wedge \mathsf{vbl}(c) \cap \mathsf{vbl}(b) \neq \varnothing\}.$$

Suppose $2^{k_\beta} \geq \frac{20}{\eta} edk$ for some $0 < \eta < 1$. For each $c \in C_i'$, let $x(B_c) \triangleq \frac{\eta}{20dk}$. Since every clause has at least k_β unmarked vertices, we have that

$$\mathbf{Pr}_{\mathcal{P}}[B_c] \leq \left(\frac{1}{2}\right)^{k_\beta} \leq x(B_c) \prod_{B \in \Gamma(B_c)} (1 - x(B)).$$

By the Lovász local lemma in Theorem 2.1, we have

$$\mathbf{Pr}_{\mathcal{P}}\left[\bigwedge_{c \in C_i^X} \overline{B_c}\right] \geq \prod_{c \in C_i^X} (1 - x(B_c)) = \prod_{c \in C_i^X} \left(1 - \frac{\eta}{20dk}\right).$$

Since $\left|\mathcal{E}_i^X\right| \leq dk \log \frac{n}{\delta}$, we have

$$\mathbf{Pr}_{\mathcal{P}}\left[\bigwedge_{c \in C_i^X} \overline{B_c}\right] \geq \left(1 - \frac{\eta}{20dk}\right)^{dk \log \frac{n}{\delta}} \geq \left(1 - \frac{1}{\frac{15}{\eta} dk + 1}\right)^{dk \log \frac{n}{\delta}} \geq \exp\left(-\frac{\eta}{15} \log \frac{n}{\delta}\right) > \left(\frac{\delta}{n}\right)^{\frac{\eta}{10}}.$$

For each Φ_i^X, our algorithm repeats the rejection sampling for $\left\lceil \left(\frac{n}{\delta}\right)^{\frac{\eta}{10}} \log \frac{n}{\delta} \right\rceil$ times. Hence, the probability that the rejection for Φ_i^X fails is at most

$$\left(1 - \left(\frac{\delta}{n}\right)^{\frac{\eta}{10}}\right)^{\left\lceil \left(\frac{n}{\delta}\right)^{\frac{\eta}{10}} \log \frac{n}{\delta} \right\rceil} \leq \frac{\delta}{n}.$$

Note that m is at most n. Taking a union bound over all Φ_i^X for $1 \leq i \leq m$, we have that if the conditions of the lemma holds, then

$$\mathbf{Pr}\left[\exists i \in [m], \mathsf{RejectionSampling}\left(\Phi_i^X, R\right) = \perp\right] \leq \delta. \qquad \square$$

We now proceed to show that, in all calls to $\mathsf{Sample}(\Phi, \delta, X, S)$ during the execution of Algorithm 2, $\left|\mathcal{E}_i^X\right| \leq dk \log \frac{n}{\delta}$ for every $i \in [\ell]$ with high probability.

Algorithm 2 calls the subroutine Sample for $T + 1$ times (T times in Line 4 and once in Line 6). For each $1 \leq t \leq T + 1$, we use the $\mathcal{B}_t$ to denote the event that $\left|\mathcal{E}_i^X\right| > dk \log \frac{n}{\delta}$ for some $1 \leq i \leq \ell$ at the t-th call to $\mathsf{Sample}(\cdot)$. Note that, in all calls to $\mathsf{Sample}(\Phi, \delta, X, S)$ during the execution of Algorithm 2, the parameter δ is always set to $\frac{\varepsilon}{4(T+1)}$. The following lemma bounds the probability of each $\mathcal{B}_t$.

Lemma 5.2. *Assume $2^{k_\alpha} \geq 4e^2 d^2 k^2$ and $2^{k_\beta} \geq 2edk$. For each $1 \leq t \leq T + 1$, it holds that in the execution of Algorithm 2, $\mathbf{Pr}[\mathcal{B}_t] \leq \delta$, where $\delta = \frac{\varepsilon}{4(T+1)}$ and $T = \left\lceil 2n \log \frac{4n}{\varepsilon} \right\rceil$.*

The rest of this section is devoted to the proof of Lemma 5.2.

Recall that $(X_t)_{t=0}^T$ is the random process defined by Algorithm 2. Fix $1 \leq t \leq T + 1$. Consider the t-th call of the subroutine $\mathsf{Sample}(\Phi, \delta, X, S)$ (Algorithm 3). If $1 \leq t \leq T$, let $v \in \mathcal{M}$ denote the random vertex picked in the t-th step. The random assignment X and the subset S in the subroutine $\mathsf{Sample}(\Phi, \delta, X, S)$ are defined as

$$X = \begin{cases} X_{t-1}(\mathcal{M} \setminus \{v\}) \ (\text{namely } \Lambda = \mathcal{M} \setminus \{v\}) & \text{if } 1 \leq t \leq T, \\ X_T \ (\text{namely } \Lambda = \mathcal{M}) & \text{if } t = T + 1, \end{cases} \tag{26}$$

$$S = \begin{cases} \{v\} & \text{if } 1 \leq t \leq T, \\ V \setminus \mathcal{M} & \text{if } t = T + 1. \end{cases} \tag{27}$$

Consider the hypergraph $H_\Phi = (V, \mathcal{E})$ as defined in (2). Given an assignment $X \in \{0,1\}^{\mathcal{M}}$, we say a hyperedge $e \in \mathcal{E}$ in H_Φ is *bad* if the clause represented by e is not satisfied by X. Recall that we use Φ^X to denote the CNF formula obtained from Φ simplified under X and use $H_{\Phi^X} = (V, \mathcal{E}^X)$ to denote its hypergraph representation. Hence $\mathcal{E}^X \subseteq \mathcal{E}$ is the set of all bad hyperedges. If the bad event $\mathcal{B}_t$ occurs, there must exist a connected component in H_{Φ^X} containing more than $dk \log \frac{n}{\delta}$ bad hyperedges.

Fix a hyperedge $e \in \mathcal{E}$, let $\mathcal{B}_e$ be the event that

- the hyperedge e is in $\mathcal{E}^X$;
- $|\mathcal{E}_e| \geq dk \log \frac{n}{\delta}$, where $H_e = (V_e, \mathcal{E}_e)$ is the connected component in H_{Φ^X} such that $e \in \mathcal{E}_e$.

By the definition of $\mathcal{B}_e$, if the event $\mathcal{B}_t$ occurs, then there must exist $e \in \mathcal{E}$ such that the event $\mathcal{B}_e$ occurs. We have

$$\mathbf{Pr}\left[\mathcal{B}_t\right] \leq \mathbf{Pr}\left[\exists\, e \in \mathcal{E} \text{ s.t. } \mathcal{B}_e\right] \leq \sum_{e \in \mathcal{E}} \mathbf{Pr}\left[\mathcal{B}_e\right]. \tag{28}$$

Next we bound the probability of $\mathcal{B}_e$. We first establish local uniformity of any intermediate assignment X_t.

Lemma 5.3. *Suppose the CNF formula Φ satisfies $2^{k\beta} \geq 2eds$ for some $s \geq k$. Let $X \subseteq \{0,1\}^\Lambda$ be the random assignment defined in (26), where $\Lambda = \mathcal{M}$ or $\mathcal{M} \setminus \{v\}$ for some v. For any subset $S \subseteq \Lambda$ and any assignment $\sigma \in \{0,1\}^S$, it holds that*

$$\mathbf{Pr}\left[X(S) = \sigma\right] \leq \left(\frac{1}{2}\right)^{|S|} \exp\left(\frac{|S|}{s}\right).$$

Proof. By the definition of the X in (26), we know that $X = X_t(\Lambda)$ for some $0 \leq t \leq T$. For each vertex $v \in S$, we define $t_v \leq t$ as follows. If v is chosen by the Algorithm 2 at least once, then let t_v be the largest $t' \leq t$ such that v is chosen at the t'-th step. Otherwise, let $t_v = 0$.

We sort all the vertices in S according to t_v. If two vertices $u, v \in S$ satisfy $t_u = t_v = 0$, we break the tie arbitrarily. Let $v_1, v_2, \ldots, v_\ell$ be the set of all vertices in S such that

$$0 \leq t_{v_1} \leq t_{v_1} \leq \ldots \leq t_{v_\ell} \leq T.$$

Thus, we have

$$\forall 1 \leq i \leq \ell: \quad X(v_i) = X_{t_{v_i}}(v_i).$$

Consider the t_{v_i}-th step. The value $X_{t'}(v_i)$ is generated by $\mathsf{Sample}(\Phi, \frac{\varepsilon}{4(T+1)}, X_{t'-1}(\mathcal{M} \setminus \{v_i\}), \{v_i\})$, where $t' = t_{v_i}$. Suppose $2^{k\beta} \geq 2eds$ for some $s \geq k$. We claim that for any $v \in \mathcal{M}$, any $X' \in \{0,1\}^{\mathcal{M}\setminus\{v\}}$ and any $0 < \delta < 1$, it holds that

$$\forall c \in \{0,1\}, \quad \mathbf{Pr}\left[\mathsf{Sample}\left(\Phi, \delta, X', \{v\}\right) \text{ returns } c\right] \leq \frac{1}{2}\exp\left(\frac{1}{s}\right). \tag{29}$$

Assume inequality (29) holds. Note that $|S| = \ell$. By the chain rule, we have

$$\begin{aligned}
\mathbf{Pr}\left[X(S) = \sigma\right] &= \prod_{i=1}^{\ell} \mathbf{Pr}\left[X(v_i) = \sigma(v_i) \mid \forall 1 \leq j < i, X(v_j) = \sigma(v_j)\right] \\
&= \prod_{i=1}^{\ell} \mathbf{Pr}\left[X_{t_{v_i}}(v_i) = \sigma(v_i) \mid \forall 1 \leq j < i, X_{t_{v_j}}(v_j) = \sigma(v_j)\right] \\
&\leq \left(\frac{1}{2}\right)^{|S|} \exp\left(\frac{|S|}{s}\right),
\end{aligned}$$

where the last inequality holds due to (29) and the fact that the initial random assignment X_0 is sampled from $\{0,1\}^{\mathcal{M}}$ uniformly at random.

We now prove the inequality (29). By Algorithm 3 and Proposition 3.4, we know that the random value c returned by the subroutine $\mathsf{Sample}(\Phi, \delta, X', \{v\})$ is either sampled from $\{0,1\}$ uniformly at random or sampled independently from the distribution $\mu_v(\cdot \mid X')$. If c is sampled from $\{0,1\}$ uniformly at random, then (29) holds trivially. We now prove that

$$\forall c \in \{0,1\}, \quad \mu_v(c \mid X') \leq \frac{1}{2}\exp\left(\frac{1}{s}\right). \tag{30}$$

Recall $X' \in \{0,1\}^{\mathcal{M}\setminus\{v\}}$. Let $\Phi' \triangleq \Phi^{X'}$ be the CNF formula obtained from Φ by deleting all the clauses satisfied by X' and all the variables in $\mathcal{M}\setminus\{v\}$, and $\mu' \triangleq \mu^{X'}$ be the uniform distribution of all solutions in Φ'. Then the two distributions $\mu'_v(\cdot)$ and $\mu_v(\cdot \mid X')$ are identical. By Condition 3.1, we have each

clause in Φ' contains at least k_β variables and at most k variables. Each variable belongs to at most d clauses. Since $2^{k_\beta} \geq 2eds$ for some $s \geq k$, inequality (30) follows from Corollary 2.2. □

To bound the size of connected components including a particular hyperedge e, recall that $\mathrm{Lin}(H)$ is the line graph of H defined in Definition 4.12. We also need the notion of 2-trees.

Definition 5.4 (2-tree)**.** Let $G = (V, E)$ be a graph. A set of vertices $T \subseteq V$ is called a 2-tree if (1) for any $u, v \in T$, $\mathrm{dist}_G(u, v) \geq 2$; (2) if one adds an edge between every $u, v \in T$ such that $\mathrm{dist}_G(u, v) = 2$, then T is connected.

The following simple observation follows directly from the definition of 2-trees.

Observation 5.5. *If a graph $G = (V, E)$ has a 2-tree of size $\ell > 1$ containing the vertex $v \in V$, then G must have a 2-tree of size $\ell - 1$ containing the vertex v.*

Proof. Let $T \subseteq V$ be a 2-tree in G. Let $G' = (T, E_T)$, where each $\{u, v\} \in E_T$ if and only if $u, v \in T$ and $\mathrm{dist}_G(u, v) = 2$. Then G' is a connected graph. We can find an arbitrary spanning tree $T_{G'}$ of graph G'. Since the number of vertices in $T_{G'}$ is $\ell > 1$, then $T_{G'}$ contains at least two leaf vertices. Let w be the leaf vertex in $T_{G'}$ such that $w \neq v$. It is easy to see $T \setminus \{w\}$ is a 2-tree of size $\ell - 1$ containing the vertex v. □

To bound the number of 2-trees, we need the following lemma in [BCKL13] to bound the number of connected subgraphs.

Lemma 5.6. *Let $G = (V, E)$ be a graph with maximum degree Δ and $v \in V$ be a vertex. Then the number of connected induced subgraphs of size ℓ containing v is at most $\frac{(\mathrm{e}\Delta)^{\ell-1}}{2}$.*

Corollary 5.7. *Let $G = (V, E)$ be a graph with maximum degree Δ and $v \in V$ be a vertex. Then the number of 2-trees in G of size ℓ containing v is at most $\frac{(\mathrm{e}\Delta^2)^{\ell-1}}{2}$.*

Proof. Consider the power graph G^2. The maximum degree of G^2 is at most Δ^2. The number of connected induced subgraphs in G^2 of size ℓ containing vertex v is at most $\frac{(e\Delta^2)^{\ell-1}}{2}$. This is an upper bound of the number of 2-trees in G of size ℓ containing v. □

Lemma 5.8. *Let $H = (V, \mathcal{E})$ be a k-uniform hypergraph such that each vertex belongs to at most d hyperedges. Let $B \subseteq \mathcal{E}$ be a subset of hyperedges which induces a connected subgraph in $\mathrm{Lin}(H)$, and $e \in B$ be an arbitrary hyperedge. Then, there must exist a 2-tree $T \subseteq B$ in the graph $\mathrm{Lin}(H)$ such that $e \in T$ and $|T| = \left\lfloor \frac{|B|}{kd} \right\rfloor$.*

Proof. Consider the graph $\mathrm{Lin}(H) = (V_L, E_L)$. For any subset of vertices S in $\mathrm{Lin}(H)$, let the extended neighbourhood of S be

$$\Gamma^+(S) \triangleq \{v \in V_L \mid v \in S \text{ or there exists } u \in S \text{ s.t. } \{u, v\} \in E_L\}.$$

We construct a 2-tree greedily. Let $T_0 = \{e\}$. For the i-th step, we set $S \leftarrow B \setminus \Gamma^+(T_{i-1})$, let e_i be the first hyperedge in S such that $\mathrm{dist}_{\mathrm{Lin}(H)}(T_{i-1}, e_i) = 2$, and set $T_i = T_{i-1} \cup \{e_i\}$. The process ends when $B = \Gamma^+(T_j)$ for some j.

We claim that the set S will become empty eventually. Suppose the current 2-tree is T, and some non-empty $S = B \setminus \Gamma^+(T)$ remains. Thus, $\forall e' \in S$, $\mathrm{dist}_{\mathrm{Lin}(H)}(T, e') \neq 2$. Note that if $\mathrm{dist}_{\mathrm{Lin}(H)}(T, e') \leq 1$, $e' \in \Gamma^+(T)$. Thus, $\forall e' \in S$, $\mathrm{dist}_{\mathrm{Lin}(H)}(T, e') \geq 3$. Note that $B \subseteq \Gamma^+(T) \cup S$, $B \cap \Gamma^+(T) \neq \emptyset$ and $B \cap S \neq \emptyset$. Hence B is disconnected in $\mathrm{Lin}(H)$. Contradiction.

In every step, at most kd hyperedges are removed, so we have $|T| \geq \left\lfloor \frac{|B|}{kd} \right\rfloor$. Then by Observation 5.5, there must exist a 2-tree $T \subseteq B$ in graph $\mathrm{Lin}(H)$ such that $e \in T$ and $|T| = \left\lfloor \frac{|B|}{kd} \right\rfloor$. □

We are now ready to prove Lemma 5.2.

Proof of Lemma 5.2. We bound the probability $\mathcal{B}_e$ in (28). If the event $\mathcal{B}_e$ occurs, there must exist a subset set $B \subseteq \mathcal{E}$ such that $e \in B$, $|B| = L \triangleq \left\lceil dk \log \frac{n}{\delta} \right\rceil$, B is connected in $\mathrm{Lin}(H)$, and all hyperedges in

B are bad hyperedges, i.e. all hyperedges in B are not satisfied by X. Let $\ell \triangleq \lfloor \frac{L}{kd} \rfloor$. By 5.8, there must exists a 2-tree in $T \subseteq B$ such that $e \in T$ and $|T| = \ell$.

By the definition of $X \in \{0,1\}^\Lambda$ in (26) and Condition 3.1, we have $|e \cap \Lambda| \geq k_\alpha - 1$ for all $e \in \mathcal{E}$. Note that all hyperedges in T are disjoint. By assumption $2^{k_\beta} \geq 2edk$. We then use Lemma 5.3 with $s = k$. This gives us the following

$$\mathbf{Pr}\left[\text{all hyperedges in } T \text{ are bad}\right] \leq \left(\frac{1}{2}\right)^{(k_\alpha - 1)\ell} \exp\left(\frac{(k_\alpha - 1)\ell}{k}\right).$$

Note that the maximum degree of the graph $\mathrm{Lin}(H)$ is at most dk. By Corollary 5.7 and a union bound over all 2-trees of size ℓ containing the hyperedge e, we have

$$\begin{aligned}\mathbf{Pr}\left[\mathcal{B}_e\right] &\leq \frac{(ed^2k^2)^{\ell-1}}{2} \cdot \left(\frac{1}{2}\right)^{(k_\alpha-1)\ell} \cdot \exp\left(\frac{(k_\alpha-1)\ell}{k}\right) \\ &\leq \frac{1}{2ed^2k^2}\left(\frac{2e^2d^2k^2}{2^{k_\alpha}}\right)^{\ell},\end{aligned}$$

where the last inequality holds because $k_\alpha - 1 \leq k$. By assumption $2^{k_\alpha} \geq 4e^2d^2k^2$, and thus for any $e \in \mathcal{E}$

$$\mathbf{Pr}\left[\mathcal{B}_e\right] \leq d^{-1}2^{-\ell-1}.$$

By (28), we have

$$\mathbf{Pr}\left[\mathcal{B}_t\right] \leq \sum_{e \in \mathcal{E}} \mathbf{Pr}\left[\mathcal{B}_e\right] \leq nd \cdot d^{-1}2^{-\ell-1} = n2^{-\ell-1} \leq \delta,$$

since $\ell = \lfloor L/(kd) \rfloor \geq \log \frac{n}{\delta} - 1$. □

6. Analyze the main sampling algorithm

Now we can finish the analysis of the main sampling algorithm, Algorithm 2.

Theorem 6.1. *The following holds for all $\xi \geq 0$. There is an algorithm such that given any $0 < \varepsilon < 1$ and (k,d)-formula Φ with n variables where $k \geq 20\log k + 20\log d + 60 + \xi$, it outputs a random assignment X of Φ satisfying $d_{\mathrm{TV}}(X,\mu) \leq \varepsilon$, where μ is the uniform distribution of satisfying assignments of Φ. The algorithm terminates in time $O\left(n\left(\frac{n}{\varepsilon}\right)^{\eta} d^2k^3\log^3\frac{n}{\varepsilon}\right)$, where $\eta = \left(\frac{1}{2}\right)^{20+\xi/3}\left(\frac{1}{dk}\right)^9$.*

The sampling result in Theorem 1.1 is a corollary of Theorem 6.1. We can set the parameter ζ in Theorem 1.1 as $\zeta = \left(\frac{1}{2}\right)^{20+\xi/3}$. The running time of the sampling algorithm in Theorem 6.1 is

$$O\left(n\left(\frac{n}{\varepsilon}\right)^{\zeta(dk)^{-9}} d^2k^3\log^3\frac{n}{\varepsilon}\right) = \widetilde{O}\left(d^2k^3n\left(\frac{n}{\varepsilon}\right)^{\zeta}\right).$$

We first prove Lemma 3.3. Then we use Lemma 3.3 to prove Theorem 6.1.

Proof of Lemma 3.3. We first couple X_T of Algorithm 2 with the idealized Glauber dynamics P_{Glauber}. At each step of the Markov chain, we couple the outcome of Sample with the idealized chain optimally. Coupling errors comes from the event $\mathcal{B}_t$ and the failure of rejection sampling. By Lemma 5.2, with probability at most $\delta = \frac{\varepsilon}{4(T+1)}$, event $\mathcal{B}_t$ happens. When $\mathcal{B}_t$ does not happen, by Lemma 5.1, the output of Sample is within total variation distance δ from the desired output. By Proposition 2.4, we can successfully couple it with the ideal output with probability at least $1-\delta$. Thus, X_T of Algorithm 2 can be coupled with the T-th step of P_{Glauber} with probability at least $1 - 2T\delta$.

Consider a sample X_{Glauber} by first running P_{Glauber} for T steps to get $X'_T \in \{0,1\}^{\mathcal{M}}$, and then draw from $\mu_{V\setminus\mathcal{M}}(\cdot \mid X'_T)$. In Line 6 of Algorithm 2, by Lemma 5.1 and Lemma 5.2, Sample returns a sample within TV distance δ from $\mu_{V\setminus\mathcal{M}}(\cdot \mid X_T)$ with probability at least $1-\delta$. Thus by Proposition 2.4 once again,

$$d_{\mathrm{TV}}\left(X_{\mathsf{alg}}, X_{\mathsf{Glauber}}\right) \leq 2(T+1)\delta = \frac{\varepsilon}{2}.$$

Moreover, consider an optimal algorithm which first obtains a perfect sample $X_{\mathcal{M}}$ from $\mu_{\mathcal{M}}$, and then complete it to all V by sampling from $\mu_{V\setminus\mathcal{M}}(\cdot \mid X_{\mathcal{M}})$. Call this sample X_{ideal}, and then the law of X_{ideal} is μ. By Proposition 2.4 and Lemma 4.1,

$$d_{\mathrm{TV}}\left(X_{\mathsf{Glauber}}, X_{\mathsf{ideal}}\right) \le \frac{\varepsilon}{4}$$

Combining everything we have that

$$d_{\mathrm{TV}}\left(X_{\mathsf{alg}}, \mu\right) = d_{\mathrm{TV}}\left(X_{\mathsf{alg}}, X_{\mathsf{ideal}}\right) \le d_{\mathrm{TV}}\left(X_{\mathsf{alg}}, X_{\mathsf{Glauber}}\right) + d_{\mathrm{TV}}\left(X_{\mathsf{Glauber}}, X_{\mathsf{ideal}}\right) \le \frac{3\varepsilon}{4}. \qquad \square$$

We now have all ingredients to show Theorem 6.1.

Proof of Theorem 6.1. We first assume $2^k \ge (2edk)^{\frac{6\ln 2\cdot(1+\alpha-\beta)}{(1-\alpha-\beta)^2}}$. Since we use the algorithm in Lemma 3.2 with $\delta = \frac{\varepsilon}{4}$ to construct the set $\mathcal{M}$, we have

$$\mathbf{Pr}\left[\text{the set } \mathcal{M} \text{ satisfying Condition 3.1 is constructed successfully}\right] \ge 1 - \frac{\varepsilon}{4}.$$

Let $X_{\mathsf{out}} \in \{0,1\}^V$ be the final assignment returned by our algorithm. If our algorithm fails to construct the set $\mathcal{M}$, then X_{out} is an arbitrary assignment in $\{0,1\}^V$; otherwise $X_{\mathsf{out}} = X_{\mathsf{alg}}$. Adding all errors together, Proposition 2.4 implies that

$$d_{\mathrm{TV}}\left(X_{\mathsf{out}}, \mu\right) \le \varepsilon.$$

Finally, we set the parameters k_α, k_β in Condition 3.1 and η in (9). We list all the constraints together

$$\begin{aligned}
2^k &\ge (2edk)^{\frac{6\ln 2\cdot(1+\alpha-\beta)}{(1-\alpha-\beta)^2}}, \quad \text{where } \alpha = \frac{k_\alpha}{k}, \beta = \frac{k_\beta}{k};\\
2^{k_\alpha} &\ge 4e^2d^2k^2;\\
2^{k_\beta} &\ge \frac{20}{\eta}edk, \quad \text{where } 0 < \eta < 1;\\
2^{k_\beta} &\ge 2^{16}d^9k^9;\\
k_\alpha &\ge 1;\\
k_\beta &\ge 1;\\
k_\alpha + k_\beta &\le k.
\end{aligned}$$

We can take

$$\begin{aligned}
k_\alpha &= \lfloor 0.1133k \rfloor,\\
k_\beta &= \lfloor 0.5097k \rfloor.
\end{aligned} \tag{31}$$

For any $\xi \ge 0$, if

$$k \ge 20\log k + 20\log d + 60 + \xi, \tag{32}$$

then it must hold that $k \ge 60$ and all the constraints are satisfied with k_α and k_β set as in (31). We can set η as

$$\eta \triangleq \left(\frac{1}{2}\right)^{20+\xi/3}\left(\frac{1}{dk}\right)^9. \tag{33}$$

Note that (32) implies $2^k \ge 2^{\xi+60}d^{20}k^{20}$. We can verify that

$$\frac{20}{\eta}edk = 20e\cdot 2^{20+\xi/3}d^{10}k^{10} \le 2^{30+\xi/2-1}d^{10}k^{10} \le 2^{\frac{k}{2}-1} \le 2^{k_\beta}.$$

We then analyze the time complexity of our algorithm. Since we run the algorithm in Lemma 3.2 with $\delta = \frac{\varepsilon}{4}$, then its time complexity is at most

$$T_{\mathsf{mark}} = O\left(ndk\log\frac{4}{\varepsilon}\right).$$

In Algorithm 2, the first $T \triangleq \left\lceil 2n \log \frac{4n}{\varepsilon} \right\rceil$ calls of the subroutine Sample(Φ, δ, X, S) satisfy $|S| = 1$ and the last call the of the subroutine Sample(Φ, δ, X, S) satisfies $|S| \le n$. By Lemma 5.1, we have

$$T_{\mathsf{alg}} = O\left(T\left(\frac{n}{\delta}\right)^{\frac{\eta}{10}} d^2k^3 \log^2 \frac{n}{\delta}\right) + O\left(n\left(\frac{n}{\delta}\right)^{\frac{\eta}{10}} d^2k^3 \log^2 \frac{n}{\delta}\right),$$

where $T = \left\lceil 2n \log \frac{4n}{\varepsilon} \right\rceil$, $\delta = \frac{\varepsilon}{4(T+1)}$ and η is defined in (33). Note that

$$\left(\frac{n}{\delta}\right)^{\frac{\eta}{10}} = O\left(\left(\frac{n}{\varepsilon}\right)^{\eta}\right).$$

This implies

$$T_{\mathsf{alg}} = O\left(n\left(\frac{n}{\varepsilon}\right)^{\eta} d^2k^3 \log^3 \frac{n}{\varepsilon}\right).$$

The total time complexity of our algorithm is

$$T = T_{\mathsf{mark}} + T_{\mathsf{alg}} = O\left(n\left(\frac{n}{\varepsilon}\right)^{\eta} d^2k^3 \log^3 \frac{n}{\varepsilon}\right). \qquad \square$$

7. Approximate counting

Let $\Phi = (V, C)$ be a k-CNF formula. One way to reduce counting to sampling is to start from a CNF formula with n variables and no clause. Then add clauses one by one and use the self-reducibility [JVV86] to count the number of solutions for Φ. This standard method gives an approximate counting algorithm which requires $\widetilde{O}(n^2d^2)$ calls to the sampling algorithm for a constant ε ($\widetilde{O}$ hides logarithmic factors).

Instead, we give a faster counting algorithm based on the simulated annealing method [BŠVV08, ŠVV09, Hub15, Kol18]. We will show that a non-adaptive annealing schedule with $\widetilde{O}(nd)$ calls to the sampling algorithm suffices (for a constant ε). The detailed time complexity bound is given in Theorem 7.1.

Theorem 7.1. *The followings hold for all $\xi \ge 0$. There is an algorithm such that given any $\varepsilon > 0$ and (k, d)-formula Φ with n variables where $k \ge 20 \log k + 20 \log d + 60 + \xi$, it outputs a number $\widehat{Z}$ that satisfies $\exp(-\varepsilon)Z \le \widehat{Z} \le \exp(\varepsilon)Z$ with probability at least $\frac{3}{4}$, where Z is the number of satisfying assignments of Φ. The algorithm terminates in time $O\left(\left(\frac{n}{\varepsilon}\right)^{2+\eta} d^3k^3 \log^{4+\eta} \frac{nd}{\varepsilon}\right)$, where $\eta = \left(\frac{1}{2}\right)^{19+\xi/3} \left(\frac{1}{dk}\right)^9$.*

The counting result in Theorem 1.1 is a corollary of Theorem 7.1. We can set the parameter ζ in Theorem 1.1 as $\zeta = \left(\frac{1}{2}\right)^{20+\xi/3}$. The running time of the counting algorithm in Theorem 7.1 is

$$O\left(\left(\frac{n}{\varepsilon}\right)^{2+2\zeta(dk)^{-9}} d^3k^3 \log^{4+2\zeta(dk)^{-9}} \frac{nd}{\varepsilon}\right) = \widetilde{O}\left(d^3k^3\left(\frac{n}{\varepsilon}\right)^{2+\zeta}\right),$$

where the equation holds due to $2(dk)^{-9} \le 1$.

7.1. The counting algorithm.

Recall $\Phi = (V, C)$ is a k-CNF formula. Given any parameter $\theta > 0$, for any $X \in \{0,1\}^V$, define the weight function:

$$w_\theta(X) \triangleq \exp(-\theta |F(X)|),$$

where $F(X) \subseteq C$ is the set of clauses that are not satisfied by X. Let the partition function $Z(\theta)$ be

$$Z(\theta) \triangleq \sum_{X \in \{0,1\}^V} w_\theta(X).$$

Then the Gibbs distribution μ_θ over $\{0,1\}^V$ is given by

$$\forall X \in \{0,1\}^V : \quad \mu_\theta(X) \triangleq \frac{w_\theta(X)}{Z(\theta)}, \tag{34}$$

Let Z denote the number of satisfying assignments for Φ, then we have

$$Z = \lim_{\theta \to \infty} Z(\theta).$$

Let $\ell = nd\left\lceil \ln \frac{4nd}{\varepsilon} \right\rceil$. Define a sequence of parameters $(\theta_i)_{i\geq 0}$ as

$$\forall i \in \mathbb{Z}_{\geq 0}: \quad \theta_i = \frac{i}{dn}. \tag{35}$$

The following lemma shows that the partition function $Z(\theta_\ell)$ is close to Z.

Lemma 7.2. *If $2^k \geq 2edk$, then given any $\varepsilon > 0$, it holds that*

$$Z \leq Z(\theta_\ell) \leq \exp\left(\frac{\varepsilon}{2}\right) Z.$$

The proof of Lemma 7.2 is deferred to Section 7.2. Note that the condition for Φ in Lemma 7.2 is weaker than that in Theorem 7.1. By Lemma 7.2, we can use $Z(\theta_\ell)$ to approximate the value of Z. We estimate the value of $Z(\theta_\ell)$ by the following telescoping product

$$Z(\theta_\ell) = \frac{Z(\theta_\ell)}{Z(\theta_{\ell-1})} \times \frac{Z(\theta_{\ell-1})}{Z(\theta_{\ell-2})} \times \ldots \times \frac{Z(\theta_1)}{Z(\theta_0)} \times 2^n, \tag{36}$$

where the equation holds because $\theta_0 = 0$ and $Z(\theta_0) = 2^n$.

We now estimate the value of each ratio $\frac{Z(\theta_{i+1})}{Z(\theta_i)}$ in (36). Let $\mu_i = \mu_{\theta_i}$ denote the Gibbs distribution specified by the parameter θ_i. Let $w_i(\cdot) = w_{\theta_i}(\cdot)$ denote the weight function for Gibbs distribution μ_i. For each $1 \leq i \leq \ell$, we define the random variable W_i as

$$W_i \triangleq \frac{w_i(X)}{w_{i-1}(X)}, \quad \text{where } X \sim \mu_{i-1}.$$

We then define W as 2^n times the product of all random variables W_i:

$$W = 2^n \prod_{i=1}^{\ell} W_i.$$

We have the following lemma for W and each W_i.

Lemma 7.3. *For each $1 \leq i \leq \ell$, the random variable W_i satisfies*

$$\mathbf{E}[W_i] = \frac{Z(\theta_i)}{Z(\theta_{i-1})}, \quad \mathbf{E}\left[W_i^2\right] = \frac{Z(\theta_{i+1})}{Z(\theta_{i-1})}.$$

Hence, the random variable W satisfies

$$\mathbf{E}[W] = Z(\theta_\ell), \quad \mathbf{E}\left[W^2\right] = \frac{4^n Z(\theta_\ell) Z(\theta_{\ell+1})}{Z(\theta_0) Z(\theta_1)}.$$

Proof. By the definition of W_i, we have

$$\mathbf{E}[W_i] = \sum_{X \in \{0,1\}^V} \frac{w_{i-1}(X)}{Z(\theta_{i-1})} \times \frac{w_i(X)}{w_{i-1}(X)} = \frac{Z(\theta_i)}{Z(\theta_{i-1})}.$$

For each $X \in \{0,1\}^V$, it holds that $w_i(X) = \exp\left(-\frac{i}{dn}|F(X)|\right)$. We have

$$\mathbf{E}\left[W_i^2\right] = \sum_{X \in \{0,1\}^V} \frac{w_{i-1}(X)}{Z(\theta_{i-1})} \times \left(\frac{w_i(X)}{w_{i-1}(X)}\right)^2 = \sum_{X \in \{0,1\}^V} \frac{w_{i+1}(X)}{Z(\theta_{i-1})} = \frac{Z(\theta_{i+1})}{Z(\theta_{i-1})}.$$

Note that all W_i are independent. By the definition of W, we have

$$\mathbf{E}[W] = 2^n \prod_{i=1}^{\ell} \mathbf{E}[W_i] = 2^n \times \frac{Z(\theta_\ell)}{Z(\theta_0)} = Z(\theta(\ell));$$

$$\mathbf{E}\left[W^2\right] = 4^n \times \prod_{i=1}^{\ell} \mathbf{E}\left[W_i^2\right] = 4^n \times \frac{Z(\theta_\ell) Z(\theta_{\ell+1})}{Z(\theta_0) Z(\theta_1)}.$$

□

By Lemma 7.3, the expectation of W is precisely the partition function $Z(\theta_\ell)$. If we can draw random samples from each distribution μ_i, then we can compute all W_i and W using these random samples. In Section 3, we have given an algorithm that samples CNF solutions uniformly at random. With a simple modification, we have the following algorithm that samples assignments from the Gibbs distribution in (34).

Lemma 7.4. *Let $\xi \geq 0$ and Φ be a (k,d)-formula with n variables where $k \geq 20\log k + 20\log d + 60 + \xi$. There is an algorithm $\mathcal{A}$ such that given any $0 < \delta < 1$ and any $\theta \geq 0$, the algorithm $\mathcal{A}(\theta,\delta)$ outputs a random assignment X of Φ satisfying $d_{\mathrm{TV}}(X,\mu_\theta) \leq \delta$, where μ_θ is the Gibbs distribution defined in (34). The algorithm terminates in time $O\left(n\left(\frac{n}{\delta}\right)^\eta d^2k^3\log^3\frac{n}{\delta}\right)$, where $\eta = \left(\frac{1}{2}\right)^{20+\xi/3}\left(\frac{1}{dk}\right)^9$.*

Our counting algorithm is described in Algorithm 7. It relies on the Algorithm $\mathcal{A}$ in Lemma 7.4 as a subroutine.

Algorithm 7: The counting algorithm

Input : a CNF formula $\Phi = (V,C)$, a parameter $\varepsilon > 0$.
Output: a number $\widehat{Z}$.

```
for each j from 1 to m = ⌈144ε^{-2}⌉ do
    for each i = 1 to ℓ = nd⌈ln 4nd/ε⌉ do
        use A(θ_{i-1}, 1/(8ℓm)) to draw sample X_i^j ∈ {0,1}^V independently;
        Ŵ_i^j ← w_i(X_i^j)/w_{i-1}(X_i^j);
    Ŵ^j ← 2^n · ∏_{i=1}^{ℓ} Ŵ_i^j;
return Ẑ = (1/m) ∑_{j=1}^{m} Ŵ^j;
```

To prove that correctness of Algorithm 7, we need the following lemma.

Lemma 7.5. *Let $\mathcal{B}$ be a sampling oracle such that given any parameter θ, $\mathcal{B}(\theta)$ returns a perfect sample from the distribution μ_θ. Suppose we replace $\mathcal{A}(\theta_{i-1}, 1/(8\ell m))$ in Line 3 of Algorithm 7 with $\mathcal{B}(\theta_{i-1})$. Denote the output of the modified algorithm by $\widehat{Z}_{\mathcal{B}}$. Then, it holds that*

$$\mathbf{Pr}\left[\exp(-\varepsilon/2)Z(\theta_\ell) \leq \widehat{Z}_{\mathcal{B}} \leq \exp(\varepsilon/2)Z(\theta_\ell)\right] \geq 7/8.$$

Proof. By the assumption in Lemma 7.5, we know that each $\widehat{W}^j$ is a perfect sample from the distribution of the random variable W. Note that $\widehat{Z}_{\mathcal{B}} = \frac{1}{m}\sum_{i=1}^m \widehat{W}^i$. Hence $\mathbf{E}\left[\widehat{Z}_{\mathcal{B}}\right] = \mathbf{E}[W] = Z(\theta_\ell)$. By Chebyshev's inequality, we have

$$\mathbf{Pr}\left[\left|\widehat{Z}_{\mathcal{B}} - \mathbf{E}\left[\widehat{Z}_{\mathcal{B}}\right]\right| \geq (\varepsilon/3)\mathbf{E}\left[\widehat{Z}_{\mathcal{B}}\right]\right] \leq \frac{9\mathbf{Var}\left[\widehat{Z}_{\mathcal{B}}\right]}{\varepsilon^2\mathbf{E}\left[\widehat{Z}_{\mathcal{B}}\right]^2} = \frac{9\mathbf{Var}[W]}{m\varepsilon^2\mathbf{E}[W]^2} \tag{37}$$

By Lemma 7.3, we have

$$\frac{\mathbf{Var}[W]}{\mathbf{E}[W]^2} = \frac{\mathbf{E}\left[W^2\right]}{\mathbf{E}[W]^2} - 1 = \frac{Z(\theta_{\ell+1})Z(\theta_0)}{Z(\theta_\ell)Z(\theta_1)} - 1,$$

where the last equation holds because $\mathbf{E}[W] = Z(\theta_\ell)$, $\mathbf{E}\left[W^2\right] = \frac{4^nZ(\theta_\ell)Z(\theta_{\ell+1})}{Z(\theta_0)Z(\theta_1)}$ and $Z(\theta_0) = 2^n$. Note that $Z(\theta_{\ell+1}) \leq Z(\theta_\ell)$, we have

$$\frac{Z(\theta_{\ell+1})}{Z(\theta_\ell)} \leq 1.$$

By the definition of the partition function $Z(\cdot)$, we have

$$\frac{Z(\theta_0)}{Z(\theta_1)} = \frac{\sum_{X\in\{0,1\}^V} w_0(X)}{\sum_{X\in\{0,1\}^V} w_1(X)} = \frac{\sum_{X\in\{0,1\}^V} 1}{\sum_{X\in\{0,1\}^V}\exp(-\theta_1|F(X)|)} \leq \max_{X\in\{0,1\}^V}\exp(\theta_1|F(X)|) \leq \mathrm{e}.$$

The last inequality is due to the fact that $\theta_1 = \frac{1}{nd}$ and $|F(X)| \le nd$.

Hence, we can bound (37) as follows

$$\mathbf{Pr}\left[\left|\widehat{Z}_{\mathcal{B}} - \mathbf{E}\left[\widehat{Z}_{\mathcal{B}}\right]\right| \le (\varepsilon/3)\mathbf{E}\left[\widehat{Z}_{\mathcal{B}}\right]\right] \le \frac{9(\mathrm{e}-1)}{m\varepsilon^2} \le \frac{1}{8},$$

where the last inequality holds because $m = \lceil 144\varepsilon^{-2}\rceil$. Note that $\mathbf{E}\left[\widehat{Z}_{\mathcal{B}}\right] = Z(\theta_\ell)$ due to the assumption in Lemma 7.5. This proves that

$$\begin{aligned}&\mathbf{Pr}\left[\exp(-\varepsilon/2)Z(\theta_\ell) \le \widehat{Z}_{\mathcal{B}} \le \exp(\varepsilon/2)Z(\theta_\ell)\right]\\ \ge\ &\mathbf{Pr}\left[(1-\varepsilon/3)Z(\theta_\ell) \le \widehat{Z}_{\mathcal{B}} \le (1+\varepsilon/3)Z(\theta_\ell)\right] \ge \frac{7}{8}.\end{aligned}$$

□

We then construct a coupling C between Algorithm 7 and the algorithm in Lemma 7.5. For each execution of Line 3, we use the optimal coupling to couple the random sample returned by algorithm $\mathcal{A}(\theta_{i-1}, 1/(8\ell m))$ and the random sample returned by $\mathcal{B}(\theta_{i-1})$. By 2.4, with probability at least $1 - 1/(8\ell m)$, two samples are perfectly coupled. Since Line 3 is executed for ℓm times, then by a union bound, with probability at least $\frac{7}{8}$, two algorithms obtain the same output, i.e. $\widehat{Z} = \widehat{Z}_{\mathcal{B}}$. Combining with Lemma 7.5, we have

$$\begin{aligned}&\mathbf{Pr}\left[\widehat{Z} < \exp(-\varepsilon/2)Z(\theta_\ell) \vee \widehat{Z} > \exp(\varepsilon/2)Z(\theta_\ell)\right]\\ \le\ &\mathbf{Pr}\left[\widehat{Z}_{\mathcal{B}} < \exp(-\varepsilon/2)Z(\theta_\ell) \vee \widehat{Z}_{\mathcal{B}} > \exp(\varepsilon/2)Z(\theta_\ell)\right] + \mathbf{Pr}_C\left[\widehat{Z} \ne \widehat{Z}_{\mathcal{B}}\right]\\ \le\ &\frac{1}{4}.\end{aligned}$$

This proves that

$$\mathbf{Pr}\left[\exp(-\varepsilon/2)Z(\theta_\ell) \le \widehat{Z} \le \exp(\varepsilon/2)Z(\theta_\ell)\right] \ge 3/4.$$

By Lemma 7.2, we know that $Z(\theta_\ell)$ approximates the value of Z, where Z is the number of solutions for Φ. We have

$$\mathbf{Pr}\left[\exp(-\varepsilon)Z \le \widehat{Z} \le \exp(\varepsilon)Z\right] \ge 3/4.$$

This proves the correctness of Algorithm 7.

The time complexity of Algorithm 7 is dominated by the time complexity of generating random samples. In Algorithm 7, the sampling algorithm $\mathcal{A}$ in Lemma 7.4 is called for $m\ell$ times. Note that we only call algorithm $\mathcal{A}$ with $\delta = 1/(8m\ell)$. The time complexity of each call of $\mathcal{A}$ is

$$T_{\mathcal{A}} = O\left(n\left(\frac{n}{\varepsilon}\right)^{2\eta} d^{2+\eta}k^3 \log^{3+\eta}\frac{nd}{\varepsilon}\right),$$

where $\eta = \left(\frac{1}{2}\right)^{20+\xi/3}\left(\frac{1}{dk}\right)^9$. Note that $m\ell = O\left(\frac{nd}{\varepsilon^2}\log\frac{nd}{\varepsilon}\right)$. Then, the total time complexity of Algorithm 7 is at most

$$T_{\mathsf{count}} = O\left(\left(\frac{n}{\varepsilon}\right)^2\left(\frac{n}{\varepsilon}\right)^{2\eta} d^{3+\eta}k^3 \log^{4+\eta}\frac{nd}{\varepsilon}\right).$$

Let $\eta' = 2\eta = \left(\frac{1}{2}\right)^{19+\xi/3}\left(\frac{1}{dk}\right)^9$, we have

$$T_{\mathsf{count}} = O\left(\left(\frac{n}{\varepsilon}\right)^2\left(\frac{nd}{\varepsilon}\right)^{\eta'} d^3k^3 \log^{4+\eta'}\frac{nd}{\varepsilon}\right) = O\left(\left(\frac{n}{\varepsilon}\right)^{2+\eta'} d^3k^3 \log^{4+\eta'}\frac{nd}{\varepsilon}\right),$$

where the last equation holds due to $d^{d^{-9}} = O(1)$. This proves the time complexity of Algorithm 7.

7.2. **Comparing Z and $Z(\theta_\ell)$ (proof of Lemma 7.2).** We first prove a lemma stating that adding a new clause to a CNF formula decrease the number of solutions by at most half if the parameters are in the local lemma regime.

Lemma 7.6. *Let $\Phi = (V, C)$ be a k-CNF formula. Let $\Phi' = (V, C')$ be a new k-CNF formula obtained from Φ by adding a new clause f, i.e. $C' = C \cup \{f\}$. Suppose each variable belongs to at most d clauses in both Φ and Φ'. If $2^k \geq 2edk$, then it holds that*

$$\frac{Z_{\Phi'}}{Z_\Phi} \geq \frac{1}{2},$$

where Z_Φ is the number of solution for Φ and $Z_{\Phi'}$ is the number of solutions for Φ'.

Proof. Let μ and μ' denote the uniform distributions of all solutions for Φ and Φ' respectively. Note that if $X \in \{0,1\}^V$ is a solution for Φ', then it is a solution for Φ as well. Therefore, we have

$$\frac{Z_{\Phi'}}{Z_\Phi} = \mathbf{Pr}_{X\sim\mu}\left[X \text{ is a solution for } \Phi'\right] = \mathbf{Pr}_{X\sim\mu}\left[f \text{ is satisfied by } X\right]. \tag{38}$$

Recall that we use $\mathbf{Pr}_{\mathcal{P}}[\cdot]$ to denote the product distribution such that each variable $v \in V$ takes a value from $\{0,1\}$ uniformly and independently. Let $\mathcal{B}_c$ denote the bad event that the clause $c \in C$ is not satisfied. Note that, in Φ, each clause contains k variables and each variable belongs to at most d clauses. By Theorem 2.1, if we take $x(\mathcal{B}_c) = \frac{1}{2dk}$ for each $\mathcal{B}_c$, it holds that for any $c \in C$,

$$\mathbf{Pr}_{\mathcal{P}}[\mathcal{B}_c] = \left(\frac{1}{2}\right)^k \leq \frac{1}{2edk} \leq x(\mathcal{B}_c) \prod_{\mathcal{B}_{c'} \in \Gamma(\mathcal{B}_c)} (1 - x(\mathcal{B}_{c'})),$$

where $\Gamma(\mathcal{B}_c)$ contains all $\mathcal{B}_{c'}$ satisfying $c' \in C$, $c' \neq c$ and $\mathsf{vbl}(c) \cap \mathsf{vbl}(c') \neq \varnothing$. We use $\mathcal{F}$ to denote the event that f is not satisfied. Since each variable belongs to at most d clauses in Φ', we have

$$\mathbf{Pr}_{\mathcal{P}}\left[\mathcal{F} \mid \bigwedge_{c\in C} \overline{\mathcal{B}_c}\right] \leq \mathbf{Pr}_{\mathcal{P}}[\mathcal{F}] \left(1 - \frac{1}{2dk}\right)^{-dk} \leq 2\left(\frac{1}{2}\right)^k \leq \frac{1}{2}, \tag{39}$$

where the last inequality holds because $k \geq 2$ if $2^k \geq 2edk$. Note that the product distribution $\mathcal{P}$ conditioned on $\bigwedge_{c\in C} \overline{\mathcal{B}_c}$ is precisely the distribution μ. Combining (38) and (39), we have

$$\frac{Z_{\Phi'}}{Z_\Phi} = 1 - \mathbf{Pr}_{X\sim\mu}\left[f \text{ is not satisfied by } X\right] = 1 - \mathbf{Pr}_{\mathcal{P}}\left[\mathcal{F} \mid \bigwedge_{c\in C} \overline{\mathcal{B}_c}\right] \geq \frac{1}{2}. \qquad \square$$

For a k-CNF formula $\Phi = (V, C)$ and any subset $S \subseteq C$, we define the value Z_S as

$$Z_S \triangleq \#\left\{X \in \{0,1\}^V \;\middle|\; \begin{array}{l}\text{all clauses in } S \text{ are not satisfied by } X,\\ \text{and all clauses in } C \setminus S \text{ are satisfied by } X\end{array}\right\}. \tag{40}$$

The value Z_S counts the number of those assignments $X \in \{0,1\}^V$ satisfying *exactly* the clauses in $C \setminus S$. The next lemma bounds the size of Z_S.

Lemma 7.7. *Suppose each variable belongs to at most d clauses. If $2^k \geq 2edk$, then for any $S \subseteq C$, it holds that $Z_S \leq 2^{|S|} Z_\varnothing$.*

Proof. Let $S \subseteq C$ be a set of clauses with $|S| = k$. Suppose $S = \{c_1, c_2, \ldots, c_k\}$. We define a sequence of CNF formulas $\Phi_0, \Phi_1, \ldots, \Phi_k$. For each $\Phi_i = (V, C_i)$, the set of clauses C_i is defined as

$$C_i \triangleq (C \setminus S) \cup \{c_j \mid 1 \leq j \leq i\}.$$

This is equivalent to let $C_0 = C \setminus S$ and $C_i = C_{i-1} \cup \{c_i\}$ for every $1 \leq i \leq k$. For every $0 \leq i \leq k-1$, since each Φ_i is a subformula of Φ, the condition $2^k \geq 2edk$ still holds and we can apply Lemma 7.6 for each Φ_i (with $\Phi = \Phi_i$ and $\Phi' = \Phi_{i+1}$ in the statement of Lemma 7.6). This yields

$$\frac{Z_{\Phi_k}}{Z_{\Phi_0}} = \prod_{j=1}^{k} \frac{Z_{\Phi_j}}{Z_{\Phi_{j-1}}} \geq 2^{-k}, \tag{41}$$

where Z_{Φ_i} is the number of solutions for Φ_i.

On the other hand, by the definition of Z_S, we have

$$Z_{\Phi_k} = Z_{\Phi} = Z_{\varnothing}, \text{ and } Z_{\Phi_0} = \sum_{S' \subseteq S} Z_{S'}.$$

Combining with Equation (41), we obtain

$$\frac{Z_{\varnothing}}{\sum_{S' \subseteq S} Z_{S'}} \geq 2^{-k}.$$

Hence, we have

$$Z_S \leq \sum_{S' \subseteq S} Z_{S'} \leq 2^k Z_{\varnothing}. \qquad \square$$

We now prove Lemma 7.2. By the definition of $Z(\theta_\ell)$, the lower bound $Z(\theta_\ell) \geq Z$ clearly holds. For the upper bonud, noting that $\theta_\ell = \lceil \ln \frac{4nd}{\varepsilon} \rceil$, we have

$$Z(\theta_\ell) = \sum_{X \in \{0,1\}^V} \exp(-\theta_\ell F(X)) \leq \sum_{X \in \{0,1\}^V} \left(\frac{\varepsilon}{4nd}\right)^{|F(X)|},$$

where $F(X) \subseteq C$ is the set of clauses that are not satisfied by X. By the definition of Z_S, we have

$$\begin{aligned}
Z(\theta_\ell) &\leq \sum_{X \in \{0,1\}^V} \left(\frac{\varepsilon}{4nd}\right)^{|F(X)|} = \sum_{S \subseteq C} \left(\frac{\varepsilon}{4nd}\right)^{|S|} Z_S \\
\text{(by Lemma 7.7)} \qquad &\leq \sum_{S \subseteq C} \left(\frac{\varepsilon}{4nd}\right)^{|S|} 2^{|S|} Z_{\varnothing} = Z_{\varnothing} \sum_{k=0}^{|C|} \binom{|C|}{k} \left(\frac{\varepsilon}{4nd}\right)^k 2^k \\
\text{(as } |C| \leq nd\text{)} \qquad &= Z_{\varnothing} \left(1 + \frac{\varepsilon}{2nd}\right)^{|C|} \leq Z_{\varnothing} \left(1 + \frac{\varepsilon}{2nd}\right)^{nd} \\
\text{(as } Z = Z_{\varnothing}\text{)} \qquad &\leq Z \exp\left(\frac{\varepsilon}{2}\right).
\end{aligned}$$

This finishes the proof of Lemma 7.2.

7.3. **The modified sampling algorithm (proof of Lemma 7.4).** In this section, we give a modified sampling algorithm to sample from the Gibbs distribution μ_θ defined in (34). Given a CNF formula $\Phi = (V, E)$ and a parameter $\theta \geq 0$, we introduce $|C|$ extra variables

$$U \triangleq \{u_c \in \{0,1\} \mid c \in C\}.$$

We now define a new CNF formula $\Phi' = (V \cup U, C')$. The set of clauses C' is defined as

$$C' \triangleq \{u_c \vee c \mid c \in C\}.$$

Hence, given any assignment $X \in \{0,1\}^{V \cup U}$, a clause $c' = u_c \vee c$ is satisfied by X if $X(u_c) = 1$ or the clause c is satisfied by X.

Observation 7.8. *The CNF formula $\Phi' = (V \cup U, C')$ is $k+1$ uniform and each variable $u \in U$ belongs to only one clause.*

Let $\mathcal{P}$ denote the product distribution over $\{0,1\}^{V \cup U}$ such that each variable $v \in V$ takes value from $\{0,1\}$ uniformly, and each variable $u \in U$ takes value 1 with probability $\exp(-\theta)$ and takes value 0 with probability $1 - \exp(-\theta)$. For each clause $c' \in C'$, we define a bad event $\mathcal{B}_{c'}$ as the clause c' is not satisfied. Recall that μ_θ is the Gibbs distribution defined in (34). We have the following proposition.

Proposition 7.9. *For any assignment $X \in \{0,1\}^V$, it holds that*

$$\mathbf{Pr}_{\mathcal{P}}\left[\textit{each variable } v \in V \textit{ takes the value } X(v) \;\middle|\; \bigwedge_{c' \in C'} \overline{\mathcal{B}_{c'}}\right] = \mu_\theta(X).$$

Proof. We use $F(X(V))$ to denote the set of clauses $c \in C$ (C is the set of clauses in original CNF formula Φ) such that c is not satisfied by $X(V)$. Consider a clause $c' \in C'$ where $c' = u_c \vee c$. Suppose the bad event $\mathcal{B}_{c'}$ does not occur. If $c \in F(X(V))$, then u_c must take the value 1. If $c \notin F(X(V))$, then u_c can take an arbitrary value from $\{0,1\}$. We have

$$\begin{aligned}&\mathbf{Pr}_{\mathcal{P}}\left[(\text{each variable } v \in V \text{ takes the value } X(v)) \ \wedge \left(\bigwedge_{c' \in C'} \overline{\mathcal{B}_{c'}}\right)\right]\\ =& \left(\frac{1}{2}\right)^{|V|} \exp(-\theta|F(X(V))|) = \left(\frac{1}{2}\right)^{|V|} w_\theta(X(V)),\end{aligned}$$

where $w_\theta(\cdot)$ is the weight function for μ_θ. Therefore,

$$\begin{aligned}&\mathbf{Pr}_{\mathcal{P}}\left[\text{each variable } v \in V \text{ takes the value } X(v) \;\middle|\; \bigwedge_{c' \in C'} \overline{\mathcal{B}_{c'}}\right]\\ =& \frac{w_\theta(X(V))}{2^{|V|}\mathbf{Pr}_{\mathcal{P}}\left[\bigwedge_{c' \in C'} \overline{\mathcal{B}_{c'}}\right]}\\ =& \frac{w_\theta(X(V))}{2^{|V|}\sum_{X' \in \{0,1\}^V} \mathbf{Pr}_{\mathcal{P}}\left[(\text{each variable } v \in V \text{ takes the value } X'(v)) \ \wedge \bigwedge_{c' \in C'} \overline{\mathcal{B}_{c'}}\right]}\\ =& \frac{w_\theta(X(V))}{\sum_{X' \in \{0,1\}^V} w_\theta(X'(V))} = \mu_\theta(X).\end{aligned}$$

□

Let μ' denote the product distribution $\mathcal{P}$ over $\{0,1\}^{V \cup U}$ conditioned on none of the bad event $\mathcal{B}_{c'}$ for $c' \in C'$ occurs. Our aim is to sample $X \in \{0,1\}^{V \cup U}$ such that

$$d_{\mathrm{TV}}(X, \mu') \le \delta,$$

which, by Proposition 7.9, implies that

$$d_{\mathrm{TV}}(X(V), \mu_\theta) \le \delta.$$

Recall that $\Phi' = (V \cup U, C')$ is a $(k+1)$-uniform CNF formula. We describe how to modify our algorithm in Section 3 to sample from the Gibbs distribution μ'.

The first step is to mark variables. We construct a set of marked variables $\mathcal{M} \subseteq V$ such that each clause $c' \in C'$ contains at least k_α marked variables and at least k_β unmarked variables. Note that we do not mark variables in the set U, i.e. $U \cap \mathcal{M} = \varnothing$. This step can be accomplished by the Moser-Tardos algorithm in Section 3.1.

We define the Glauber dynamics chain $(X_t)_{t \ge 0}$ for marked variables, whose stationary distribution is the marginal distribution $\mu'_{\mathcal{M}}$ on $\mathcal{M}$ projected from μ'. We start with an initial assignment $X_0 \in \{0,1\}^{\mathcal{M}}$ where $X_0(v)$ is uniformly at random for all $v \in \mathcal{M}$. In the t-th step, the chain evolves as follows:

- pick $v \in \mathcal{M}$ uniformly at random and set $X_t(u) \leftarrow X_{t-1}(u)$ for all $u \in \mathcal{M} \setminus \{v\}$;
- sample $X_t(v) \in \{0,1\}$ from the distribution $\mu'_v(\cdot \mid X_{t-1}(\mathcal{M} \setminus \{v\}))$.

We use Algorithm 2 to simulate the Glauber dynamics chain defined above. There are two modifications. First, in Line 4, we use the subroutine $\mathrm{Sample}(\Phi', \frac{\varepsilon}{4(T+1)}, X_{t-1}(\mathcal{M} \setminus \{v\}), \{v\})$ to draw random sample $X_t(v) \in \{0,1\}$. Second, in Line 6, we use the subroutine $\mathrm{Sample}(\Phi', \frac{\varepsilon}{4(T+1)}, X_T, V \cup U \setminus \mathcal{M})$ to draw the random sample $X_{V \cup U \setminus \mathcal{M}}$.

We also need to adjust the Sample subroutine in the rejection sampling step

$$Y_i^X \leftarrow \mathsf{RejectionSampling}\left(\Phi_i^X, R\right),$$

namely Line 7 of $\mathrm{Sample}(\Phi, \delta, X, S)$. Recall that $\Phi_i^X = (V_i^X, C_i^X)$. In the rejection sampling, for each variable $v \in V_i^X \cap V$, we sample its value from $\{0,1\}$ uniformly and independently; for each variable $u \in V_i^X \cap U$, we sample its value from $\{0,1\}$ independently such that $\mathbf{Pr}[u = 1] = \exp(-\theta)$.

We need to verify Lemma 4.1 and Lemma 3.3 for the algorithm above. Due to the definition of $\Phi' = (V \cup U, C')$ and Observation 7.8, the following two facts hold for Φ':

- each variable in Φ' belongs to at most d clauses;
- for any $c' \in C'$, it holds that $\mathbf{Pr}_{\mathcal{P}}[\mathcal{B}_{c'}] = \exp(-\theta)\left(\frac{1}{2}\right)^k \leq \left(\frac{1}{2}\right)^k$.

With these two facts, we can verify that all results based on the local lemma still hold for Φ' with the product distribution $\mathcal{P}$. An analogue to Lemma 3.3 holds by the identical proof in Section 5.

To prove the rapid mixing analogue to Lemma 4.1, we need to sightly modify the two coupling procedures C_v and C in Algorithm 5 and Algorithm 6, respectively. Let $X, Y \in \{0,1\}^{V\cup U}$ be two assignments for path coupling that disagree only on a variable $v_0 \in \mathcal{M}$. Recall that μ' is the Gibbs distribution specified by Φ'. Fix a variable $v \in \mathcal{M} \setminus \{v_0\}$. We use ν' to denote the distribution μ' conditional on the assignment of the set $\Lambda = \mathcal{M} \setminus \{v_0, v\}$ is specified by $X(\Lambda) = Y(\Lambda)$, where $X, Y \in \{0,1\}^{\mathcal{M}}$ differ at only v_0. Formally,

$$\forall \sigma \in \{0,1\}^{V\cup U}: \quad \nu'(\sigma) = \frac{\mathbf{1}\left[\sigma(\Lambda) = X(\Lambda)\right] \cdot \mu'(\sigma)}{\sum_{\tau\in\{0,1\}^{V\cup U}} \mathbf{1}\left[\tau(\Lambda) = X(\Lambda)\right] \cdot \mu'(\tau)}. \tag{42}$$

We define a hypergraph $H' \triangleq (V, \mathcal{E}')$ for $\Phi' = (V \cup U, C')$, which is obtained from $H_{\Phi'}$ (defined in (2)) by removing all variables in U. Namely, the variable set in H is V rather than $V \cup U$, and the hyperedge set $\mathcal{E}$ is defined by

$$\mathcal{E}' \triangleq \{V \cap \mathsf{vbl}(c') \mid c' \in C'\} = \{\mathsf{vbl}(c) \mid c \in C\}.$$

The two coupling procedures are modified as follows.

- Algorithm 5, C_v: the input hypergraph is H'. In Line 6, we set $p_u^X = \nu'_u(0 \mid X^{C_v})$ and $p_u^Y = \nu'_u(0 \mid Y^{C_v})$, where ν' is defined in (42); in Line 19, we use the optimal coupling between $\nu'_{U\cup V_1\setminus V_{\text{set}}}(\cdot \mid X^{C_v}(V_{\text{set}} \cup V_2))$ and $\nu'_{U\cup V_1\setminus V_{\text{set}}}(\cdot \mid Y^{C_v}(V_{\text{set}} \cup V_2))$ to extend X^{C_v} and Y^{C_v} further on the set $U \cup V_1 \setminus V_{\text{set}}$.
- Algorithm 6, C: the input hypergraph is H'. In Line 6, we set $p_u^X = \mu'_u(0 \mid X^C)$ and $p_u^Y = \mu'_u(0 \mid Y^C)$, where μ' is the Gibbs distribution defined in (34).

In other words, in the while-loop (namely, Line 4) of C_v and C, we do not choose any variable in U. However, the effect of U needs to be taken into consideration in the calculation of the probabilities p_u^X and p_u^Y in Line 6.

Consider the modified coupling procedure C_v. Let $V_1, V_2, \mathcal{S}, V_{\text{set}}$ and X^{C_v}, Y^{C_v} be the sets and assignments after the execution of C_v. Due to Observation 7.8, each variable $u \in U$ belongs to only one clauses. Then we can verify that two distributions $\nu'_{V_2\setminus V_{\text{set}}}(\cdot \mid X^{C_v}(V_{\text{set}}))$ and $\nu'_{V_2\setminus V_{\text{set}}}(\cdot \mid X^{C_v}(V_{\text{set}} \cap V_2))$ are identical, and two distributions $\nu'_{V_2\setminus V_{\text{set}}}(\cdot \mid Y^{C_v}(V_{\text{set}}))$ and $\nu'_{V_2\setminus V_{\text{set}}}(\cdot \mid Y^{C_v}(V_{\text{set}} \cap V_2))$ are identical. With these two facts, we can prove that $X^{C_v}(u) = Y^{C_v}(u)$ for all $u \in V_2$. Therefore the rapid mixing result, Lemma 4.1, follows from the identical proof in Section 4.

References

[BCKL13] Christian Borgs, Jennifer Chayes, Jeff Kahn, and László Lovász. Left and right convergence of graphs with bounded degree. *Random Struct. Algorithms*, 42(1):1–28, 2013.

[BD97] Russ Bubley and Martin Dyer. Path coupling: A technique for proving rapid mixing in Markov chains. In *FOCS*, pages 223–231, 1997.

[BGG+19] Ivona Bezáková, Andreas Galanis, Leslie Ann Goldberg, Heng Guo, and Daniel Štefankovič. Approximation via correlation decay when strong spatial mixing fails. *SIAM J. Comput.*, 48(2):279–349, 2019.

[BŠVV08] Ivona Bezáková, Daniel Štefankovič, Vijay V Vazirani, and Eric Vigoda. Accelerating simulated annealing for the permanent and combinatorial counting problems. *SIAM J. Comput.*, 37(5):1429–1454, 2008.

[CGG+19] Zongchen Chen, Andreas Galanis, Leslie Ann Goldberg, Will Perkins, James Stewart, and Eric Vigoda. Fast algorithms at low temperatures via Markov chains. In *APPROX-RANDOM*, volume 145 of *LIPIcs*, pages 41:1–41:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

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

[EHS+19] Charilaos Efthymiou, Thomas P. Hayes, Daniel Stefankovic, Eric Vigoda, and Yitong Yin. Convergence of MCMC and loopy BP in the tree uniqueness region for the hard-core model. *SIAM J. Comput.*, 48(2):581–643, 2019.

[FA17] Alan M. Frieze and Michael Anastos. Randomly coloring simple hypergraphs with fewer colors. *Inf. Process. Lett.*, 126:39–42, 2017.

[GJL19] Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the Lovász local lemma. *J. ACM*, 66(3):18:1–18:31, 2019.

[GLLZ19] Heng Guo, Chao Liao, Pinyan Lu, and Chihao Zhang. Counting hypergraph colorings in the local lemma regime. *SIAM J. Comput.*, 48(4):1397–1424, 2019.

[HH17] Bernhard Haeupler and David G. Harris. Parallel algorithms and concentration bounds for the Lovász local lemma via witness-DAGs. In *SODA*, pages 1170–1187. SIAM, 2017.

[HPR19] Tyler Helmuth, Will Perkins, and Guus Regts. Algorithmic Pirogov-Sinai theory. In *STOC*, pages 1009–1020. ACM, 2019.

[HSS11] Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. New constructive aspects of the Lovász local lemma. *J. ACM*, 58(6):28, 2011.

[HSZ19] Jonathan Hermon, Allan Sly, and Yumeng Zhang. Rapid mixing of hypergraph independent sets. *Random Struct. Algorithms*, 54(4):730–767, 2019.

[Hub15] Mark Huber. Approximation algorithms for the normalizing constant of Gibbs distributions. *Ann. Appl. Probab.*, 25(2):974–985, 2015.

[JKP19] Matthew Jenssen, Peter Keevash, and Will Perkins. Algorithms for #BIS-hard problems on expander graphs. In *SODA*, pages 2235–2247. SIAM, 2019.

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

[JVV86] Mark R. Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoret. Comput. Sci.*, 43:169–188, 1986.

[Kol18] Vladimir Kolmogorov. A faster approximation algorithm for the Gibbs partition function. In *COLT*, pages 228–249, 2018.

[LP17] David A Levin and Yuval Peres. *Markov chains and mixing times*. American Mathematical Soc., 2017.

[Moi19] Ankur Moitra. Approximate counting, the Lovász local lemma, and inference in graphical models. *J. ACM*, 66(2):10:1–10:25, 2019.

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

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and computing: randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[ŠVV09] Daniel Štefankovič, Santosh Vempala, and Eric Vigoda. Adaptive simulated annealing: A near-optimal connection between sampling and counting. *J. ACM*, 56(3):18, 2009.

[Wei06] Dror Weitz. Counting independent sets up to the tree threshold. In *STOC*, pages 140–149. ACM, 2006.

[Wig19] Avi Wigderson. *Mathematics and Computation: A Theory Revolutionizing Technology and Science*. Princeton University Press, 2019.