arXiv:2011.03915v2 [cs.DS] 10 Apr 2021

SAMPLING CONSTRAINT SATISFACTION SOLUTIONS IN THE LOCAL LEMMA REGIME

WEIMING FENG, KUN HE, AND YITONG YIN

Abstract. We give a Markov chain based algorithm for sampling almost uniform solutions of constraint satisfaction problems (CSPs). Assuming a canonical setting for the Lovász local lemma, where each constraint is violated by a small number of forbidden local configurations, our sampling algorithm is accurate in a local lemma regime, and the running time is a fixed polynomial whose dependency on n is close to linear, where n is the number of variables. Our main approach is a new technique called *state compression*, which generalizes the "mark/unmark" paradigm of Moitra [Moi19], and can give fast local-lemma-based sampling algorithms. As concrete applications of our technique, we give the current best almost-uniform samplers for hypergraph colorings and for CNF solutions.

1. Introduction

The space of constraint satisfaction solutions is one of the most well-studied subjects in Computer Science. Given a collection of constraints defined on a set of variables, a solution to the *constraint satisfaction problem* (CSP) is an assignment of variables such that all constraints are satisfied. A fundamental criterion for the existence of constraint satisfaction solutions is given by the *Lovász local lemma* (LLL) [EL75]. Interpreting the space of all assignment as a probability space and the violation of each constraint as a bad event, the local lemma characterizes a regime within which a constraint satisfaction solution always exists, by the tradeoff between: (1) the chance for the occurrence of each bad event and (2) the degree of dependency between them.

In Computer Science, the studies of the Lovász local lemma are more focused on the *algorithmic LLL* (also called *constructive LLL*), which is concerned with not just existence of a constraint satisfaction solution, but also how to find such a solution efficiently. The studies of algorithmic LLL constitute an important line of modern algorithm researches [Bec91, Alo91, MR98, CS00, Mos09, MT10, KM11, HSS11, HS17b, HS19]. A major breakthrough was the Moser-Tardos algorithm [MT10], which finds a satisfaction solution efficiently up to a sharp condition known as the Shearer's bound [She85, KM11].

In this paper, we are concerned with a problem that we call the *sampling LLL*, which asks for the regimes in which a nearly uniform (instead of an arbitrary) satisfaction solution can be generated efficiently. This is a *distribution-sensitive* variant of the algorithmic LLL. The problem is closely related to the problem of estimating the total number of satisfaction solutions, usually via standard reductions [JVV86, ŠVV09]; besides, it may also serve as a standard toolkit for solving the inference problems that are well motivated from machine learning applications [Moi19].

This sampling variant of algorithmic LLL is computationally more challenging than the conventional algorithmic LLL. For example, for k-CNF formulas with variable-degree d, the Moser-Tardos algorithm for generating an arbitrary solution is known to be efficient when $k \gtrsim \log_2 d$, while the problem of generating a nearly uniform solution requires $k \gtrsim 2\log_2 d$ to be tractable [BGG$^+$19].

Meanwhile, much less positive progress was known for the sampling LLL. A fundamental obstacle is that the space of satisfaction solutions may not be connected via local updates of variables [Wig19], whereas such connectivity is crucial for mainstream sampling techniques. In [GJL19], Guo, Jerrum and Liu proposed to study the sampling LLL, and resolved the problem for the CSPs with extremal constraints. In a major breakthrough [Moi19], Moitra introduced a novel approach for approximately

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

(Kun He) Institute of Computing Technology, Chinese Academy of Sciences. E-mail: hekun.threebody@foxmail.com.

This research is supported by the National Key R&D Program of China 2018YFB1003202 and the National Science Foundation of China under Grant Nos. 61722207 and 61672275.

counting k-SAT solutions. The approach utilizes the algorithmic LLL to properly *mark/unmark* variables, which helps construct efficient linear programmings for estimating marginal probabilities. For k-CNF formulas with variable-degree d within a local lemma regime $k \gtrsim 60 \log d$, the algorithm approximately counts the total number of SAT solutions in time $n^{\mathrm{poly}(dk)}$. Further extensions of Moitra's approach were made to hypergraph colorings [GLLZ19] and random CNF formulas [GGGY20], where the running times are both $n^{\mathrm{poly}(dk)}$ for constraint-width k and variable-degree d. Recently, a much faster algorithm for sampling k-SAT solutions inspired by Moitra's algorithm was given in [FGYZ20]. It implements a Markov chain on the assignments of the marked variables chosen via Moitra's approach. The resulting sampling algorithm enjoys a close-to-linear running time $\widetilde{O}(d^2k^3n^{1.000001})$ with an improved regime $k \gtrsim 20 \log d$. It also formally confirms that the originally disconnected solution space is changed to be very well connected after restricting onto a wisely chosen set of marked variables. However, such approach of fast sampling seems rather restricted to CNF formulas, where the variables can be marked/unmarked non-adaptively to the assignments, whereas for CSPs with larger domains where marking/unmarking variables adaptively to their assignments is crucial [GLLZ19], the current approach for fast sampling has met some fundamental barriers.

For sampling general constraint satisfaction solutions, we do not know whether the problem is tractable in a local lemma type of regime, neither do we know any general algorithmic approach that can achieve this. New ideas beyond the paradigm of marking/unmarking variables are needed.

1.1. **Our results.** We consider the problem of uniform sampling constraint satisfaction solutions, formulated by the variable-framework LLL with *uniform random variable* and *atomic bad events*. Let V be a collection of $n = |V|$ mutually independent uniform random variables and $\mathcal{B}$ be a collection of atomic bad events such that

- **uniform random variables**: the value of each $v \in V$ is uniformly drawn from a domain Q_v;
- **atomic bad events**: each $B \in \mathcal{B}$ is determined by the variables in $\mathsf{vbl}(B) \subseteq V$, and B occurs if the assignment of $\mathsf{vbl}(B)$ is as specified by the unique forbidden pattern $\sigma_B \in \bigotimes_{v\in \mathsf{vbl}(B)} Q_v$.

We assume uniform random variables because our goal is to uniformly sample constraint satisfaction solutions. Meanwhile, the atomicity of bad events is a natural and fundamental setting assumed in various studies of LLL [AI16, HH17, HS17a, Kol18a, Har21, AIS19, HS19, HV20].

Let $p = \max_{B\in\mathcal{B}} \Pr[B]$, where the probability is taken over independent random variables in V. Let $G = (\mathcal{B}, E)$ be the *dependency graph*, where each vertex is a bad event in $\mathcal{B}$, and the neighborhood of each $B \in \mathcal{B}$ in G is $\Gamma(B) \triangleq \{B' \in \mathcal{B} \setminus \{B\} \mid \mathsf{vbl}(B) \cap \mathsf{vbl}(B') \neq \emptyset\}$. Let $D \triangleq \max_{B\in\mathcal{B}} |\Gamma(B)|$ denote the maximum degree of the dependency graph. By the Lovász local lemma, there exists a satisfying assignment that avoids all bad events in $\mathcal{B}$ if

$$\ln\frac{1}{p} \geq \ln D + 1. \tag{1}$$

Such an instance of LLL naturally specifies a uniform distribution over all satisfying assignments, called the **LLL-distribution** [Har20]. Formally, it is the distribution of the independent random variables in V conditioned on that none of the bad events in $\mathcal{B}$ occurs.

Theorem 1.1. *The following holds for any $0 < \zeta \leq 2^{-400}$. There is an algorithm such that given a Lovász local lemma instance with uniform random variables and atomic bad events, if*

$$\ln\frac{1}{p} \geq 350\ln D + 3\ln\frac{1}{\zeta}, \tag{2}$$

then the algorithm outputs a random assignment $\boldsymbol{X} \in \bigotimes_{v\in V} Q_v$ in time $\widetilde{O}\left((D^2k+q)n\left(\frac{n}{\varepsilon}\right)^{\zeta}\right)$, such that the distribution of $\boldsymbol{X}$ is ε-close to the LLL-distribution in total variation distance, where $q = \max_{v\in V}|Q_v|$, $k = \max_{B\in\mathcal{B}} |\mathsf{vbl}(B)|$, and $\widetilde{O}(\cdot)$ hides a factor of $\mathrm{polylog}(n, \frac{1}{\varepsilon}, q, D)$.

This gives a unified approach for sampling uniform LLL-distributions. It is achieved by a new technique called "state compression" (see Section 1.3 and Section 3). The time complexity of the sampling algorithm is controlled by a constant parameter ζ which also controls the gap to the local lemma condition (2), so the running time can be arbitrarily close to linear in n as ζ approaches 0.

Though Theorem 1.1 is stated for uniform sampling, our main result can be extended to the LLL-distributions that arise from non-uniform random variables with arbitrary constant biases, a setting that corresponds to the statistical physics models with constant *local fields*, which are considered interesting for sampling and counting. For such a general setting, Theorem 1.1 remains to hold by replacing the condition (2) with $\ln \frac{1}{p} \geq C \ln(D/\zeta)$ where the constant factor C depends on the maximum bias. The formal proof of this general result is postpone to the full version of the paper.

On the other hand, any general non-atomic bad event can be seen as a union of disjoint atomic bad events. Let B be a bad event defined on $\mathsf{vbl}(B) \subseteq V$ and $\mathcal{N}_B \triangleq \{\sigma \in \bigotimes_{v\in\mathsf{vbl}(B)} Q_v \mid B \text{ occurs at } \sigma\}$ denote the set of assignments of $\mathsf{vbl}(B)$ that make B occur. Event B can thus be decomposed to $|\mathcal{N}_B|$ atomic events, each corresponding to a forbidden assignment $\sigma \in \mathcal{N}_B$. Therefore, any general LLL instance with $p = \max_{B\in\mathcal{B}} \Pr[B]$ and maximum degree D of the dependency graph, can be equivalently represented as an LLL instance with atomic bad events, by blowing up each bad event $B \in \mathcal{B}$ for at most $N \triangleq \max_{B\in\mathcal{B}} |\mathcal{N}_B|$ times. The resulting LLL instance with atomic bad events can be constructed within $\widetilde{O}(DNkn)$ time, such that every atomic bad event occurs with probability at most p and has the degree of dependency at most $(D+1)N$. Hence, we have the following corollary.

Corollary 1.2. *The following holds for any $0 < \zeta \leq 2^{-400}$. There is an algorithm such that given a Lovász local lemma instance with uniform random variables, if*

$$\ln \frac{1}{p} \geq 350 \ln(D+1) + 350 \ln N + 3 \ln \frac{1}{\zeta},$$

then the algorithm outputs a random assignment $\boldsymbol{X} \in \bigotimes_{v\in V} Q_v$ in time $\widetilde{O}\left((D^2N^2k + q)n\left(\frac{n}{\varepsilon}\right)^{\zeta}\right)$ such that the distribution of $\boldsymbol{X}$ is ε-close to the LLL-distribution in total variation distance, where $q = \max_{v\in V} |Q_v|$, $k = \max_{B\in\mathcal{B}} |\mathsf{vbl}(B)|$, and $\widetilde{O}(\cdot)$ hides a factor of $\mathrm{polylog}(n, \frac{1}{\varepsilon}, q, D, N)$.

To the best of our knowledge, this is the first result that achieves efficient uniform sampling of general CSP solutions within such a local lemma type of regime. In the current result, both the regime and the complexity depend on an extra parameter N, namely the maximum number of violating local configurations for any bad event. Whether such dependency is necessary is an open problem.

Our approach also produces sharper bounds for specific subclasses of LLL instances. We consider the problem of uniformly sampling proper colorings of hypergraphes. Let $H = (V, \mathcal{E})$ be a k-uniform hypergraph i.e. $|e| = k$ for all $e \in \mathcal{E}$. A proper hypergraph q-coloring $\boldsymbol{X} \in [q]^V$ assigns each vertex a color such that no hyperedge is monochromatic. Let Δ denote the maximum degree of hypergraph, i.e. each vertex belongs to at most Δ hyperedges. By LLL, a proper q-coloring exists if $q \geq C\Delta^{\frac{1}{k-1}}$ for some suitable constant C. We have the following result for sampling hypergraph colorings.

Theorem 1.3. *There is an algorithm such that given any k-uniform hypergraph on n vertices with maximum degree Δ and a set of colors $[q]$, assuming $k \geq 30$ and $q \geq 15\Delta^{\frac{9}{k-12}} + 650$, the algorithm returns a random q-coloring $\boldsymbol{X} \in [q]^V$ in time $\widetilde{O}(q^2k^3\Delta^2 n \left(\frac{n}{\varepsilon}\right)^{\frac{1}{q}})$, such that the distribution of $\boldsymbol{X}$ is ε-close in total variation distance to the uniform distribution of all proper q-colorings of the input hypergraph.*

In fact, our algorithm works for a regime where $k \geq 13$ and $q \geq q_0(k) = \Omega(\Delta^{\frac{9}{k-12}})$. See Theorem 5.4 for a more technical statement. The running time of our algorithm is always polynomially bounded for any bounded or unbounded k and Δ, and is getting arbitrarily close to linear in n as q grows.

Hypergraph colorings are important combinatorial objects. The classic local Markov chain on hypergraph colorings rapidly mixes in $O(n \log n)$ steps if $k \geq 4$ and $q > \Delta$ [BDK06, BDK08]. For "simple" hypergraphs where any two hyperedges share at most one vertex, the mixing condition was improved to $q \geq \max\{C_k \log n, 500k^3\Delta^{1/(k-1)}\}$ [FM11, FA17]. The first algorithm for sampling and counting hypergraph colorings that works in a local lemma regime was given in [GLLZ19]. The algorithm is obtained by extending Moitra's approach [Moi19] to adaptively marking/unmarking hypergraph vertices, and runs in time $n^{\mathrm{poly}(\Delta k)}$ if $k \geq 28$ and $q > 798\Delta^{\frac{16}{k-16/3}}$. Our algorithm both substantially improves the running time and improves the regime to $q \geq 15\Delta^{\frac{9}{k-12}} + O(1)$. Our algorithm utilizes a novel projection scheme instead of the mark/unmark strategy of Moitra, to transform the space of proper colorings. And our algorithm implements a rapidly mixing Markov chain on the projected space.

A canonical subclass of CSPs are the CNF (conjunctive normal form) formulas. In a k-CNF, each clause contains k distinct variables. And the maximum (variable-)degree d is given by maximum number of clauses a variable appears in. By LLL, a satisfying assignment exists if $k \geq \log d + \log k + C$[1] for some suitable constant C. We have the following result for uniform sampling k-CNF solutions.

Theorem 1.4. *The following holds for any $0 < \zeta \leq 2^{-20}$. There is an algorithm such that given any k-CNF formula on n variables with maximum degree d, assuming $k \geq 13 \log d + 13 \log k + 3 \log \frac{1}{\zeta}$, the algorithm returns a random assignment $\boldsymbol{X} \in \{\texttt{True}, \texttt{False}\}^V$ in time $\widetilde{O}(d^2 k^3 n \left(\frac{n}{\varepsilon}\right)^{\zeta})$ such that the distribution of $\boldsymbol{X}$ is ε-close in total variation distance to the uniform distribution of all satisfying assignments.*

A more detailed version is stated as Theorem 5.5. The regime $k \gtrsim 13 \log d$ in Theorem 1.4 improves the state-of-the-art regime $k \gtrsim 20 \log d$ in [FGYZ20] with the same running time.

1.2. **Implications to approximate counting.** All our sampling results imply efficient algorithms for approximate counting. Given an LLL instance Φ with uniform random variables, let Z_Φ denote the total number of satisfying assignments that avoid all bad events. For any $0 < \delta < 1$, the problem $\mathcal{P}_{\text{count}}(\Phi, \delta)$ asks to output a random number $\widehat{Z}$ such that $\widehat{Z} \in (1 \pm \delta) Z_\Phi$ with probability at least $\frac{3}{4}$.

In our results (Theorem 1.1, Corollary 1.2, Theorem 1.3, and Theorem 1.4), for several subclasses of LLL instances, we give such sampling algorithms that given an LLL instance Φ and an error bound $\varepsilon > 0$, a random $\boldsymbol{X}$ is returned in time $T(\varepsilon) = T_\Phi(\varepsilon)$ such that $\boldsymbol{X}$ is ε-close in total variation distance to the LLL-distribution of Φ, which is the uniform distribution over all satisfying assignments for Φ.

It is well known that one can solve the approximate counting problem $\mathcal{P}_{\text{count}}(\Phi, \delta)$ by calling to such oracles for nearly uniform sampling, either via the self-reducibility [JVV86] that adds one bad event at a time, or via the simulated annealing approach [BŠVV08, ŠVV09, Hub15, Kol18b] that alters a temperature. The simulated annealing gives more efficient reduction. Specifically, by routinely going through the annealing process in [FGYZ20], one can obtain a non-adaptive simulated annealing strategy to solve the approximate counting problem $\mathcal{P}_{\text{count}}(\Phi, \delta)$ in time $O\left(\frac{m}{\delta^2} T(\varepsilon) \log \frac{m}{\delta}\right)$, where $\varepsilon = \Theta\left(\frac{\delta^2}{m \log(m/\delta)}\right)$, and m denotes the number of bad events in Φ.

1.3. **Technique overview.** As addressed in [Wig19], in general, the space of SAT solutions may not be connected via local updates of variables, even when the existence of SAT solutions is guaranteed by the local lemma. A major challenge for efficiently sampling constraint satisfaction solutions in a local lemma regime is to bypass such connectivity barrier.

Several previous works that have successfully bypassed this fundamental barrier fell into the same "mark/unmark" paradigm initiated by Moitra in [Moi19]. Let V be the set of variables, and let μ denote the uniform distribution over all satisfying assignments. The paradigm effectively constructs a random pair (M, X_M) where $M \subseteq V$ is a set of marked variables and X_M is a random assignment of the marked variables in M, such that the random pair (M, X_M) satisfies the so-called "*pre-Gibbs*" property [GLLZ19], which means that if we complete X_M to an assignment $\boldsymbol{X}$ of all variables in V by sampling the complement $X_{V \setminus M}$ according to the marginal distribution induced by μ on $V \setminus M$ conditioning on X_M, then the resulting $\boldsymbol{X}$ indeed follows the correct distribution μ. The paradigm may construct the marked set M either non-adaptively to the random X_M (as in [Moi19, FGYZ20, GGGY20] for CNFs), or adaptively to it (as in [GLLZ19] for hypergraph colorings). The random pair (M, X_M) can thus be jointly distributed, so that being pre-Gibbs does not necessarily mean that X_M is distributed as the marginal distribution μ_M. Indeed, it can be much more complicated than that.

In this paper, we introduce a novel technique called "***state compression***" to bypass the connectivity barrier for general spaces of satisfaction solutions and obtain fast sampling algorithms.

For each variable $v \in V$ with domain Q_v, we construct a projection $h_v : Q_v \to \Sigma_v$ that maps from domain Q_v to an alphabet Σ_v, so that each assignment $\boldsymbol{X} \in \boldsymbol{Q} \triangleq \bigotimes_{v \in V} Q_v$ is mapped to a string $\boldsymbol{h}(\boldsymbol{X}) \triangleq (h_v(X_v))_{v \in V}$ in $\Sigma \triangleq \bigotimes_v \Sigma_v$. Therefore, the LLL-distribution μ over satisfying assignments, is transformed to a joint distribution ν over Σ as:

$$\forall \boldsymbol{Y} \in \Sigma, \quad \nu(\boldsymbol{Y}) = \Pr_{\boldsymbol{X} \sim \mu}[\boldsymbol{h}(\boldsymbol{X}) = \boldsymbol{Y}].$$

[1]Throughout the paper, we use log to denote the logarithm base 2.

Our algorithm first simulates the Glauber dynamics with stationary distribution ν to draw a sample $\boldsymbol{Y} \in \Sigma$ approximately according to ν. At each transition, the Glauber dynamics:

- picks a variable v uniformly at random;
- updates Y_v by a random value sampled according to $\nu_v^{Y_{V\setminus\{v\}}}$, which stands for the marginal distribution at v induced by ν conditioned on the assignment on $V \setminus \{v\}$ being fixed as $Y_{V\setminus\{v\}}$.

After running the Glauber dynamics for a sufficiently many $O(n \log n)$ steps, the algorithm generates a random string $\boldsymbol{Y} \in \Sigma$ which hopefully is distributed approximately as ν. Finally, the algorithm still needs to "invert" the sampled string $\boldsymbol{Y} \in \Sigma$ to a random satisfying assignment $\boldsymbol{X} \in \boldsymbol{Q}$ that follows the LLL-distribution μ conditioning on $\boldsymbol{h}(\boldsymbol{X}) = \boldsymbol{Y}$.

Both in the final step of the algorithm and at each transition of the Glauber dynamics, we are in fact trying to invert a completely specified string $\boldsymbol{Y} \in \Sigma$ (or an almost completely specified string $Y_{V\setminus\{v\}}$) to a uniform random satisfying assignment $\boldsymbol{X} \in \boldsymbol{Q}$ within its pre-image $\boldsymbol{h}^{-1}(Y)$ (or that of $Y_{V\setminus\{v\}}$).

Therefore, the efficiency of above algorithmic framework for sampling relies on that:

(1) the Glauber dynamics for ν mixes in $O(n \log n)$ steps;
(2) there is a procedure that can efficiently invert a completely (or almost completely) specified string $\boldsymbol{Y}$ to a uniform random satisfying assignment $\boldsymbol{X} \in \boldsymbol{Q}$ within the pre-image $\boldsymbol{h}^{-1}(\boldsymbol{Y})$.

As we know, the original space of satisfying assignments $\boldsymbol{X} \in \boldsymbol{Q}$ may not be connected via the local updates used by the Glauber dynamics. To achieve above item 1, intuitively, the projection $\boldsymbol{h}$ should be able to map many far-apart solutions $\boldsymbol{X}, \boldsymbol{X}' \in \boldsymbol{Q}$ to the same $\boldsymbol{h}(\boldsymbol{X}) = \boldsymbol{h}(\boldsymbol{X}')$, so the random walk in the projected space becomes well connected. This suggests that *the projection $\boldsymbol{h}$ should substantially compress the original state space*. On the other hand, the above item 2 is easier to solve when the projection $\boldsymbol{h}$ is somehow close to a one-to-one mapping, because in such case, by assuming $\boldsymbol{h}(\boldsymbol{X}) = \boldsymbol{Y}$, the original LLL instance is very likely to be decomposed into small clusters. This suggests that *the projection $\boldsymbol{h}$ should not compress the original state space too much.*

The above two seemingly contradicting requirements can in fact be captured by a set of simple and local entropy constraints, formulated in Condition 3.4. A good projection $\boldsymbol{h}$ satisfying these requirements can thus be constructed by algorithmic LLL.

The original mark/unmark paradigm can be treated as a special case of our approach of state compression. Recall that the paradigm generates a pre-Gibbs pair (M, X_M), where each variable $v \in V$ is either marked ($v \in M$) so that its value X_v is revealed, or is unmarked ($v \notin M$) so that its value X_v is unrevealed. This can be represented by a projection $\boldsymbol{h}$ where for each marked v, the projection $h_v : Q_v \to \Sigma_v$ is a one-to-one mapping to Σ_v where $|\Sigma_v| = |Q_v|$; and for each unmarked v, the projection $h_v : Q_v \to \Sigma_v$ is a all-to-one mapping to Σ_v of size $|\Sigma_v| = 1$. General projections provide a broad middle ground between the two extremal cases for the one-to-one and the all-to-one mappings, so that our technique is applicable to more general settings. And for large enough Q_v's, it indeed is such middle ground $h_v : Q_v \to \Sigma_v$ with $|\Sigma_v| \approx |Q_v|^{3/4}$ that resolves the problem well.

1.4. **Open problems.** An open problem is to remove the assumption on the atomicity of bad events. In general, the LLL is defined by arbitrary bad events on arbitrary probability space. The LLL distribution can thus be generalized. And the sampling LLL corresponds to the problems of sampling from non-uniform distributions or distributions arising from global constraints.

It is well-known that the Shearer's bound is tight for general LLL [She85]. A central open problem for sampling LLL is to find the "Shearer's bound" for sampling LLL, namely, to give a tight condition under which one can efficiently draw random samples from general LLL distributions.

Even for interesting special classes of LLL instances such as k-CNFs or hypergraph colorings, the critical thresholds for the computational phase transition for sampling are major open problems in the field of sampling algorithms.

1.5. **Organization of the paper.** Models and preliminaries are described in Section 2. The rules for state compression are given in Section 3. The main sampling algorithm is described in Section 4. In Section 5, we prove all main results in Section 1. In Section 6, we give the algorithms for constructing projections. In Section 7, we analyze the inverse sampling subroutine. The rapid mixing of the Markov chain is proved in Section 8.

2. Models and preliminaries

2.1. **CSP formulas defined by atomic bad events.** Let V be a set of variables with finite domains $(Q_v)_{v\in V}$, where each $v \in V$ takes its value from Q_v with $|Q_v| \geq 2$. Let $\mathbf{Q} \triangleq \bigotimes_{v\in V} Q_v$ denote the space for all assignments, and for any subset $\Lambda \subseteq V$, denote $Q_\Lambda \triangleq \bigotimes_{v\in\Lambda} Q_v$. Let C be a collection of *local constraints*, where each $c \in C$ is defined on a subset of variables $\mathsf{vbl}(c) \subseteq V$ that maps every assignment $\boldsymbol{x}_{\mathsf{vbl}(c)} \in Q_{\mathsf{vbl}(c)}$ to a $\texttt{True}$ or $\texttt{False}$, which indicates whether c is *satisfied* or *violated*. A CSP (constraint-satisfaction problem) formula Φ is specified by the tuple $(V, \mathbf{Q}, C)$ such that:

$$\forall \boldsymbol{x} \in \mathbf{Q}, \qquad \Phi(\boldsymbol{x}) = \bigwedge_{c\in C} c\left(\boldsymbol{x}_{\mathsf{vbl}(c)}\right),$$

where $\boldsymbol{x}_{\mathsf{vbl}(c)}$ denotes the restriction of $\boldsymbol{x}$ on $\mathsf{vbl}(c)$. In LLL's language, each $c \in C$ corresponds to a *bad event* A_c defined on $\mathsf{vbl}(c)$ that occurs if c is violated, and Φ is satisfied by $\boldsymbol{x}$ if and only if none of these bad events occurs.

In this paper, we restrict ourselves to the CSP formulas defined by atomic bad events. A constraint c defined on $\mathsf{vbl}(c)$ is called *atomic* if $|c^{-1}(\texttt{False})| = 1$, that is, if c is violated by a unique "forbidden configuration" in $Q_{\mathsf{vbl}(c)}$. Such CSP formulas with atomic constraints have drawn studies in the context of LLL [AI16, HH17, HS17a, Kol18a, Har21, AIS19, HS19, HV20]. Similar classes of CSP formulas have also been studied under the name "multi-valued/non-Boolean CNF formulas" in the field of classic Artificial Intelligence [LKM03, FP01]. Clearly, any general constraint c on $\mathsf{vbl}(c)$ can be simulated by $|c^{-1}(\texttt{False})|$ atomic constraints, each forbidding a configuration in $c^{-1}(\texttt{False})$.

The *dependency graph* of a CSP formula $\Phi = (V, \mathbf{Q}, C)$ is defined on the vertex set C, such that any two constraints $c, c' \in C$ are adjacent if $\mathsf{vbl}(c)$ and $\mathsf{vbl}(c')$ intersect. We use $\Gamma(c) \triangleq \{c' \in C \setminus \{c\} \mid \mathsf{vbl}(c) \cap \mathsf{vbl}(c') \neq \emptyset\}$ to denote the neighborhood of $c \in C$ and let

$$D = D_\Phi \triangleq \max_{c\in C} |\Gamma(c)|$$

denote the maximum degree of the dependency graph.

The followings are some typical special cases of CSP formulas with atomic constraints.

2.1.1. *k-CNF formula.* The CNF formulas $\Phi = (V, \mathbf{Q}, C)$ are formulas with atomic constraints on Boolean domains $Q_v = \{\texttt{True}, \texttt{False}\}$, for all $v \in V$. Now each constraint $c \in C$ is a *clause*. For k-CNF formulas, we have $|\mathsf{vbl}(c)| = k$ for all clauses $c \in C$.

2.1.2. *Hypergraph coloring.* Let $H = (V, \mathcal{E})$ be a k-uniform hypergraph, where every hyperedge $e \in \mathcal{E}$ has $|e| = k$. Let $[q] = \{1, 2, \ldots, q\}$ be a set of q colors. A proper hypergraph coloring $\boldsymbol{X} \in [q]^V$ assigns each vertex $v \in V$ a color X_v such that no hyperedge is monochromatic.

Define the following set C of atomic constraints. For each hyperedge $e \in \mathcal{E}$ and color $i \in [q]$, add an atomic constraint $c_{e,i}$ into C, where $c_{e,i}$ is defined as $\mathsf{vbl}\left(c_{e,i}\right) = e$ and for any $\boldsymbol{x} \in [q]^e$, $c_{e,i}(\boldsymbol{x}) = \texttt{False}$ if and only if $x_v = i$ for all $v \in e$. It is straightforward to see that there is a one-to-one correspondence between the proper q-colorings in H and the satisfying assignments to $\Phi = (V, [q]^V, C)$.

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 $\Gamma(B) \triangleq \{B' \in \mathcal{B} \mid B' \neq B \text{ and } \mathsf{vbl}(B') \cap \mathsf{vbl}(B) \neq \emptyset\}$. For any event $A \notin \mathcal{B}$ and its determining variables $\mathsf{vbl}(A) \subseteq \mathcal{R}$, we define $\Gamma(A) \triangleq \{B \in \mathcal{B} \mid \mathsf{vbl}(A) \cap \mathsf{vbl}(B) \neq \emptyset\}$. Let $\Pr_{\mathcal{D}}[\cdot]$ denote the product distribution of variables in $\mathcal{R}$. The following version of the Lovász local lemma will be used in this paper.

Theorem 2.1 ([HSS11]). *If there is a function $x : \mathcal{B} \to (0, 1)$ such that for any $B \in \mathcal{B}$,*

$$\Pr_{\mathcal{D}}[B] \leq x(B) \prod_{B'\in\Gamma(B)} (1 - x(B')), \tag{3}$$

then it holds that

$$\Pr_{\mathcal{D}}\left[\bigwedge_{B\in\mathcal{B}} \overline{B}\right] \geq \prod_{B\in\mathcal{B}} (1-x(B)) > 0.$$

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

Moreover, for any event A, it holds that

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

2.3. **Coupling, Markov chain and mixing time.** Let Ω be a state space. Let μ and ν be two distributions over Ω. The *total variation distance* between μ and ν are defined by

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

A coupling of μ and ν is a joint distribution $(X,Y)\in\Omega\times\Omega$ such that the marginal distribution of X is μ and the marginal distribution of Y is ν. The following coupling lemma is well-known.

Lemma 2.2 (coupling lemma [LP17, Proposition 4.7])**.** *For any coupling (X,Y) between μ and ν,*

$$d_{\mathrm{TV}}(\mu,\nu) \leq \Pr[X\neq Y].$$

Moreover, there exists an optimal coupling *that achieves the equality.*

A *Markov chain* is a random sequence $(X_t)_{t\geq 0}$ over a state space Ω such that the transition rule is specified by the *transition matrix* $P:\Omega\times\Omega\to\mathbb{R}_{\geq 0}$. We often use the transition matrix to denote the corresponding Markov chain. The Markov chain P is *irreducible* if for any $X,Y\in\Omega$, there exists $t>0$ such that $P^t(X,Y)>0$. The Markov chain P is *aperiodic* if $\gcd\{t \mid P^t(X,X)>0\}=1$ for all $X\in\Omega$. A distribution π over Ω is a *stationary distribution* of P if $\pi P=\pi$. If a Markov chain is irreducible and aperiodic, then it has a unique stationary distribution. The Markov chain P is *reversible* with respect to the distribution π if the following *detailed balance equation* holds

$$\forall X,Y\in\Omega: \quad \pi(X)P(X,Y)=\pi(Y)P(Y,X),$$

which implies π is a stationary distribution of P. Given a Markov chain P with the unique stationary distribution π, the *mixing time* of P is defined by

$$\forall 0<\varepsilon<1, \quad T_{\mathsf{mix}}(\varepsilon) \triangleq \max_{X_0\in\Omega}\min\{t \mid d_{\mathrm{TV}}\left(P^t(X_0,\cdot),\pi\right)\leq\varepsilon\}.$$

A coupling of Markov chain P is a joint random process $(X_t,Y_t)_{t\geq 0}$ such that both $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$ follow the transition rule of P individually, and if $X_s=Y_s$, then $X_k=Y_k$ for all $k\geq s$. The coupling is a widely-used tool to bound the mixing times of Markov chains, because by the coupling lemma, it holds that $\max_{X_0\in\Omega} d_{\mathrm{TV}}\left(P^t(X_0,\cdot),\pi\right)\leq \max_{X_0,Y_0\in\Omega}\Pr[X_t\neq Y_t]$.

The *path coupling* [BD97] is a powerful tool to construct the coupling of Markov chains. Assume $\Omega=\bigotimes_{v\in V} Q_v$, where $|V|=n$ and each Q_v is a finite domain. For any $X,Y\in\Omega$, define the *Hamming distance* between X and Y by

$$d_{\mathrm{ham}}(X,Y)\triangleq |\{v\in V \mid X_v\neq Y_v\}|.$$

In this paper, we will use the following simplified version of path coupling.

Lemma 2.3 (path coupling [BD97])**.** *Let $0<\delta<1$ be a parameter. Let P be an irreducible and aperiodic Markov chain over the state space $\Omega=\bigotimes_{v\in V} Q_v$, where $|V|=n$. If there is a coupling of Markov chain $(X,Y)\to(X',Y')$ defined over all $X,Y\in\Omega$ with $d_{\mathrm{ham}}(X,Y)=1$ such that*

$$\mathrm{E}[d_{\mathrm{ham}}(X',Y') \mid X,Y]\leq 1-\delta,$$

then the mixing time of the Markov chain satisfies

$$T_{mix}(\varepsilon)\leq\left\lceil\frac{1}{\delta}\log\frac{n}{\varepsilon}\right\rceil.$$

Readers can refer to the textbook [LP17] for more backgrounds of Markov chains and mixing times.

A CSP formula $\Phi = (V, \boldsymbol{Q}, C)$ with uniformly distributed random variables defines an LLL instance.

Definition 3.1 (LLL-distribution)**.** For each $v \in V$, let π_v denote the uniform distribution over domain Q_v. Let $\pi \triangleq \bigotimes_{v\in V} \pi_v$ be the uniform distribution over $\boldsymbol{Q}$. Let $\mu = \mu_\Phi$ denote the distribution of $\boldsymbol{X} \sim \pi$ conditioned on $\Phi(\boldsymbol{X})$, that is, the uniform distribution over satisfying solutions of Φ.

This distribution μ over satisfying solutions of Φ is what we want to sample from. In order to do so, this uniform probability space of satisfying solutions is transformed by a projection. A *projection scheme* $\boldsymbol{h} = (h_v)_{v\in V}$ specifies for each $v \in V$, a mapping from v's domain Q_v to a finite *alphabet* Σ_v:

$$h_v : Q_v \to \Sigma_v.$$

Let $\Sigma \triangleq \bigotimes_{v\in V} \Sigma_v$, and for any $\Lambda \subseteq V$, we denote $\Sigma_\Lambda \triangleq \bigotimes_{v\in\Lambda} \Sigma_v$.

We also naturally interpret $\boldsymbol{h}$ as a function on (partial) assignments such that

$$\forall \Lambda \subseteq V, \forall \boldsymbol{x} \in Q_\Lambda, \qquad \boldsymbol{h}(\boldsymbol{x}) \triangleq (h_v(x_v))_{v\in\Lambda}. \tag{4}$$

Definition 3.2 (projected LLL-distribution)**.** For each $v \in V$, let ρ_v be the distribution of $Y_v = h_v(X_v)$ where $X_v \sim \pi_v$. Let $\rho \triangleq \bigotimes_{v\in V} \rho_v$ be the product distribution over Σ.

For each $v \in V$ and any $y_v \in \Sigma_v$, let $\pi_v^{y_v}$ denote the distribution of $X_v \sim \pi_v$ conditioned on $h_v(X_v) = y_v$. For any $\Lambda \subseteq V$ and $y_\Lambda \in \Sigma_\Lambda$, let π^{y_Λ} be the distribution of $\boldsymbol{X} \sim \pi$ conditioned on $\boldsymbol{h}(\boldsymbol{X}_\Lambda) = y_\Lambda$.

Let $\nu = \nu_{\Phi,\boldsymbol{h}}$ denote the distribution of $\boldsymbol{Y} = \boldsymbol{h}(\boldsymbol{X})$ where $\boldsymbol{X} \sim \mu$.

Note that the original LLL-distribution μ is a *Gibbs* distribution [MM09], defined by local constraints on independent random variables. Whereas, the distribution ν of projected satisfying solution, is a joint distribution over Σ, which may no longer be a Gibbs distribution nor can it be represented as any LLL instance, because $\boldsymbol{x}, \boldsymbol{x}' \in Q$ with $\Phi(\boldsymbol{x}) \neq \Phi(\boldsymbol{x}')$ may be mapped to the same $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}')$.

In the algorithm, a projection scheme $\boldsymbol{h} = (h_v)_{v\in V}$ is accessed through the following oracle.

Definition 3.3 (projection oracle)**.** A *projection oracle* with query cost t for a projection scheme $\boldsymbol{h} = (h_v)_{v\in V}$ is a data structure that can answer each of the following two types of queries within time t:

- *evaluation*: given an input value $x_v \in Q_v$ of a variable $v \in V$, output $h_v(x_v) \in \Sigma_v$;
- *inversion*: given a projected value $y_v \in \Sigma_v$ of a variable $v \in V$, return a random $X_v \sim \pi_v^{y_v}$.

Our algorithm for sampling a uniform random satisfying solution is then outlined below.

Algorithm for sampling from μ

1. Construct a good projection scheme $\boldsymbol{h}$ (formalized by Condition 3.4);
2. sample a uniform random $\boldsymbol{X} \sim \pi$ and let $\boldsymbol{Y} = \boldsymbol{h}(\boldsymbol{X})$;
3. (Glauber dynamics on ν) repeat the followings for sufficiently many iterations:

   pick a $v \in V$ uniformly at random;

   update Y_v by redrawing its value independently according to $\nu_v^{Y_{V\setminus\{v\}}}$;
4. sample $\boldsymbol{X} \sim \mu$ conditioned on $\boldsymbol{h}(\boldsymbol{X}) = \boldsymbol{Y}$.

The algorithm simulates a Markov chain (known as the Glauber dynamics) on space Σ for drawing a random configuration $\boldsymbol{Y} \in \Sigma$ approximately according to the joint distribution ν, after which, the algorithm "inverts" $\boldsymbol{Y}$ to a uniform random satisfying assignment $\boldsymbol{X}$ for Φ within the pre-image $\boldsymbol{h}^{-1}(\boldsymbol{Y})$.

The key to the effectiveness of this sampling algorithm is that we should be able to sample accurately and efficiently from $\nu_v^{Y_{V\setminus\{v\}}}$ (which is the marginal distribution at v induced by ν conditioning on that the configuration on $V \setminus \{v\}$ being fixed as $Y_{V\setminus\{v\}}$) as well as from $\mu^{\boldsymbol{Y}}$ (which is the distribution of $\boldsymbol{X} \sim \mu$ conditioned on that $\boldsymbol{h}(\boldsymbol{X}) = \boldsymbol{Y}$). In fact, both of these are realized by sampling generally from the following marginal distribution $\mu_S^{y_\Lambda}$, for $S \subseteq V$ and $y_\Lambda \in \Sigma_\Lambda$, where either $\Lambda = V$ or $|\Lambda| = |V| - 1$.

$$\mu_S^{y_\Lambda} : \text{ distribution of } X_S, \text{ where } \boldsymbol{X} \in \boldsymbol{Q} \text{ is drawn from } \mu \text{ conditioning on that } \boldsymbol{h}(X_\Lambda) = y_\Lambda. \tag{5}$$

The distribution μ^Y corresponds to the special case of $\mu_S^{y_\Lambda}$ with $S = \Lambda = V$. And also we can sample from $\nu_v^{Y_{V\setminus\{v\}}}$ by first sampling a $X_v \sim \mu_v^{Y_{V\setminus\{v\}}} \triangleq \mu_{\{v\}}^{Y_{V\setminus\{v\}}}$ and then outputting $h_v(X_v)$.

Since y_Λ is either completely or almost completely specified on V, sampling from $\mu_S^{y_\Lambda}$ is essentially trying to invert y_Λ according to distribution μ. And this task becomes tractable when the projection $\boldsymbol{h}$ is somehow close to a 1-1 mapping, i.e. when $\boldsymbol{h}(\boldsymbol{X})$'s entropy remains significant compared to $\boldsymbol{X} \sim \mu$.

On the other hand, the efficiency of the sampling algorithm relies on the mixing of the Markov chain for sampling from ν. It was known that the original state space of all satisfying solutions might not be well connected through single-site updates [Wig19, FGYZ20]. The projection may increase the connectivity of the state space by mapping many far-apart satisfying solutions to the same configuration in Σ, but this means that the projection $\boldsymbol{h}$ should not be too close to a 1-1 mapping. In other words, the projection $\boldsymbol{h}(\boldsymbol{X})$ shall reduce the entropy of $\boldsymbol{X} \sim \mu$ by a substantial amount.

These two seemingly contradicting requirements are formally captured by the following condition.

Condition 3.4 (entropy criterion)**.** *Let $0 < \beta < \alpha < 1$ be two parameters.The followings hold for the CSP formula $\Phi = (V, \boldsymbol{Q}, C)$ and the projection scheme $\boldsymbol{h}$. For each $v \in V$, let $q_v \triangleq |Q_v|$ and $s_v \triangleq |\Sigma_v|$. The projection $\boldsymbol{h}$ is* balanced, *which means for any $v \in V$ and $y_v \in \Sigma_v$,*

$$\left\lfloor \frac{q_v}{s_v} \right\rfloor \le \left|h_v^{-1}(y_v)\right| \le \left\lceil \frac{q_v}{s_v} \right\rceil.$$

And for any constraint $c \in C$, it holds that

$$\sum_{v \in \mathsf{vbl}(c)} \log \left\lceil \frac{q_v}{s_v} \right\rceil \le \alpha \sum_{v \in \mathsf{vbl}(c)} \log q_v, \tag{6}$$

$$\sum_{v \in \mathsf{vbl}(c)} \log \left\lfloor \frac{q_v}{s_v} \right\rfloor \ge \beta \sum_{v \in \mathsf{vbl}(c)} \log q_v. \tag{7}$$

Note that for uniform random variable $X_v \in Q_v$, the entropy $H(X_v) = \log q_v$, and for $Y_v = h_v(X_v)$ where $\boldsymbol{h}$ is balanced, we have $\log \frac{q_v}{\lceil q_v/s_v \rceil} \le H(Y_v) \le \log \frac{q_v}{\lfloor q_v/s_v \rfloor}$. Therefore, the two inequalities (6) and (7) are in fact slightly stronger versions of the entropy upper and lower bounds for $\boldsymbol{X} \sim \pi$:

$$(1-\alpha) \sum_{v \in \mathsf{vbl}(c)} H(X_v) \le \sum_{v \in \mathsf{vbl}(c)} H(h_v(X_v)) \le (1-\beta) \sum_{v \in \mathsf{vbl}(c)} H(X_v).$$

So how may such a projection satisfying Condition 3.4 change the properties of a solution space and help sampling? Next, we introduce two consequent conditions of Condition 3.4 to explain this.

Recall that after projection, the joint distribution ν over projected solutions may no longer be represented by any LLL instance. Nevertheless, we can modify it to a valid LLL instance by proper rounding.

Definition 3.5 (the "round-down" CSP formula)**.** Given a CSP formula $\Phi = (V, \boldsymbol{Q}, C)$ and a projection scheme $\boldsymbol{h} = (h_v)_{v \in V}$, let CSP formula $\Phi^{\lfloor \boldsymbol{h} \rfloor} = (V, \Sigma, C^{\lfloor \boldsymbol{h} \rfloor})$ be constructed as follows:

- the variable set is still V and each variable $v \in V$ now takes values from Σ_v;
- corresponding to each constraint $c \in C$ of Φ, a constraint $c' \in C^{\lfloor \boldsymbol{h} \rfloor}$ is constructed as follows:

$$\mathsf{vbl}(c') = \mathsf{vbl}(c) \quad \text{and}$$

$$\forall \boldsymbol{y} \in \Sigma_{\mathsf{vbl}(c')}, \quad c'(\boldsymbol{y}) = \begin{cases} \texttt{True} & \text{if } c(\boldsymbol{x}) \text{ for all } \boldsymbol{x} \in \Omega_{\mathsf{vbl}(c)} \text{ that } \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{y}, \\ \texttt{False} & \text{if } \neg c(\boldsymbol{x}) \text{ for some } \boldsymbol{x} \in \Omega_{\mathsf{vbl}(c)} \text{ that } \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{y}. \end{cases}$$

The CSP formula $\Phi^{\lfloor \boldsymbol{h} \rfloor}$ is considered a "*round-down*" version of the CSP formula Φ under projection $\boldsymbol{h}$, because it always holds that $c'(\boldsymbol{y}) = \left\lfloor \Pr_{\boldsymbol{X} \sim \pi}\left[c\left(X_{\mathsf{vbl}(c)}\right) \mid \boldsymbol{h}\left(X_{\mathsf{vbl}(c)}\right) = \boldsymbol{y}\right] \right\rfloor$ for all $\boldsymbol{y} \in \Sigma_{\mathsf{vbl}(c')} = \Sigma_{\mathsf{vbl}(c)}$.

Recall that the following "LLL condition" is assumed for the LLL instance defined by CSP formula $\Phi = (V, \boldsymbol{Q}, C)$ on uniform random variables $\boldsymbol{X} \sim \pi$:

$$\ln \frac{1}{p} > A \ln D + B, \quad \text{(for some suitable constants } A \text{ and } B) \tag{8}$$

where $p \triangleq \max_{c \in C} \Pr_{X \sim \pi}\left[\neg c\left(X_{\mathsf{vbl}(c)}\right)\right]$ denotes the maximum probability that a constraint $c \in C$ is violated and D denotes the maximum degree of the dependency graph.

For CSP formula Φ defined by atomic constraints, the LLL condition (8) and the inequality (6) in Condition 3.4 together imply the following condition.

Condition 3.6 (round-down LLL criterion)**.** *The LLL instance defined by the round-down CSP formula $\Phi^{\lfloor \boldsymbol{h} \rfloor} = (V, \Sigma, C^{\lfloor \boldsymbol{h} \rfloor})$ on variables distributed as ρ, satisfies that*

$$\ln \frac{1}{p} > (1-\alpha)(A \ln D + B),$$

where $p \triangleq \max_{c \in C^{\lfloor \boldsymbol{h} \rfloor}} \Pr_{Y \sim \rho}\left[\neg c\left(Y_{\mathsf{vbl}(c)}\right)\right]$ and D denotes the maximum degree of the dependency graph.

The projection $\boldsymbol{h}$ may map both satisfying $\boldsymbol{x} \in \boldsymbol{Q}$ and unsatisfying $\boldsymbol{x}' \in \boldsymbol{Q}$ to the same $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}') \in \Sigma$, which causes ambiguity for classifying those "satisfying" $\boldsymbol{y} \in \Sigma$. The round-down CSP formula resolves such ambiguity with a pessimistic mindset: it refutes any $\boldsymbol{y} \in \Sigma$ whenever even a single $\boldsymbol{x} \in \boldsymbol{h}^{-1}(\boldsymbol{y})$ is unsatisfying. Condition 3.6 basically says that an LLL condition holds even up to such a pessimistic interpretation. This is crucial for sampling from $\mu_S^{y_\Lambda}$ defined in (5), because within such regime, the probability space of μ^{y_Λ} is decomposed into small clusters of sizes $O(\log n)$.

Meanwhile, the LLL condition (8) and the inequality (7) in Condition 3.4 together imply the following condition.

Condition 3.7 (conditional LLL criterion)**.** *For any $\Lambda \subseteq V$ and $y_\Lambda \in \Sigma_\Lambda$, the LLL instance defined by CSP formula $\Phi = (V, \boldsymbol{Q}, C)$ on variables distributed as π^{y_Λ}, satisfies that*

$$\ln \frac{1}{p} > \beta(A \ln D + B),$$

where $p \triangleq \max_{c \in C} \Pr_{X \sim \pi^{y_\Lambda}}\left[\neg c\left(X_{\mathsf{vbl}(c)}\right)\right]$ and D denotes the maximum degree of the dependency graph.

Condition 3.7 is basically a self-reducibility property. A major obstacle for sampling satisfying solution is that the regime (8) for the original CSP formula Φ may not be self-reducible: it is not closed under pinning of variables to arbitrary evaluations. Condition 3.7 states that the self-reducibility property is achieved under projection: the LLL regime is closed under pinning of variables to arbitrary projected evaluations. This is crucial for rapid mixing of the Markov chain on projected space Σ.

We have efficient procedures for constructing the projection scheme satisfying Condition 3.4.

Theorem 3.8 (projection construction)**.** *Let $0 < \beta < \alpha < 1$ be two parameters. Let $\Phi = (V, \boldsymbol{Q}, C)$ be a CSP formula where all constraints in C are atomic. Let D denotes the maximum degree of its dependency graph and $p \triangleq \max_{c \in C} \prod_{v \in \mathsf{vbl}(c)} \frac{1}{|Q_v|}$. If $\log \frac{1}{p} \geq \frac{25}{(\alpha-\beta)^3}(\log D + 3)$, then for any $0 < \delta < 1$, with probability at least $1 - \delta$ a projection oracle (Definition 3.3) with query cost $O(\log q)$ can be successfully constructed within time $O(n(Dk+q)\log\frac{1}{\delta}\log q)$, where $q \triangleq \max_{v \in V}|Q_v|$, $k \triangleq \max_{c \in C}|\mathsf{vbl}(c)|$ and the oracle is for a projection scheme $\boldsymbol{h} = (h_v)_{v \in V}$ that satisfies Condition 3.4 with parameters (α, β).*

The above result can be strengthened for the (k,d)*-CSP formulas*, where $|\mathsf{vbl}(c)| = k$ for all $c \in C$ and each $v \in V$ appears in at most d constraints, on homogeneous domains $Q_v = [q]$ for all $v \in V$.

Theorem 3.9. *Let $0 < \beta < \alpha < 1$ be two parameters. The followings hold for any (k,d)-CSP formula $\Phi = (V, [q]^V, C)$ where all constraints in C are atomic:*

- *If $7 \leq q^{\frac{\alpha+\beta}{2}} \leq \frac{q}{6}$ and $\log q \geq \frac{1}{\alpha-\beta}$, then a projection oracle with query cost $O(\log q)$ for a projection scheme $\boldsymbol{h}$ satisfying Condition 3.4 with parameters (α,β), can be constructed in time $O(n \log q)$.*
- *If $k \geq \frac{2\ln 2}{(\alpha-\beta)^2}\log(2ekd)$, then for any $0 < \delta < 1$, with probability at least $1-\delta$ a projection oracle as above can be successfully constructed within time $O(ndk\log\frac{1}{\delta})$.*

The proofs of Theorem 3.8 and Theorem 3.9 are given in Section 6.

4. The sampling algorithm

Let $\Phi = (V, \mathcal{Q}, \mathcal{C})$ be the input CSP formula with atomic constraints, which defines a uniform distribution μ over satisfying assignments as in Definition 3.1. Let $\varepsilon > 0$ be an error bound. The goal is to output a random assignment $\boldsymbol{X} \in \mathcal{Q}$ such that $d_{\mathrm{TV}}(\boldsymbol{X}, \mu) \leq \varepsilon$.

Depending on the classes of CSP formulas, the algorithm first applies one of the procedures in Theorem 3.8 and Theorem 3.9 to construct a projection scheme $\boldsymbol{h} = (h_v)_{v \in V}$, where $h_v : Q_v \to \Sigma_v$ for each $v \in V$, such that $\boldsymbol{h}$ satisfies Condition 3.4 with parameters (α, β), where $0 < \beta < \alpha < 1$ are going to be fixed later in the analysis in Section 5. For randomized construction procedure, we set its failure probability to be $\frac{\varepsilon}{4}$, and if it fails, the sampling algorithm simply returns an arbitrary $\boldsymbol{X} \in \mathcal{Q}$.

Suppose that the projection scheme $\boldsymbol{h}$ is given. The sampling algorithm is described in Algorithm 1.

Algorithm 1: The sampling algorithm (given a proper projection scheme)

input : a CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$ with atomic constraints, a projection scheme $\boldsymbol{h} = (h_v)_{v \in V}$ satisfying Condition 3.4 with parameters (α, β), and an error bound $\varepsilon > 0$;

output: a random assignment $\boldsymbol{X} \in \mathcal{Q}$;

sample a uniform random $\boldsymbol{X} \sim \pi$ and let $Y \leftarrow \boldsymbol{h}(\boldsymbol{X})$;

for *each* t *from* 1 *to* $T \triangleq \lceil 2n \log \frac{4n}{\varepsilon} \rceil$ **do** // `Glauber dynamics for` $Y \in \Sigma$

  pick a variable $v \in V$ uniformly at random;

  $X_v \leftarrow \mathsf{InvSample}\left(\Phi, \boldsymbol{h}, \frac{\varepsilon}{4(T+1)}, Y_{V \setminus \{v\}}, \{v\}\right)$; // `sample` $X_v \in Q_v$ `approx. from` $\mu_v^{Y_{V \setminus \{v\}}}$

  $Y_v \leftarrow h_v(X_v)$;

$\boldsymbol{X} \leftarrow \mathsf{InvSample}\left(\Phi, \boldsymbol{h}, \frac{\varepsilon}{4(T+1)}, Y, V\right)$; // `sample` $\boldsymbol{X} \in \mathcal{Q}$ `approx. from` μ^Y

return $\boldsymbol{X}$;

Algorithm 1 implements the sampling algorithm outlined in Section 3. It first implements the Glauber dynamics on space Σ for sampling from ν, the distribution of projected satisfying assignments in Definition 3.2. It simulates the Glauber dynamics for $T = \lceil 2n \log \frac{4n}{\varepsilon} \rceil$ steps to draw a random $Y \in \Sigma$ distributed approximately as ν. At each step, Y_v for a uniformly picked $v \in V$ is redrawn approximately from the marginal distribution $\nu_v^{Y_{V \setminus \{v\}}}$. At last, the algorithm inverts the sampled $Y \in \Sigma$ to a random satisfying assignment $\boldsymbol{X} \in \mathcal{Q}$ distributed approximately as μ conditioning on that $\boldsymbol{h}(\boldsymbol{X}) = Y$.

Algorithm 1 relies on an *Inverse Sampling* subroutine for sampling approximately from $\mu_v^{Y_{V \setminus \{v\}}}$ or μ^Y.

4.1. The InvSample subroutine (Algorithm 2).

The goal of the subroutine $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, y_\Lambda, S)$, where $S \subseteq V$, $\Lambda \subseteq V$, and $y_\Lambda \in \Sigma_\Lambda$, is to sample a random $X_S \in Q_S$ according to the distribution $\mu_S^{y_\Lambda}$, as defined in (5). In principle, computing the distribution $\mu_S^{y_\Lambda}$ involves computing some nontrivial partition function, which is intractable in general. Here, for an error bound $\delta > 0$, we only ask for that with probability at least $1 - \delta$, the subroutine returns a random sample that is δ-close to $\mu_S^{y_\Lambda}$ in total variation distance, where the probability is taken over the randomness of the input y_Λ.

We define some notions to describe the subroutine. Let $c \in \mathcal{C}$ be a constraint in CSP formula Φ. Recall that c is atomic. Let

$$F^c \triangleq c^{-1}(\texttt{False})$$

denote the unique "forbidden configuration" in $\mathcal{Q}_{\mathsf{vbl}(c)}$ that violates c. We say that an atomic constraint $c \in \mathcal{C}$ is *satisfied* by $y_\Lambda \in \Sigma_\Lambda$ for $\Lambda \subseteq V$, if

$$\boldsymbol{h}\left(F^c_{\Lambda \cap \mathsf{vbl}(c)}\right) \neq y_{\Lambda \cap \mathsf{vbl}(c)}, \tag{9}$$

where the function $\boldsymbol{h}(\cdot)$ is formally defined in (4). For atomic constraint $c \in \mathcal{C}$, the above condition (9) implies that c is satisfied by any $\boldsymbol{x} \in \mathcal{Q}$ that $\boldsymbol{h}(x_\Lambda) = y_\Lambda$. Hence, the constraint c must be satisfied by any configuration in the support of the distribution $\mu^{y_\Lambda} = \mu_V^{y_\Lambda}$.

The key idea of the subroutine is that we can remove all the constraints that have already been satisfied by y_Λ to obtain a new CSP formula $\Phi' = (V, \mathcal{Q}, \mathcal{C}')$, where $\mathcal{C}' \triangleq \{c \in \mathcal{C} \mid c \text{ is not satisfied by } y_\Lambda\}$.

Algorithm 2: $\textsf{InvSample}\,(\Phi, \boldsymbol{h}, \delta, y_\Lambda, S)$

Input : a CSP formula $\Phi = (V, \boldsymbol{Q}, C)$ with atomic constraints, a projection scheme $\boldsymbol{h}$, an error bound $\delta > 0$, a configuration $y_\Lambda \in \Sigma_\Lambda$ specified on $\Lambda \subseteq V$, and a subset $S \subseteq V$;

Output : a random assignment $\boldsymbol{X} \in \boldsymbol{Q}_S$;

let Φ' be the new formula obtained by removing all the constraints in Φ already satisfied by y_Λ;

factorize Φ' and find all the sub-formulas $\left\{\Phi'_i = (V_i, \boldsymbol{Q}_{V_i}, C'_i) \mid 1 \le i \le \ell\right\}$ s.t. each $V_i \cap S \neq \emptyset$;

if *there exists* $1 \le i \le \ell$ *s.t.* $|C'_i| > 2D \log \frac{nD}{\delta}$ **then** // existence of giant component

  return *a uniform random* $\boldsymbol{X}_S \sim \pi_S$;

for *each i from 1 to* ℓ **do**

  repeat *for at most* $R \triangleq \left\lceil 10 \left(\frac{n}{\delta}\right)^\eta \log \frac{n}{\delta} \right\rceil$ *times:* // rejection sampling with $\le R$ trials

    sample $\boldsymbol{X}_i \sim \pi_{V_i}^{y_{\Lambda_i}}$, where $\Lambda_i \triangleq V_i \cap \Lambda$;

  until $\Phi'_i(\boldsymbol{X}_i) = \texttt{True}$;

  if $\Phi'_i(\boldsymbol{X}_i) = \texttt{False}$ **then** // overflow of rejection sampling

    return *a uniform random* $\boldsymbol{X}_S \sim \pi_S$;

return $\boldsymbol{X}'_S$, *where* $\boldsymbol{X}' = \bigcup_{i=1}^{\ell} \boldsymbol{X}_i$;

Define $\mu_{\Phi'}^{y_\Lambda}$ to be the distribution of $\boldsymbol{X} \sim \pi^{y_\Lambda}$ conditioned on $\Phi'(\boldsymbol{X})$, where the product distribution π^{y_Λ} is as in Definition 3.2. It is straightforward to verify that $\mu_{\Phi'}^{y_\Lambda} \equiv \mu^{y_\Lambda}$.

Furthermore, the new CSP formula Φ' can be *factorized* into a set of disjoint formulas:

$$\Phi' = \Phi'_1 \wedge \Phi'_2 \wedge \ldots \wedge \Phi'_m.$$

Our plan is to show that it almost always holds that the size of every sub-formula Φ'_i is logarithmically bounded. Thus, we can apply the naïve rejection sampling independently on each sub-formula Φ'_i, which remains to be efficient altogether.

Formally, let $H' = (V, \mathcal{E}')$ denote the (multi-)hypergraph induced by the CSP formula $\Phi' = (V, \boldsymbol{Q}, C')$, constructed by adding a hyperedge $e_c = \textsf{vbl}\,(c)$ into $\mathcal{E}'$ for each constraint $c \in C'$. Note that H' may contain duplicated hyperedges. Let $H'_1, H'_2, \ldots, H'_m$ denote the connected components of H', where $H'_i = (V_i, \mathcal{E}'_i)$. Let $\Phi'_i = (V_i, \boldsymbol{Q}_{V_i}, C'_i)$ denote sub-formula corresponding to H'_i, where C'_i is the set of constraints corresponding to hyperedges in $\mathcal{E}'_i$. This defines the *factorization* $\Phi' = \Phi'_1 \wedge \Phi'_2 \wedge \ldots \wedge \Phi'_m$. For each sub-formula $\Phi'_i = (V_i, \boldsymbol{Q}_{V_i}, C'_i)$, let $\Lambda_i = \Lambda \cap V_i$, and define $\mu_{\Phi'_i}^{y_{\Lambda_i}}$ to be the distribution of $\boldsymbol{X} \sim \pi_{V_i}^{y_{\Lambda_i}}$ conditioned on $\Phi'_i(\boldsymbol{X})$, where $\pi_{V_i}^{y_{\Lambda_i}}$ denotes restriction of the product distribution $\pi^{y_{\Lambda_i}}$ on V_i. It is then straightforward to verify:

$$\mu^{y_\Lambda} \equiv \mu_{\Phi'}^{y_\Lambda} \equiv \mu_{\Phi'_1}^{y_{\Lambda_1}} \times \mu_{\Phi'_2}^{y_{\Lambda_2}} \times \ldots \times \mu_{\Phi'_m}^{y_{\Lambda_m}}.$$

Without loss of generality, we assume $S \cap V_i \neq \emptyset$ for $1 \le i \le \ell$ and $S \cap V_i = \emptyset$ for $\ell < i \le m$. It suffices to draw random samples $\boldsymbol{X}_i \sim \mu_{\Phi'_i}^{y_{\Lambda_i}}$ independently for all $1 \le i \le \ell$, adjoin them together $\boldsymbol{X}' = \cup_{i=1}^{\ell} \boldsymbol{X}_i$, and output its restriction $\boldsymbol{X}'_S$ on S, where each $\boldsymbol{X}_i \sim \mu_{\Phi'_i}^{y_{\Lambda_i}}$ can be drawn by the *rejection sampling* procedure: repeatedly and independently sampling $\boldsymbol{X}_i \sim \pi_{V_i}^{y_{\Lambda_i}}$ until $\Phi'_i(\boldsymbol{X}_i)$ is true.

The subroutine $\textsf{InvSample}\,(\Phi, \boldsymbol{h}, \delta, y_\Lambda, S)$ does precisely as above with two exceptions:

- **existence of giant connected component**: $|C'_i| \ge 2D \log \frac{nD}{\delta}$ for some $1 \le i \le \ell$, where D stands for the maximum degree of the dependency graph for Φ;
- **overflow of rejection sampling**: the rejection sampling from $\mu_{\Phi'_i}^{y_{\Lambda_i}}$ for some $1 \le i \le \ell$, has used more than $R = \left\lceil 10 \left(\frac{n}{\delta}\right)^\eta \log \frac{n}{\delta} \right\rceil$ trials, where η is a parameter to be fixed in Section 5.

If either of the above exceptions occurs, the algorithm terminates and returns a random $\boldsymbol{X}_S \sim \pi_S$.

In Section 7, we will show that assuming Condition 3.4 for the projection scheme $\boldsymbol{h}$ with properly chosen parameters (α, β) and by properly choosing η, for the random y_Λ upon which the subroutine is called in Algorithm 1, with high probability none of these exceptions occurs. Therefore, the random sample returned by the subroutine is accurate enough when being called in Algorithm 1.

5. Proofs of the main results

In this section, we prove the main theorems of this paper. Our algorithm first constructs a projection scheme using one of the procedures in Theorem 3.8 and Theorem 3.9, which gives us the projection oracle that can answer queries within time cost $O(\log q)$, where $q = \max_{v\in V} |Q_v|$. We then execute Algorithm 1 for sampling $\boldsymbol{X}$ approximately according to μ. We assume the following basic operations for uniform sampling:

- draw a variable $v \in V$ uniformly at random within time cost $O(\log n)$;
- for any variable $v \in V$, draw a uniform sample $X \sim \pi_v$ from Q_v within time cost $O(\log q)$.

When measuring the time cost of Algorithm 1, we count the number of calls to the projection oracle as well as the above two basic sampling operations. The time complexity of Algorithm 1 is dominated by these oracle costs.

Next, we prove Theorem 1.1 for general CSP formulas with atomic constraints, while Theorem 1.3 and Theorem 1.4 for specific subclasses of formulas are proved in Section 5.2.

5.1. **CSP formulas with atomic constraints.** For CSP formulas $\Phi = (V, \mathcal{Q}, C)$ defined by atomic constraints, we show that sampling uniform solution is efficient within the following regime:

$$\ln \frac{1}{p} \geq 350 \ln D + 3 \ln \frac{1}{\zeta} \tag{10}$$

where $p = \max_{c\in C} \prod_{v\in\mathsf{vbl}(c)} \frac{1}{|Q_v|}$ stands for the maximum probability that a constraint $c \in C$ is violated by uniform random assignment, and D stands for the maximum degree of the dependency graph of Φ. The positive constant parameter ζ specifies a gap to the boundary of the regime.

Theorem 5.1. *The following holds for any $0 < \zeta \leq 2^{-400}$. There is an algorithm such that given any $0 < \varepsilon < 1$ and CSP formula $\Phi = (V, \mathcal{Q}, C)$ with atomic constraints satisfying (10), the algorithm outputs a random assignment $\boldsymbol{X} \in \mathcal{Q}$ whose distribution is ε-close in total variation distance to the uniform distribution μ over all solutions to Φ, using time cost $O\left((D^2k + q)n\left(\frac{n}{\varepsilon}\right)^{\zeta}\log^4\left(\frac{nDq}{\varepsilon}\right)\right)$, where $k = \max_{c\in C} |\mathsf{vbl}(c)|$.*

Theorem 1.1 is implied by Theorem 5.1, by interpreting any LLL instance with uniform random variables and atomic bad events as a CSP formula with atomic constraints.

Let $\boldsymbol{h} = (h_v)_{v\in V}$ be a projection scheme satisfying Condition 3.4 with parameters α and β. To prove Theorem 5.1, we have the following lemma which shows that assuming a Lovász local lemma condition, the Glauber dynamics for the projected distribution ν is rapidly mixing.

Lemma 5.2. *If $\log\frac{1}{p} \geq \frac{50}{\beta}\log\left(\frac{2000D^4}{\beta}\right)$, then the Markov chain P_{Glauber} on ν has $T_{mix}(\varepsilon) \leq \left\lceil 2n\log\frac{n}{\varepsilon}\right\rceil$.*

The proof of Lemma 5.2 is given in Section 8.

We also need the following lemma for analyzing the subroutine $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, X_\Lambda, S)$. In Algorithm 1 the subroutine is called for $T+1$ times. For $1 \leq t \leq T+1$, define the following bad events:

- $\mathcal{B}_t^{(1)}$: in the t-th call of $\mathsf{InvSample}(\cdot)$, a random assignment $\boldsymbol{X}$ is returned in Line 10.
- $\mathcal{B}_t^{(2)}$: in the t-th call of $\mathsf{InvSample}(\cdot)$, a random assignment $\boldsymbol{X}$ is returned in Line 4

Lemma 5.3. *Let $1 \leq t \leq T+1$ and $0 < \eta < 1$. In Algorithm 1, for the t-th calling to the subroutine $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, y_\Lambda, S)$ with parameter η, it holds that*

- *given access to a projection oracle with query cost $O(\log q)$, the time cost of $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, y_\Lambda, S)$ is bounded as*
$$O\left(|S|\,D^2k\left(\frac{n}{\delta}\right)^{\eta}\log^2\left(\frac{nD}{\delta}\right)\log q\right),$$
where $k = \max_{c\in C} |\mathsf{vbl}(c)|$ and $q = \max_{v\in V} |Q_v|$;
- *conditioned on $\neg\mathcal{B}_t^{(1)} \wedge \neg\mathcal{B}_t^{(2)}$, the t-th calling to $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, y_\Lambda, S)$ returns a $\boldsymbol{X}_S \in Q_S$ that is distributed precisely according to $\mu_S^{y_\Lambda}$.*

Furthermore, if $\log \frac{1}{p} \geq \frac{1}{1-\alpha} \log(20D^2)$ *and* $\log \frac{1}{p} \geq \frac{1}{\beta} \log\left(\frac{40\mathrm{e}D^2}{\eta}\right)$ *it holds that*

$$\Pr\left[\mathcal{B}_t^{(1)}\right] \leq \delta \quad \textit{and} \quad \Pr\left[\mathcal{B}_t^{(2)}\right] \leq \delta.$$

The proof of Lemma 5.3 is given in Section 7.

Proof of Theorem 5.1. Let α, β, η be three parameters to be fixed later. Our algorithm first uses the algorithm in Theorem 3.8 with $\delta = \frac{\varepsilon}{4}$ to construct a projection scheme satisfying Condition 3.4 with parameters α and β. If the algorithm in Theorem 3.8 fails to find such projection scheme, our algorithm terminates and outputs an arbitrary $\boldsymbol{X}_{\text{out}} \in \boldsymbol{Q}$. If the algorithm finds such projection scheme, we run Algorithm 1 to obtain the random sample $\boldsymbol{X}_{\text{out}} = \boldsymbol{X}_{\text{alg}}$, where $\boldsymbol{X}_{\text{alg}}$ denotes the output of Algorithm 1.

We first analyze the running time of the whole algorithm. By Theorem 3.8, the running time for constructing the projection scheme is

$$T_{\text{proj}} = O\left(n(Dk+q)\log\frac{1}{\varepsilon}\log q\right).$$

If the algorithm in Theorem 3.8 succeeds, then it gives a projection oracle with query cost $O(\log q)$. In Algorithm 1, we simulate the Glauber dynamics for $T = \lceil 2n \log \frac{4n}{\varepsilon} \rceil$ transition steps. In each step, the algorithm first picks a variable $v \in V$ uniformly at random, the cost is $O(\log n)$. The algorithm then calls the subroutine $\mathsf{InvSample}\left(\Phi, \boldsymbol{h}, \frac{\varepsilon}{4(T+1)}, Y_{V\setminus\{v\}}, \{v\}\right)$ to draw a random $X_v \in Q_v$. By Lemma 5.3, the cost of the subroutine is $O\left(D^2 k \left(\frac{n}{\delta}\right)^{\eta} \log^2\left(\frac{nD}{\delta}\right)\log q\right)$, where

$$\delta = \frac{\varepsilon}{4(T+1)} = \Theta\left(\frac{\varepsilon}{n\log\frac{n}{\varepsilon}}\right) = \Omega\left(\frac{\varepsilon^2}{n^2}\right).$$

After X_v is sampled in Line 4, the algorithm calls the projection oracle to map $X_v \in Q_v$ to $Y_v = h_v(v) \in \Sigma_v$, the cost of this step is $O(\log q)$. Thus, the cost for simulating each transition step is

$$T_{\text{step}} = O\left(D^2 k \left(\frac{n}{\varepsilon}\right)^{3\eta}\log^2\left(\frac{nD}{\varepsilon}\right)\log q\right). \tag{11}$$

Finally, the algorithm uses $\mathsf{InvSample}\left(\Phi, \boldsymbol{h}, \frac{\varepsilon}{4(T+1)}, Y, V\right)$ in Line 6 to sample the final output. By Lemma 5.3, the cost is $O\left(nD^2 k \left(\frac{n}{\delta}\right)^{\eta}\log^2\left(\frac{nD}{\delta}\right)\log q\right)$, where $\delta = \frac{\varepsilon}{4(T+1)} = \Omega\left(\frac{\varepsilon^2}{n^2}\right)$. Hence, the cost for the last step is

$$T_{\text{final}} = O\left(nD^2 k \left(\frac{n}{\varepsilon}\right)^{3\eta}\log^2\left(\frac{nD}{\varepsilon}\right)\log q\right). \tag{12}$$

Combining all of them together, the total running time is

$$T_{\text{total}} = T_{\text{proj}} + T\cdot T_{\text{step}} + T_{\text{final}} = O\left(n(Dk+q)\log\frac{1}{\varepsilon}\log q\right) + O\left((T+n)D^2 k\left(\frac{n}{\varepsilon}\right)^{3\eta}\log^2\left(\frac{nD}{\varepsilon}\right)\log q\right)$$

$$= O\left((D^2k+q)n\left(\frac{n}{\varepsilon}\right)^{3\eta}\log^3\left(\frac{nD}{\varepsilon}\right)\log q\right). \tag{13}$$

Next, we prove the correctness of the algorithm, i.e., the total variation distance between the output $\boldsymbol{X}_{\text{out}}$ and the uniform distribution μ is at most ε. It suffices to prove

$$d_{\mathrm{TV}}\left(\boldsymbol{X}_{\text{alg}}, \mu\right) \leq \frac{3\varepsilon}{4}. \tag{14}$$

Because if $0 < \beta < \alpha < 1$ and $\log\frac{1}{p} \geq \frac{25}{(\alpha-\beta)^3}(\log D + 3)$, then with probability at least $1 - \frac{\varepsilon}{4}$, the algorithm in Theorem 3.8 constructs the projection scheme successfully, i.e. $\boldsymbol{X}_{\text{out}} = \boldsymbol{X}_{\text{alg}}$. Let $\boldsymbol{X} \sim \mu$. By coupling lemma, we can couple $\boldsymbol{X}$ and $\boldsymbol{X}_{\text{alg}}$ such that $\boldsymbol{X} \neq \boldsymbol{X}_{\text{alg}}$ with probability $\frac{3\varepsilon}{4}$. Thus, we can coupling $\boldsymbol{X}$ and $\boldsymbol{X}_{\text{out}}$ such that $\boldsymbol{X} \neq \boldsymbol{X}_{\text{out}}$ with probability at most $\frac{\varepsilon}{4} + \frac{3\varepsilon}{4} = \varepsilon$. By coupling lemma,

$$d_{\mathrm{TV}}\left(\boldsymbol{X}_{\text{out}}, \mu\right) \leq \varepsilon.$$

We then verify (14). Consider an idealized algorithm that first runs the idealized Glauber dynamics for $T = \lceil 2n \log \frac{4n}{\varepsilon} \rceil$ steps to obtain a random sample Y_{G}, then samples X_{idea} from the distribution $\mu^{Y_{\mathrm{G}}}$. By Lemma 5.2, if $\log \frac{1}{p} \geq \frac{50}{\beta} \log \left(\frac{2000D^4}{\beta} \right)$, then $d_{\mathrm{TV}}\left(Y_{\mathrm{G}}, \nu\right) \leq \frac{\varepsilon}{4}$. Consider the following process to draw a random sample $X \sim \mu$. First sample $Y \sim \nu$, then sample $X \sim \mu^{Y}$. Thus, we can couple Y and Y_G such that $Y \neq Y_{\mathrm{G}}$ with probability $\frac{\varepsilon}{4}$. Conditional on $Y = Y_{\mathrm{G}}$, X and X_{idea} can be perfectly coupled. By coupling lemma,

$$d_{\mathrm{TV}}\left(X_{\mathrm{idea}}, \mu\right) \leq \frac{\varepsilon}{4}. \tag{15}$$

We now couple Algorithm 1 with this idealized algorithm. For each transition step, they pick the same variable, then couple each transition step optimally. In the last step, they use the optimal coupling to draw random samples from the conditional distributions. Note that in Line 4 of Algorithm 1, if the random sample $X_v \in Q_v$ returned by the subroutine is a perfect sample from $\mu_v^{Y_{V \setminus \{v\}}}$, then the $Y_v \in \Sigma_v$ constructed in Line 5 follows the distribution $\nu_v^{Y_{V \setminus \{v\}}}$. By Lemma 8.12, if none of $\mathcal{B}_t^{(1)}$ and $\mathcal{B}_t^{(2)}$ for $1 \leq t \leq T+1$ occurs, then all the $(T+1)$ executions of the subroutine $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, y_\Lambda, S)$ return perfect samples from $\mu_S^{y_\Lambda}$. In this case, Algorithm 1 and the idealized algorithm can be coupled perfectly. Note that $\delta = \frac{\varepsilon}{4(T+1)}$. By coupling lemma and Lemma 5.3, we have

$$d_{\mathrm{TV}}\left(X_{\mathrm{alg}}, X_{\mathrm{idea}}\right) \leq \Pr\left[\bigvee_{i=1}^{T+1} \left(\mathcal{B}_t^{(1)} \vee \mathcal{B}_t^{(2)}\right)\right] \leq 2(T+1)\delta = \frac{\varepsilon}{2}.$$

Hence, (14) can be proved by the following triangle inequality

$$d_{\mathrm{TV}}\left(X_{\mathrm{alg}}, \mu\right) \leq d_{\mathrm{TV}}\left(X_{\mathrm{alg}}, X_{\mathrm{idea}}\right) + d_{\mathrm{TV}}\left(X_{\mathrm{idea}}, \mu\right) \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{4} \leq \frac{3\varepsilon}{4}.$$

We then set the parameters α, β and η. We put all the constraints in Theorem 3.8, Lemma 5.2 and Lemma 5.3 together:

$$\begin{aligned}
&0 < \beta < \alpha < 1, \quad 0 < \eta < 1;\\
&\log \frac{1}{p} \geq \frac{25}{(\alpha-\beta)^3} (\log D + 3);\\
&\log \frac{1}{p} \geq \frac{50}{\beta} \log \left(\frac{2000D^4}{\beta} \right);\\
&\log \frac{1}{p} \geq \frac{1}{1-\alpha} \log(20D^2);\\
&\log \frac{1}{p} \geq \frac{1}{\beta} \log \left(\frac{40\mathrm{e}D^2}{\eta} \right).
\end{aligned}$$

We can take $\alpha = 0.994$ and $\beta = 0.577$. The following condition implies all the above constraints

$$\log \frac{1}{p} \geq 350 \log D + 3 \log \frac{1}{\zeta} \quad \text{and} \quad \eta = \frac{\zeta}{3}, \quad \text{where } 0 < \zeta \leq 2^{-400}.$$

Remark that $\log \frac{1}{p} \geq 350 \log D + 3 \log \frac{1}{\zeta}$ is equivalent to $\ln \frac{1}{p} \geq 350 \ln D + 3 \ln \frac{1}{\zeta}$. By (13), under this condition, the total running time is

$$T_{\mathrm{total}} = O\left((D^2k + q)n \left(\frac{n}{\varepsilon}\right)^{3\eta} \log^3 \left(\frac{nD}{\varepsilon}\right) \log q \right) = O\left((D^2k + q)n \left(\frac{n}{\varepsilon}\right)^{\zeta} \log^4 \left(\frac{nDq}{\varepsilon}\right) \right). \qquad \square$$

5.2. Sharper bounds for subclasses of CSP formulas.

We prove the following theorems on specific subclasses of CSP formulas. Our first result is for hypergraph coloring.

Theorem 5.4. *There is an algorithm such that given any k-uniform hypergraph with maximum degree Δ and a set of colors $[q]$, assuming $k \geq 13$ and $q \geq \max\left((7k\Delta)^{\frac{9}{k-12}}, 650 \right)$, the algorithm returns a random q-coloring $X \in [q]^V$ in time $O\left(q^2 k^3 \Delta^2 n \left(\frac{n}{\varepsilon}\right)^{\frac{1}{100(qk\Delta)^4}} \log^4 \left(\frac{nqk\Delta}{\varepsilon}\right) \right)$ such that the distribution of*

$\boldsymbol{X}$ is ε-close in total variation distance to the uniform distribution of all proper q-colorings of the input hypergraph.

Theorem 1.3 is implied by Theorem 5.4: when $k \geq 30$, we have $(7k)^{\frac{9}{k-12}} \leq 15$, which means that $q \geq 15\Delta^{\frac{9}{k-12}} + 650$ suffices to imply the condition in Theorem 5.4.

Our next result is for CNF formulas. For a k-CNF formula, each clause contains k variables. And the *maximum degree* of the formula is given by the maximum number of clauses a variable belongs to. The following theorem is is a formal restatement of Theorem 1.4.

Theorem 5.5. *The following holds for any $0 < \zeta \leq 2^{-20}$. There is an algorithm such that given any k-CNF formula with maximum degree d, assuming $k \geq 13\log d + 13\log k + 3\log\frac{1}{\zeta}$, the algorithm returns a random assignment $\boldsymbol{X} \in \{\mathtt{True}, \mathtt{False}\}^V$ in time $O\left(d^2k^3n\left(\frac{n}{\varepsilon}\right)^{\zeta/(dk)^4}\log^3\left(\frac{ndk}{\varepsilon}\right)\right)$ such that the distribution of $\boldsymbol{X}$ is ε-close in total variation distance to the uniform distribution of all satisfying assignments.*

Let $\Phi = (V, [q]^V, C)$ denote the CSP formula where all variables have the same domain $[q]$. Suppose that for every constraint $c \in C$, c is atomic and $|\mathsf{vbl}(c)| = k$, and each variable belongs to at most d constraints. Let $\boldsymbol{h}$ denote a projection scheme satisfying Condition 3.4 with parameters α and β. For such special CSP formulas, we have the following lemma with an improved mixing condition.

Lemma 5.6. *If $k\log q \geq \frac{1}{\beta}\log\left(3000q^2d^6k^6\right)$, then the Markov chain P_{Glauber} on ν has $T_{mix}(\varepsilon) \leq \left\lceil 2n\log\frac{n}{\varepsilon}\right\rceil$.*

The proof of Lemma 5.6 is given in Section 8. We use Lemma 5.3 and Lemma 5.6 to prove our results.

Proof of Theorem 5.4. Consider the hypergraph q-coloring on a k-uniform hypergraph $H = (V, \mathcal{E})$ with maximum degree Δ. We first transform the hypergraph coloring instance into a CSP formula $\Phi = (V, [q]^V, C)$ with atomic constraints. For each hyperedge $e \in \mathcal{E}$, we add q constraints such that the i-th constraint c_i forbids the bad event that the hyperedge e is monochromatic with color $i \in [q]$. Namely, $\mathsf{vbl}(c_i) = e$ and c_i is $\mathtt{False}$ if and only if all variables in $\mathsf{vbl}(c_i)$ take the value i. The time complexity for this reduction is $O(nq\Delta\log q)$.

In CSP formula $\Phi = (V, [q]^V, C)$, c is atomic and $|\mathsf{vbl}(c)| = k$ for all $c \in C$; each variable belongs to at most $q\Delta$ constraints. The maximum degree D of the dependency graph of Φ is at most $qk\Delta$. We assume $D = qk\Delta$. If each variable $v \in V$ draws a random value from $[q]$ uniformly and independently, then the maximum probability p that one constraint becomes $\mathtt{False}$ is $p = \left(\frac{1}{q}\right)^k$.

Let α, β, η be three parameters to be fixed later. Our algorithm first uses the deterministic algorithm in Theorem 3.9 to construct a projection scheme satisfying Condition 3.4 with parameters α and β. The deterministic algorithm in Theorem 3.9 always finds such a projection scheme, which gives a projection oracle with query cost $O(\log q)$. Remark that the cost for constructing the projection scheme is

$$T_{\text{proj}} = O\left(n\log q\right). \tag{16}$$

We then run Algorithm 1 to obtain the output $\boldsymbol{X}_{\text{out}} = \boldsymbol{X}_{\text{alg}}$, where $\boldsymbol{X}_{\text{alg}}$ denotes the output of Algorithm 1. The correctness result can be proved by going through the proof of Theorem 1.1.

We set parameters α, β and η. Note that $\mathsf{vbl}(c) = k$ for all $c \in C$; $p = q^{-k}$; and each variable belongs to at most $d = q\Delta$ constraints; and $D = qk\Delta$. We put all the constraints in Theorem 3.9, Lemma 5.6 and Lemma 5.3 together:

$$0 < \beta < \alpha < 1, \quad 7 \leq q^{\frac{\alpha+\beta}{2}} \leq \frac{q}{6}, \quad \log q \geq \frac{1}{\alpha-\beta}, \quad 0 < \eta < 1;$$
$$k\log q \geq \frac{1}{\beta}\log\left(3000q^8\Delta^6k^6\right);$$
$$k\log q \geq \frac{1}{1-\alpha}\log(20q^2k^2\Delta^2);$$
$$k\log q \geq \frac{1}{\beta}\log\left(\frac{40\mathrm{e}q^2k^2\Delta^2}{\eta}\right).$$

We can take $\alpha = \frac{7}{9}$ and $\beta = \frac{2}{3}$. The following condition suffices to imply all the above constraints: assume $k > 12$,

$$\log q \geq \frac{9}{k-12}\log\Delta + \frac{9}{k-12}\log k + \frac{25}{k-12}, \quad q \geq 650, \quad \eta = \frac{1}{2^9(qk\Delta)^4}.$$

The following condition suffices to imply the above one

$$q \geq \max\left((7k\Delta)^{\frac{9}{k-12}}, 650\right) \quad \text{and} \quad \eta = \frac{1}{2^9(qk\Delta)^4}.$$

Note that $D = kq\Delta$. Under this condition, by (11), (12) and (16), the total running time is

$$T_{\text{total}} = O\left(D^2 kn\left(\frac{n}{\varepsilon}\right)^{3\eta}\log^3\left(\frac{nD}{\varepsilon}\right)\log q\right) = O\left(q^2k^3\Delta^2 n\left(\frac{n}{\varepsilon}\right)^{\frac{1}{100(qk\Delta)^4}}\log^4\left(\frac{nqk\Delta}{\varepsilon}\right)\right). \qquad \square$$

Proof of Theorem 5.5. Let $\Phi = (V, \{\texttt{True}, \texttt{False}\}^V, C)$ be a k-CNF formula, where each variable belongs to at most d clauses. Each variable takes its value for the Boolean domain $\{\texttt{True}, \texttt{False}\}$, thus the size of the domain is $q = 2$. The maximum degree D of the dependency graph is at most kd. We assume $D = kd$. If each variable $v \in V$ draws a random value from the Boolean domain $\{\texttt{True}, \texttt{False}\}$ uniformly and independently, the maximum probability p that one clause is not satisfied is $p = \left(\frac{1}{2}\right)^k$.

Let α, β, η be three parameters to be fixed later. Our algorithm first uses the randomized algorithm in Theorem 3.9 with $\delta = \frac{\varepsilon}{4}$ to construct a projection scheme satisfying Condition 3.4 with parameters α and β. If the randomized algorithm in Theorem 3.9 fails to find such projection scheme, our algorithm terminates and outputs an arbitrary $\boldsymbol{X}_{\text{out}} \in \{\texttt{True}, \texttt{False}\}^V$. If the randomized algorithm in Theorem 3.9 succeeds, it gives a projection oracle with query cost $O(\log q)$. By Theorem 3.9, the cost for constructing the projection scheme is

$$T_{\text{proj}} = O\left(ndk\log\frac{1}{\varepsilon}\right). \tag{17}$$

We then run Algorithm 1 to obtain the output $\boldsymbol{X}_{\text{out}} = \boldsymbol{X}_{\text{alg}}$, where $\boldsymbol{X}_{\text{alg}}$ denotes the output of Algorithm 1. The correctness result can be proved by going through the proof of Theorem 1.1.

We set parameters α, β and η. We put all the constraints in Theorem 3.9, Lemma 5.6 and Lemma 5.3 together:

$$\begin{aligned} 0 < \beta < \alpha < 1, \quad k &\geq \frac{2\ln 2}{(\alpha-\beta)^2}\log(2\mathrm{e}kd), \quad 0 < \eta < 1;\\ k &\geq \frac{1}{\beta}\log\left(3000\cdot 4\cdot d^6k^6\right);\\ k &\geq \frac{1}{1-\alpha}\log(20d^2k^2);\\ k &\geq \frac{1}{\beta}\log\left(\frac{40\mathrm{e}d^2k^2}{\eta}\right). \end{aligned}$$

We can take $\alpha = \frac{21}{25}$ and $\beta = \frac{1}{2}$. The following condition suffices to imply all the above constraints

$$k \geq 13\log d + 13\log k + 3\log\frac{1}{\zeta} \quad \text{and} \quad \eta = \frac{\zeta}{3d^4k^4}, \quad \text{where } 0 < \zeta \leq 2^{-20}.$$

Note that $D = dk$ and $q = 2$. Under this condition, by (11), (12) and (17), the total running time is

$$T_{\text{total}} = O\left(D^2kn\left(\frac{n}{\varepsilon}\right)^{3\eta}\log^3\left(\frac{nD}{\varepsilon}\right)\log q\right) = O\left(d^2k^3n\left(\frac{n}{\varepsilon}\right)^{\frac{\zeta}{d^4k^4}}\log^3\left(\frac{ndk}{\varepsilon}\right)\right). \qquad \square$$

6. Projection construction

In this section, we give the algorithms to construct the projection schemes. We first give the projection algorithm for (k, d)-CSP formulas (Theorem 3.9), then give the projection algorithm for general CSP formulas (Theorem 3.8).

Proof of Theorem 3.9. We start from the first part of the lemma. For each $v \in V$, we set s_v as

$$s_v = \left\lceil q^{\frac{2-\alpha-\beta}{2}} \right\rceil.$$

For each variable $v \in V$, we partition $[q] = \{1, 2, \ldots, q\}$ into s_v intervals, where the sizes of the first $(q \mod s_v)$ intervals are $\lceil q/s_v \rceil$, and the sizes of the last $s_v - (q \mod s_v)$ intervals are $\lfloor q/s_v \rfloor$. Let $\Sigma_v = \{1, 2, \ldots, s_v\}$. For each $i \in [q]$, $h_v(i) = j \in \Sigma_v$, where i belongs to the j-th interval. This constructs the function $h_v : [q] \to \Sigma_v$. To implement the projection oracle, we only need to calculate s_v for each $v \in V$, the total cost is $O(n \log q)$. Consider the two queries in Definition 3.3.

- evaluation: given an input value $i \in [q]$ of a variable $v \in V$, the algorithm should return $j \in \Sigma_v$ such that i is in the j-th interval, this query can be answered with the cost $O(\log q)$;
- inversion: given a projected value $j \in \Sigma_v$ of a variable $v \in V$, the algorithm should return a random element in the j-th interval uniformly at random, this query can be answered with the cost $O(\log q)$.

Next, we prove that this projection scheme satisfies Condition 3.4. For any $v \in V$, it holds that

$$\left\lceil \frac{q}{s_v} \right\rceil \le \left\lceil q^{(\alpha+\beta)/2} \right\rceil \le q^{(\alpha+\beta)/2} + 1 \overset{\diamond}{\le} \frac{7}{6} q^{(\alpha+\beta)/2},$$

where $(\diamond)$ holds because $q^{(\alpha+\beta)/2} + 1 \le \frac{7}{6} q^{(\alpha+\beta)/2}$ if $q^{(\alpha+\beta)/2} \ge 6$. Note that $\log \frac{7}{6} \le 0.23$. This implies the following inequality

$$\sum_{v \in \mathsf{vbl}(c)} \left\lceil \frac{q}{s_v} \right\rceil \le k \left(\frac{\alpha+\beta}{2} \log q + 0.23 \right) \overset{(\star)}{\le} k \cdot \alpha \log q = \alpha \sum_{v \in \mathsf{vbl}(c)} \log q, \tag{18}$$

where inequality $(\star)$ holds because $\alpha > \beta$ and $\log q \ge \frac{0.8}{\alpha - \beta}$. For any $v \in V$, it holds that

$$\left\lfloor \frac{q}{s_v} \right\rfloor = \left\lfloor \frac{q}{\left\lceil q^{(2-\alpha-\beta)/2} \right\rceil} \right\rfloor \ge \left\lfloor \frac{q}{q^{(2-\alpha-\beta)/2} + 1} \right\rfloor \overset{(*)}{\ge} \left\lfloor \frac{q}{\left(1 + \frac{1}{6}\right) q^{(2-\alpha-\beta)/2}} \right\rfloor \ge \frac{6}{7} q^{\frac{\alpha+\beta}{2}} - 1 \overset{(\diamond)}{\ge} \frac{5}{7} q^{\frac{\alpha+\beta}{2}},$$

where inequality $(*)$ holds because $\left(1 + \frac{1}{6}\right) q^{(2-\alpha-\beta)/2} \ge q^{(2-\alpha-\beta)/2} + 1$ if $q^{(2-\alpha-\beta)/2} \ge 6$; inequality $(\diamond)$ holds because $q^{(\alpha+\beta)/2} \ge 7$. Note that $\log \frac{5}{7} \ge -0.5$. This implies

$$\sum_{v \in \mathsf{vbl}(c)} \log \left\lfloor \frac{q}{s_v} \right\rfloor \ge k \left(\frac{\alpha+\beta}{2} \log q - 0.5 \right) \overset{(\star)}{\ge} k \cdot \beta \log q = \beta \sum_{v \in \mathsf{vbl}(c)} \log q, \tag{19}$$

where inequality $(\star)$ holds because $\alpha > \beta$ and $\log q \ge \frac{1}{\alpha - \beta}$. Combining (18) and (19) proves the first part of the lemma.

We then prove the second part of the lemma. The algorithm constructs a subset of variables $\mathcal{M} \subseteq V$. We call $\mathcal{M}$ the set of marked variables. If $v \in \mathcal{M}$, let $\Sigma_v = [q]$, and $h_v(i) = i$ for all $i \in [q]$. If $v \notin \mathcal{M}$, let $\Sigma_v = \{1\}$, and $h_v(i) = 1$ for all $i \in [q]$. Remark that $s_v = q$ if v is a marked variable, and $s_v = 1$ if v is an unmarked variable. To implement the projection oracle, we only need to construct $\mathcal{M}$. Suppose the set $\mathcal{M}$ is given (the construction will be explained later). Consider the two queries in Definition 3.3.

- evaluation: given an input value $i \in [q]$ of a variable $v \in V$, the algorithm should return the input i if $v \in \mathcal{M}$, or return $1 \in \Sigma_v$ if $v \notin \mathcal{M}$; this query can be answered with the cost $O(\log q)$;
- inversion: given a projected value $j \in \Sigma_v$ of a variable $v \in V$, the algorithm should return $j \in [q]$ if $v \in \mathcal{M}$; or return a uniform random element $X \in [q]$ if $v \notin \mathcal{M}$; this query can be answered with the cost $O(\log q)$.

Now, we construct the set of marked variables $\mathcal{M} \subseteq V$. For each constraint $c \in C$, define t_c as the number of marked variables in c, i.e.

$$t_c \triangleq |\mathcal{M} \cap \mathsf{vbl}(c)|.$$

Hence, Condition 3.4 becomes for each $c \in C$,

$$(1-\alpha)k \le t_c \le (1-\beta)k.$$

In other words, each constraint contains at least $(1-\alpha)k$ marked variables and at least βk unmarked variables. We use Lovász local lemma to show that such set $\mathcal{M}$ exists, then use Moser-Tardos algorithm to find a set $\mathcal{M}$. Let $\mathcal{D}$ denote the product distribution such that each variables is marked independently with probability $\frac{2-\alpha-\beta}{2}$. For each constraint $c \in C$, let B_c denote the bad event that c contains less than $(1-\alpha)k$ marked variables or less than βk unmarked variables. We use concentration inequality to bound the probability of B_c. In [FGYZ20], the probability of the bad event B_c is bounded by the Chernoff bound. Now, we use Hoeffding's inequality to obtain a better result

$$\Pr_{\mathcal{D}}[B_c] = \Pr[t_c < (1-\alpha)k \vee t_c > (1-\beta)k] = \Pr\left[|t_c - \mathrm{E}[t_c]| \ge \frac{\alpha-\beta}{2}k\right] \le 2\exp\left(-\frac{(\alpha-\beta)^2}{2}k\right).$$

The maximum degree of dependency graph is at most $k(d-1)$. By Lovász local lemma (Theorem 2.1), the set $\mathcal{M}$ exist if

$$\mathrm{e}\cdot 2\exp\left(-\frac{(\alpha-\beta)^2}{2}k\right)\cdot kd \le 1.$$

Note that $\alpha > \beta$ and $k \ge \frac{2\ln 2}{(\alpha-\beta)^2}(\log k + \log d + \log 2\mathrm{e})$ implies the above condition.

The Moser-Tardos algorithm can find such set $\mathcal{M}$ within $\frac{2n}{k}$ resampling steps in expectation [MT10]. We can 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. The cost of each resampling step is $O(dk^2)$. The cost for constructing data structure is $O(ndk\log\frac{1}{\delta})$. ◻

Proof of Theorem 3.8. The domain of each variable $v \in V$ is Q_v, where $q_v = |Q_v|$. Assume each element $x \in Q_v$ can be in-coded by $O(\log q_v)$ bits. For each $v \in V$, suppose the input provides an array $\mathcal{A}_v$ of size q_v containing all the elements in Q_v. For each $v \in V$, we construct a data structure $\mathcal{S}_v$ that can answer the following two types of the queries: (1) given any index $i \in [q_v]$, we can access the i-th element in this array with cost $O(\log q_v)$. (2) given any $x \in Q_v$, we can find the unique index i such that $\mathcal{A}_v(i) = x$ with the cost $O(\log q_v)$. For each $v \in V$, the cost of the construction is $O(q_v \log q_v)$.

The algorithm divides all variables into two parts S_{large} and S_{small} such that

$$S_{\mathrm{large}} = \left\{v \in V \mid \log q_v \ge \frac{5}{\alpha-\beta}\right\}, \qquad S_{\mathrm{small}} = \left\{v \in V \mid \log q_v < \frac{5}{\alpha-\beta}\right\}.$$

For each variable $v \in S_{\mathrm{large}}$, the algorithm sets

$$\forall v \in S_{\mathrm{large}}, \quad s_v = \left\lceil q_v^{\frac{2-\alpha-\beta}{2}} \right\rceil.$$

We partition $[q] = \{1,2,\ldots,q\}$ into s_v intervals, where the sizes of the first $(q \mod s_v)$ intervals are $\lceil q/s_v \rceil$, and the sizes of the last $s_v - (q \mod s_v)$ intervals are $\lfloor q/s_v \rfloor$. Let $\Sigma_v = \{1,2,\ldots,s_v\}$, where each $j \in \Sigma_v$ represents an interval $[L_j, R_j]$. For any $x \in Q_v$, let i denote the unique index such that $\mathcal{A}_v(i) = x$, we set $h_v(x) = j$ such that $i \in [L_j, R_j]$. This defines the function $h_v : Q_v \to \Sigma_v$. To implement the projection oracle for S_{large}, the algorithm only needs to compute the value of s_v, where the cost is $O(\log q_v)$. Consider the two queries of the projection oracle in Definition 3.3.

- evaluation: given an input value $x \in Q_v$ of a variable $v \in S_{\mathrm{large}}$, with the data structure $\mathcal{S}_v$, the algorithm can return $h_v(x)$ in time $O(\log q_v)$;
- inversion: given a projected value $j \in \Sigma_v$ of a variable $v \in S_{\mathrm{large}}$, the algorithm should return a uniform element in set $\{x \in \mathcal{A}_v(i) \mid L_j \le i \le R_j\}$; with the data structure $\mathcal{S}_v$, this query can be answered with the cost $O(\log q_v)$.

Let $q = \max_{v \in V} q_v$. For any $v \in S_{\text{large}}$, the cost for answering each query is $O(\log q)$.

For variables in S_{small}, the algorithm constructs a subset of variables $\mathcal{M} \subseteq S_{\text{small}}$. We call $\mathcal{M}$ the set of marked variables. If $v \in \mathcal{M}$, let $\Sigma_v = Q_v$, and $h_v(x) = x$ for all $x \in Q_v$. If $v \notin \mathcal{M}$, let $\Sigma_v = \{1\}$, and $h_v(x) = 1$ for all $x \in Q_v$. To implement the projection oracle, the algorithm only needs to construct the set $\mathcal{M}$. The construction of $\mathcal{M}$ will be explained later. Suppose the set $\mathcal{M} \subseteq S_{\text{small}}$ is given. Consider the two queries of the projection oracle in Definition 3.3.

- evaluation: given an input value $x \in Q_v$ of a variable $v \in S_{\text{small}}$, the algorithm should return the input x if $v \in \mathcal{M}$, or return $1 \in \Sigma_v$ if $v \notin \mathcal{M}$; this query can be answered in time $O(\log q_v)$;
- inversion: given a projected value $x \in \Sigma_v$ of a variable $v \in S_{\text{small}}$, the algorithm should return the input x if $v \in \mathcal{M}$; or return a uniform random element $X \in Q_v$ if $v \notin \mathcal{M}$; with the data structure $\mathcal{S}_v$, this query can be answered in time $O(\log q_v)$.

Let $q = \max_{v \in V} q_v$. For any $v \in S_{\text{small}}$, the cost for answering each query is $O(\log q)$.

Again, we use Lovász local lemma to prove that there is a subset $\mathcal{M}$ such that the above projection scheme satisfies Condition 3.4, then use Moser-Tardos algorithm to find such set $\mathcal{M}$. Let $\mathcal{D}$ denote the product distribution such that each variable $v \in S_{\text{small}}$ is marked with probability $\frac{2-\alpha-\beta}{2}$. For each $c \in C$, let B_c denote the bad event

$$\sum_{v \in \mathsf{vbl}(c)} \log \left\lceil \frac{q_v}{s_v} \right\rceil > \alpha \sum_{v \in \mathsf{vbl}(c)} \log q_v \quad \text{or} \quad \sum_{v \in \mathsf{vbl}(c)} \log \left\lceil \frac{q_v}{s_v} \right\rceil < \beta \sum_{v \in \mathsf{vbl}(c)} \log q_v. \tag{20}$$

Fix a constraint $c \in C$. Suppose $v_1, v_2, \ldots, v_k$ are variables in $\mathsf{vbl}(c)$, where $k = k(c) = |\mathsf{vbl}(c)|$. Let $0 \le \ell \le k$ be an integer and assume $v_i \in S_{\text{large}}$ for all $1 \le i \le \ell$ and $v_j \in S_{\text{small}}$ for all $\ell+1 \le j \le k$. For each $1 \le i \le k$, we define random variable

$$X_i \triangleq \log \left\lceil \frac{q_{v_i}}{s_{v_i}} \right\rceil.$$

For each $1 \le i \le \ell$, since $v_i \in S_{\text{large}}$, $X_i = \log \left\lceil q_{v_i} / \lceil q_{v_i}^{(2-\alpha-\beta)/2} \rceil \right\rceil$ with probability 1. We have

$$\forall 1 \le i \le \ell, \quad \mathrm{E}[X_i] = \log \left\lceil \frac{q_{v_i}}{\lceil q_{v_i}^{(2-\alpha-\beta)/2} \rceil} \right\rceil \le \log \left\lceil q_{v_i}^{(\alpha+\beta)/2} \right\rceil \le \log \left(\frac{5}{4} q_{v_i}^{(\alpha+\beta)/2} \right),$$

where the last inequality holds because $\log q_{v_i} \ge \frac{5}{\alpha-\beta}$, which implies $\frac{5}{4} q_{v_i}^{(\alpha+\beta)/2} \ge q_{v_i}^{(\alpha+\beta)/2} + 1 \ge \left\lceil q_{v_i}^{(\alpha+\beta)/2} \right\rceil$. Note that $\log \frac{5}{4} \le 0.33$ and $\log q_{v_i} \ge \frac{5}{\alpha-\beta}$. It holds that

$$\forall 1 \le i \le \ell, \quad \mathrm{E}[X_i] \le 0.33 + \frac{\alpha+\beta}{2} \log q_{v_i} \le \alpha \log q_{v_i} - \frac{\alpha-\beta}{3} \log q_{v_i}. \tag{21}$$

For each $\ell+1 \le j \le k$, since $v_j \in S_{\text{small}}$, $X_j = \log q_{v_j}$ with probability $\frac{\alpha+\beta}{2}$; and $X_j = 0$ with probability $\frac{1-\alpha-\beta}{2}$. We have

$$\forall \ell+1 \le j \le k, \quad \mathrm{E}[X_i] = \frac{\alpha+\beta}{2} \log q_{v_i} \le \alpha \log q_{v_i} - \frac{\alpha-\beta}{3} \log q_{v_i}. \tag{22}$$

Consider the sum $\sum_{i=1}^k X_i$. For any $v_i \in S_{\text{large}}$, the value of X_i is fixed. For any $v_j \in S_{\text{small}}$, X_j takes a random value and it must hold that $X_j \in \{0, \log q_{v_j}\}$. By Hoeffding's inequality,

$$\Pr_{\mathcal{D}} \left[\sum_{i=1}^k X_i > \sum_{i=1}^k \mathrm{E}[X_i] + t \right] \le \exp\left(-\frac{2t^2}{\sum_{j=\ell+1}^k \log^2 q_{v_j}} \right) \overset{(\star)}{\le} \exp\left(-\frac{2(\alpha-\beta)t^2}{5 \sum_{j=\ell+1}^k \log q_{v_j}} \right), \tag{23}$$

where $(\star)$ holds due to $\log q_{v_j} \le \frac{5}{\alpha-\beta}$ for all $\ell+1 \le j \le k$. Combining (21), (22) and (23), we have

$$\Pr_{\mathcal{D}} \left[\sum_{i=1}^k X_i > \alpha \sum_{i=1}^k \log q_{v_i} \right] \le \exp\left(-\frac{\frac{2(\alpha-\beta)^3}{9} \left(\sum_{i=1}^k \log q_{v_i} \right)^2}{5 \sum_{j=\ell+1}^k \log q_{v_j}} \right) \le \exp\left(-\frac{(\alpha-\beta)^3}{23} \sum_{i=1}^k \log q_{v_i} \right). \tag{24}$$

Similarly, for each $1 \le i \le k$, we define random variable

$$Y_i \triangleq \log \left\lfloor \frac{q_{v_i}}{s_{v_i}} \right\rfloor.$$

For each $1 \le i \le \ell$, since $v_i \in S_{\text{large}}$, $Y_i = \log \left\lfloor \frac{q_{v_i}}{\lceil q_{v_i}^{(2-\alpha-\beta)/2} \rceil} \right\rfloor$ with probability 1. We have

$$\forall 1 \le i \le \ell, \quad \mathrm{E}\left[Y_i\right] = \log \left\lfloor \frac{q_{v_i}}{\left\lceil q_{v_i}^{(2-\alpha-\beta)/2} \right\rceil} \right\rfloor \ge \log \left\lfloor \frac{4}{5} q_{v_i}^{(\alpha+\beta)/2} \right\rfloor \ge \log \left(\frac{3}{5} q_{v_i}^{(\alpha+\beta)/2} \right),$$

where the last two inequalities hold because $0 < \beta < \alpha < 1$ and $\log q_{v_i} \ge \frac{5}{\alpha-\beta}$, which implies $\frac{5}{4} q_{v_i}^{(2-\alpha-\beta)/2} \ge q_{v_i}^{(2-\alpha-\beta)/2} + 1 \ge \left\lceil q_{v_i}^{(2-\alpha-\beta)/2} \right\rceil$ and $\left\lfloor \frac{4}{5} q_{v_i}^{(\alpha+\beta)/2} \right\rfloor \ge \frac{4}{5} q_{v_i}^{(\alpha+\beta)/2} - 1 \ge \frac{3}{5} q_{v_i}^{(\alpha+\beta)/2}$. Note that $\log \frac{3}{5} \ge -0.74$. Again, by $\log q_{v_i} \ge \frac{5}{\alpha-\beta}$, we have

$$\forall 1 \le i \le \ell, \quad \mathrm{E}\left[Y_i\right] \ge -0.74 + \frac{\alpha+\beta}{2} \log q_{v_i} \ge \beta \log q_{v_i} + \frac{\alpha-\beta}{3} \log q_{v_i}.$$

For each $\ell+1 \le j \le k$, since $v_j \in S_{\text{small}}$, $Y_j = 0$ with probability $\frac{2-\alpha-\beta}{2}$; and $Y_j = \log q_{v_j}$ with probability $\frac{\alpha+\beta}{2}$. We have

$$\forall \ell+1 \le j \le k, \quad \mathrm{E}\left[Y_i\right] = \frac{\alpha+\beta}{2} \log q_{v_i} \ge \beta \log q_{v_i} + \frac{\alpha-\beta}{3} \log q_{v_i}.$$

Again, by Hoeffding's inequality, we have

$$\Pr_{\mathcal{D}}\left[\sum_{i=1}^{k} Y_i < \beta \sum_{i=1}^{k} \log q_{v_i}\right] \le \exp\left(-\frac{(\alpha-\beta)^3}{23} \sum_{i=1}^{k} \log q_{v_i}\right). \tag{25}$$

Combining (24) and (25) we have

$$\Pr_{\mathcal{D}}\left[B_c\right] \le 2\exp\left(-\frac{(\alpha-\beta)^3}{23} \sum_{i=1}^{k} \log q_{v_i}\right) \overset{(\star)}{\le} 2\exp\left(-\frac{25}{23}\log D - 3\right) \le 2\exp\left(-\frac{25}{23}\ln D - 3\right) \le \frac{1}{\mathrm{e}(D+1)},$$

where $(\star)$ holds because $\sum_{i=1}^{k} \log q_{v_i} \ge \frac{25}{(\alpha-\beta)^3}(\log D + 3)$. By Lovász local lemma, there exists a set of marked variables $\mathcal{M} \subseteq S_{\text{small}}$ such that the condition in (20) is satisfied.

Similar to the proof of Theorem 3.9, we can use Moser-Tardos algorithm [MT10] to construct such projection scheme. With probability at least $1-\delta$, the algorithm constructs a projection scheme in time $O(nDk \log \frac{1}{\delta})$, where $k = \max_{c \in C} |\mathsf{vbl}(c)|$.

We now combine all the steps together. The construction of the data structures $\mathcal{S}_v$ for all $v \in V$ has the cost $O(nq \log q)$. Computing the s_v for all $v \in S_{\text{large}}$ has the costs $O(n \log q)$. Computing the marked set $\mathcal{M} \subseteq S_{\text{small}}$ has the cost $O(nDk \log \frac{1}{\delta})$. The total cost is $O(n(Dk+q) \log \frac{1}{\delta} \log q)$. □

7. Analysis of the Inverse Sampling subroutine

In this section, we prove Lemma 5.3. Let $\Phi = (V, \mathcal{Q}, C)$ be a CSP formula, where each variable v takes value in Q_v. Let $\boldsymbol{h} = (h_v)_{v \in V}$ be a balanced projection scheme satisfying Condition 3.4 with parameters α and β, where for each $v \in V$, $h_v : Q_v \to \Sigma_v$, $|Q_v| = q_v$ and $|\Sigma_v| = s_v$. Let $(Y_t)_{t \ge 0}$ denote random sequence generated by Algorithm 1, where $Y_t \in \Sigma$ is the random Y after the t-th iteration of the for-loop. Recall that for each $1 \le t \le T+1$, we have defined the following bad events:

- $\mathcal{B}_t^{(1)}$: in the t-th call of $\mathsf{InvSample}(\cdot)$, the random assignment $\boldsymbol{X}$ is returned in Line 10;
- $\mathcal{B}_t^{(2)}$: in the t-th call of $\mathsf{InvSample}(\cdot)$, the random assignment $\boldsymbol{X}$ is returned in Line 4.

In the t-th calling of the subroutine $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, y_\Lambda, S)$ (Algorithm 2), conditional on $\neg\mathcal{B}_t^{(1)} \wedge \neg\mathcal{B}_t^{(2)}$, all the connected components that intersect with S are small, and the rejection sampling on each component succeeds. It is straightforward to verify the subroutine returns a perfect sample from $\mu_S^{y_\Lambda}$.

Next, we analyze the running time of the subroutine $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, y_\Lambda, S)$. Let $G = (C, E)$ denote the dependency graph of $\Phi = (V, \boldsymbol{Q}, C)$. We assume the dependency graph is stored in an adjacent list. We can construct such adjacent list at the beginning of the whole algorithm. The cost of construction is $O(nDk)$, which is dominated by the cost in Theorem 5.1.

Assume that the algorithm can access a projection oracle with query cost $O(\log q)$. The first step of the subroutine is to find all the connected components that intersect with set S. For each variable $v \in S$, we find all the constraints $C(v) = \{c \in C \mid v \in \mathsf{vbl}(c)\}$ (note that $|C(v)| \le D$), then perform a deep first search (DFS) in G starting from $C(v)$. During the DFS, suppose the current constraint is $c \in C$. We can find the unique configuration $\sigma \in \mathcal{Q}_{\mathsf{vbl}(c)}$ forbidden by c, i.e. $c(\sigma) = \mathtt{False}$. We call the projection oracle to obtain $\tau \in \Sigma_{\mathsf{vbl}(c)}$, where $\tau_v = h_v(\sigma_v)$ for each $v \in V$. The cost of this step is $O(k \log q)$. If for all $v \in \Lambda \cap \mathsf{vbl}(c)$, $y_\Lambda(v) = \tau_v$ (which means c is not satisfied by y_Λ), we do DFS recursively starting from c; otherwise, we stop current DFS branch and remove c from the graph G. If the size of current connected component is greater than $2D \log \frac{nD}{\delta}$, the connected component is too large, we stop the whole DFS process. The total cost of DFS is

$$T_{\mathrm{DFS}} = O\left(|S|\, D^2 k \log \frac{nD}{\delta} \log q\right).$$

Another cost of the subroutine comes from the rejection sampling from Line 5 to Line 10. To perform the rejection sampling, for each variable v, we either draw X_v from $\pi_v^{y_v}$ or draw X_v from the π_v. This step can be achieved by calling oracles. The cost is $O(\log q)$. Since there are at most $|S|$ connected components and each of the size at most $2D \log \frac{nD}{\delta}$, the total number of variables is $O(|S|\, Dk \log \frac{nD}{\delta})$. For each component, the algorithm uses the rejection sampling for at most $R = \left\lceil 10 \left(\frac{n}{\delta}\right)^\eta \log \frac{n}{\delta} \right\rceil$ times. The total cost of rejection sampling is

$$T_{\mathrm{rej}} = O\left(|S|\, Dk \left(\frac{n}{\delta}\right)^\eta \log^2 \left(\frac{nD}{\delta}\right) \log q\right).$$

The total cost of the subroutine is

$$T_{\mathrm{DFS}} + T_{\mathrm{rej}} = O\left(|S|\, D^2 k \left(\frac{n}{\delta}\right)^\eta \log^2 \left(\frac{nD}{\delta}\right) \log q\right).$$

Finally, we use the following lemma to bound the probabilities of the bad events $\mathcal{B}_t^{(1)}$ and $\mathcal{B}_t^{(2)}$.

Lemma 7.1. *Let $\Phi = (V, \boldsymbol{Q}, C)$ be the input CSP formula and $\boldsymbol{h}$ a projection scheme satisfying Condition 3.4 with parameters α and β. Let D denote the maximum degree of the dependency graph of Φ. Let $p = \max_{c \in C} \prod_{v \in \mathsf{vbl}(c)} \frac{1}{|Q_v|}$. Let $0 < \eta < 1$ be a parameter. Suppose $\log \frac{1}{p} \ge \frac{1}{1-\alpha} \log(20D^2)$ and $\log \frac{1}{p} \ge \frac{1}{\beta} \log \left(\frac{40\mathrm{e}D^2}{\eta}\right)$. The subroutine $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, y_\Lambda, S)$ in Algorithm 2 with parameter η satisfies that for any $1 \le t \le T+1$,*

$$\Pr\left[\mathcal{B}_t^{(1)}\right] \le \delta \quad \text{and} \quad \Pr\left[\mathcal{B}_t^{(2)}\right] \le \delta.$$

The rest of this section is dedicated to the proof of Lemma 7.1. Let $v_i \in V$ denote the random variable picked by Algorithm 1 in the i-th iteration of the for-loop. In the proof of Lemma 7.1, we always fix a $1 \le t \le T+1$ and a sequence $v_1, v_2, \ldots, v_T$. Hence, we always consider the probability space generated by Algorithm 1 conditional on v_i is picked in the i-th iteration of the for-loop.

Define (possibly partial) projected configuration

$$Y = y_\Lambda \triangleq \begin{cases} Y_{t-1}(V \setminus \{v_t\}) & \text{if } 1 \le t \le T; \\ Y_T & \text{if } t = T+1, \end{cases} \tag{26}$$

where $\Lambda = V \setminus \{v_t\}$ if $1 \le t \le T$, and $\Lambda = V$ if $t = T+1$. We analyze $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, Y, S)$, where

$$S = \begin{cases} \{v_t\} & \text{if } 1 \le t \le T; \\ V & \text{if } t = T+1. \end{cases}$$

7.1. Analysis of rejection sampling (bound $\Pr[\mathcal{B}_t^{(1)}]$).

We first prove that

$$\Pr\left[\mathcal{B}_t^{(1)}\right] \le \delta. \tag{27}$$

Let $\Phi' = (V, \mathcal{Q}, C')$ denote the CSP formula obtained from $\Phi = (V, \mathcal{Q}, C)$ by removing constraints satisfied by Y. Let $H' = H_{\Phi'} = (V, \mathcal{E}')$ denote the hypergraph modeling Φ', where $\mathcal{E}' = \{\mathsf{vbl}(c) \mid c \in C'\}$ is a multi-set. Suppose H'_Φ has ℓ connected components $H'_1, H'_2, \ldots, H'_\ell$ that intersect with S, where $H'_i = (V_i, \mathcal{E}'_i)$ and $V_i \cap S \neq \varnothing$ for all $1 \le i \le \ell$. Let $\Phi'_i = (V_i, \mathcal{Q}_{V_i}, C'_i)$ denote the CSP formula represented by H'_i, where C'_i denotes the set of constraints represented by $\mathcal{E}'_i$.

Fix an integer $1 \le i \le \ell$. Lines 6 – 8 in Algorithm 2 actually run rejection sampling on $\widetilde{\Phi}_i = (V_i, \widetilde{\mathcal{Q}}_{V_i}, C'_i)$, where each $\widetilde{Q}_v \subseteq Q_v$, such that

$$\forall v \in V_i, \quad \widetilde{Q}_v \triangleq \begin{cases} h_v^{-1}(Y_v) & \text{if } v \in V_i \cap \Lambda; \\ Q_v & \text{if } v \in V_i \setminus \Lambda. \end{cases}$$

Since the maximum degree of the dependency graph of Φ is D, the maximum degree of the dependency graph of $\widetilde{\Phi}_i$ is at most D. Let $\mathcal{D}$ denote the product distribution such that each $v \in V_i$ samples a value from $\widetilde{Q}_v$ uniformly at random. For each constraint $c \in C'_i$, let B_c denote the bad event that c is not satisfied. Note that $\boldsymbol{h}$ is a balanced projection scheme. By the definition of $\widetilde{\mathcal{Q}}_{V_i}$, it holds that $|\widetilde{Q}_v| \ge \lfloor q_v/s_v \rfloor$ for all $v \in V_i$, where $q_v = |Q_v|$. In other words, $\widetilde{\Phi}_i$ is the conditional LLL instance in Condition 3.7. By Condition 3.4, we have for each $c \in C'_i$,

$$\Pr_{\mathcal{D}}[B_c] = \prod_{v \in \mathsf{vbl}(c)} \frac{1}{\left|\widetilde{Q}_v\right|} \le \prod_{v \in \mathsf{vbl}(c)} \frac{1}{\lfloor q_v/s_v \rfloor} \le \left(\prod_{v \in \mathsf{vbl}(c)} \frac{1}{q_v}\right)^\beta,$$

Recall that in Lemma 7.1, we assume that for each $c \in C$, $\sum_{v \in \mathsf{vbl}(c)} \log q_v \ge \frac{1}{\beta} \log\left(\frac{40\mathrm{e}D^2}{\eta}\right)$ for $0 < \eta < 1$. Note that $C'_i \subseteq C$, we have for each $c \in C'_i$,

$$\Pr_{\mathcal{D}}[B_c] \le \frac{\eta}{40\mathrm{e}D^2}.$$

For each B_c, define $x(B_c) = \frac{\eta}{40D^2}$. We have

$$\begin{aligned}\Pr_{\mathcal{D}}[B_c] &\le \frac{\eta}{40\mathrm{e}D^2} \le \frac{\eta}{40D^2}\left(1 - \frac{\eta}{40D^2}\right)^{\frac{40D^2}{\eta}-1} \le \frac{\eta}{40D^2}\left(1 - \frac{\eta}{40D^2}\right)^{D} \\ &\le x(B_c) \prod_{B_{c'} \in \Gamma(B_c)} (1 - x(B_{c'})),\end{aligned}$$

where $\Gamma(\cdot)$ is defined as in the Lovász local lemma (Theorem 2.1). Since $B_t^{(1)}$ occurs, it must hold that $\left|C'_i\right| \le 2D \log \frac{nD}{\delta}$. By Lovász local lemma (Theorem 2.1), we have

$$\begin{aligned}\Pr_{\mathcal{D}}\left[\bigwedge_{c \in C'_i} \overline{B_c}\right] &\ge \prod_{c \in C'_i} (1 - x(B_c)) \ge \prod_{c \in C'_i}\left(1 - \frac{\eta}{40D^2}\right) \\ \left(\text{by } \left|C'_i\right| \le 2D \log \frac{nD}{\delta}\right) \qquad &\ge \left(1 - \frac{\eta}{40D^2}\right)^{2D \log \frac{nD}{\delta}} \ge \exp\left(-\frac{\eta}{5D} \log \frac{Dn}{\delta}\right) \\ &= \left(\frac{\delta}{Dn}\right)^{\frac{\eta}{5D \ln 2}} \ge \left(\frac{\delta}{Dn}\right)^{\frac{\eta}{2D}} \ge \frac{1}{2}\left(\frac{\delta}{n}\right)^\eta.\end{aligned}$$

Hence, each trial of the rejection sampling in Lines 6 – 8 succeeds with probability at least $\frac{1}{2}\left(\frac{\delta}{n}\right)^\eta$. Since the algorithm uses rejection sampling independently for $R = \left\lceil 10\left(\frac{n}{\delta}\right)^\eta \log \frac{n}{\delta}\right\rceil$ times, the probability that the rejection sampling fails in one connected component is at most

$$\left(1 - \frac{1}{2}\left(\frac{\delta}{n}\right)^\eta\right)^R \le \exp\left(-\frac{R}{2}\left(\frac{\delta}{n}\right)^\eta\right) \le \frac{\delta}{n}.$$

Since there are at most n connected components, by a union bound,

$$\Pr\left[\mathcal{B}_t^{(1)}\right] \le \delta$$

This proves (27).

7.2. **Analysis of connected component (bound $\Pr[\mathcal{B}_t^{(2)}]$).** We now bound the probability of bad event $\mathcal{B}_t^{(2)}$. Consider the subroutine $\mathsf{InvSample}(\Phi, \boldsymbol{h}, \delta, Y, S)$. Recall $\Phi' = (V, \boldsymbol{Q}, C')$ is the CSP formula obtained from $\Phi = (V, \boldsymbol{Q}, C)$ by removing all the constraints satisfied by Y. Recall hypergraph $H' = H_{\Phi'} = (V, \mathcal{E}')$ models Φ'. Let $H = H_\Phi = (V, \mathcal{E})$ denote the hypergraph modeling Φ, where $\mathcal{E} = \{\mathsf{vbl}(c) \mid c \in C\}$ is a multi-set. For any edge $e \in \mathcal{E}$, we use $\mathcal{B}_e$ to denote the bad event that $e \in \mathcal{E}'$ and the number of hyperedges in the connected component in H' that contains e is at least L, where $L = \lceil 2D \log \frac{Dn}{\delta} \rceil$. By a union bound, we have

$$\Pr\left[\mathcal{B}_t^{(2)}\right] \le \sum_{e \in \mathcal{E}} \Pr\left[\mathcal{B}_e\right].$$

Recall D is the maximum degree of the dependency graph. Since $|\mathcal{E}| \le n(D+1)$, it suffices to prove

$$\Pr\left[\mathcal{B}_e\right] \le \frac{\delta}{n(D+1)}. \tag{28}$$

To bound the probability of $\mathcal{B}_e$, we need the following lemma.

Lemma 7.2. *Let $\Phi = (V, \boldsymbol{Q}, C)$ be a CSP formula. Let $\boldsymbol{h}$ be the projection scheme satisfying Condition 3.4 with parameters α and β. Let $q_v = |Q_v|$ and D denote the maximum degree of the dependency graph of Φ. If for any constraint $c \in C$,*

$$\sum_{v \in \mathsf{vbl}(c)} \log q_v \ge \frac{1}{\beta} \log(40\mathrm{e}D^2),$$

then for any subset $H \subseteq \Lambda$, any projected configuration $\sigma \in \Sigma_H = \bigotimes_{v \in H} \Sigma_v$,

$$\Pr\left[Y_H = \sigma\right] \le \exp\left(\sum_{u \in H} \frac{1}{20D}\right) \prod_{v \in H} \left(\frac{1}{q_v} \left\lceil \frac{q_v}{s_v} \right\rceil\right),$$

where $Y \in \Sigma_\Lambda$ is defined in (26).

The proof of Lemma 7.2 is deferred to Section 7.3. Next, we introduce the following definitions of line graph and 2-tree.

Definition 7.3 (line graph)**.** Let $H = (V, \mathcal{E})$ be a hypergraph. The line graph $\mathrm{Lin}(H)$ is a graph such that each vertex represents a hyperedge in $\mathcal{E}$, two vertices $e, e' \in \mathcal{E}$ are adjacent iff $e \cap e' \neq \emptyset$.

Definition 7.4 (2-tree)**.** Let $G = (V, E)$ be a graph. A subset of vertices $S_{\mathrm{tree}} \subseteq V$ is a 2-tree if (1) for any $u, v \in S_{\mathrm{tree}}$, their distance $\mathrm{dist}_G(u, v)$ in graph G is at least 2; (2) if one adds an edge between $u, v \in S_{\mathrm{tree}}$ such that $\mathrm{dist}_G(u, v) = 2$, then S_{tree} becomes connected.

The following two propositions are proved in the full version [FGYZ19] of [FGYZ20].

Proposition 7.5 ([FGYZ19, Corollary 5.7])**.** *Let $G = (V, E)$ be a graph with maximum degree Δ and $v \in V$ a vertex. The number of 2-trees in graph G of size ℓ containing vertex v is at most $\frac{(\mathrm{e}\Delta^2)^{\ell-1}}{2}$.*

Proposition 7.6 ([FGYZ19, Lemma 5.8])**.** *Let $H = (V, \mathcal{E})$ be hypergraph. Let $\mathrm{Lin}(H)$ denote the line graph of H. Let $B \subseteq \mathcal{E}$ be a subset of hyperedges that induces a connected subgraph in $\mathrm{Lin}(H)$ and $e \in B$ an arbitrary hyperedge. There exists a 2-tree $S_{\mathrm{tree}} \subseteq \mathcal{E}$ in $\mathrm{Lin}(H)$ such that $e \in S_{\mathrm{tree}}$ and $|S_{\mathrm{tree}}| = \left\lfloor \frac{|B|}{D+1} \right\rfloor$, where D is the maximum degree of the line graph $\mathrm{Lin}(H)$.*

Suppose $\boldsymbol{h}$ satisfies Condition 3.4. Recall $Y \in \Sigma_\Lambda$, where $\Lambda = V \setminus \{v_t\}$ for $1 \le t \le T$ and $\Lambda = V$ for $t = T+1$. We say an edge $e \in \mathcal{E}$ is bad if e is not satisfied by Y. Suppose e represents the constraint c such that $c(\boldsymbol{x}) = \texttt{False}$ for a unique configuration $\boldsymbol{x} \in \mathcal{Q}_e$. Given the projected configuration $Y \in \Sigma_\Lambda$, we have

$$e \text{ is bad} \iff \forall u \in \Lambda \cap e, Y_u \neq h_u(\boldsymbol{x}_u). \tag{29}$$

In other words, if e is bad, then the constraint corresponding to c in the "round-down" CSP formula (Definition 3.5) is not satisfied by Y. If $\mathcal{B}_e$ occurs, there must exist a connected component $B \subseteq \mathcal{E}$ in line graph $\mathrm{Lin}(H)$ such that $e \in B$ and all hyperedges in B are bad and $|B| = L$, where $L = \lceil 2D \log \frac{Dn}{\delta} \rceil$ and D is the maximum degree of the dependency graph of the input formula. By Proposition 7.6, there must exist a 2-tree S_{tree} in $\mathrm{Lin}(H)$ with size $\ell = \lfloor \frac{L}{D+1} \rfloor$ such that $e \in S_{\mathrm{tree}}$ and all edges in S_{tree} are bad. Fix such a 2-tree S_{tree}. By definition, each vertex in S_{tree} is a hyperedge $e \in \mathcal{E}$, and for all $e, e' \in S_{\mathrm{tree}}$, $e \cap e' = \emptyset$. Let $S'_{\mathrm{tree}} \subseteq S_{\mathrm{tree}}$ denote the subset of edges $e \in S_{\mathrm{tree}}$ such that $e \subseteq \Lambda$. Since Y is a random projected configuration, by (29), we have

$$\begin{aligned}\Pr\left[\forall e \in S_{\mathrm{tree}}, e \text{ is bad}\right] &= \Pr\left[\forall e \in S_{\mathrm{tree}}, \forall u \in e \cap \Lambda, Y_u \neq h_u(\boldsymbol{x}_u)\right] \\ &\le \Pr\left[\forall e \in S'_{\mathrm{tree}}, \forall u \in e, Y_u \neq h_u(\boldsymbol{x}_u)\right].\end{aligned}$$

Fix an edge $e \in S'_{\mathrm{tree}}$. By Condition 3.4 and the condition $\sum_{v \in e} \log q_v \ge \frac{1}{1-\alpha} \log(20D^2)$ assumed in Lemma 7.1, it holds that

$$\prod_{v \in e} \frac{1}{q_v} \left\lceil \frac{q_v}{s_v} \right\rceil \le \left(\prod_{v \in e} \frac{1}{q_v} \right)^{1-\alpha} \le \frac{1}{20D^2}.$$

Note that if $s_v = 1$, then $\frac{1}{q_v}\left\lceil \frac{q_v}{s_v} \right\rceil = 1$. For any $v \in e$ such that $s_v > 1$(thus $q_v \ge s_v > 1$), we have $\frac{1}{q_v}\left\lceil \frac{q_v}{s_v} \right\rceil \le \frac{1}{q_v}\left\lceil \frac{q_v}{2} \right\rceil \le \frac{2}{3}$. Let $r = \log_{2/3} \frac{1}{20D^2} + 1$. We can find a subset of variables $R(e) \subseteq e$ such that

$$\prod_{v \in R(e)} \frac{1}{q_v} \left\lceil \frac{q_v}{s_v} \right\rceil \le \frac{1}{20D^2}, \quad \text{and} \quad |R(e)| \le r.$$

Note that Lemma 7.1 assumes that $\sum_{v \in \mathsf{vbl}(c)} \log q_v \ge \frac{1}{\beta} \log\left(\frac{40\mathrm{e}D^2}{\eta}\right) \ge \frac{1}{\beta} \log(40\mathrm{e}D^2)$. We use Lemma 7.2 on subset $H = \cup_{e \in S'_{\mathrm{tree}}} R(e)$. Note that all hyperedges in S'_{tree} are disjoint. We have

$$\begin{aligned}\Pr\left[\forall e \in S_{\mathrm{tree}}, e \text{ is bad}\right] &\le \Pr\left[\forall e \in S'_{\mathrm{tree}}, \forall u \in R(e), Y_u \neq h_u(\boldsymbol{x}_u)\right] \le \Pr\left[\forall u \in H, Y_u \neq h_u(\boldsymbol{x}_u)\right] \\ &\le \prod_{e \in S'_{\mathrm{tree}}} \prod_{v \in R(e)} \left(\frac{1}{q_v} \left\lceil \frac{q_v}{s_v} \right\rceil \exp\left(\frac{1}{20D} \right) \right) \le \prod_{e \in S'_{\mathrm{tree}}} \left(\frac{1}{20D^2} \exp\left(\frac{r}{20D} \right) \right) \\ \left(\text{by } r = \log_{2/3} \frac{1}{20D^2} + 1\right) \quad &\le \prod_{e \in S'_{\mathrm{tree}}} \left(\frac{1}{12D^2} \right).\end{aligned}$$

Since $|\Lambda| \ge n-1$ and all hyperedges in S_{tree} are disjoint, $\left|S'_{\mathrm{tree}}\right| \ge |S_{\mathrm{tree}}| - 1 = \ell - 1$. We have

$$\Pr\left[\forall e \in S_{\mathrm{tree}}, e \text{ is bad}\right] \le \left(\frac{1}{12D^2} \right)^{\ell-1}.$$

Note that the maximum degree of line graph is at most D. By Proposition 7.5, we have

$$\Pr\left[\mathcal{B}_e\right] \le \frac{1}{2}\left(\mathrm{e}D^2\right)^{\ell-1} \left(\frac{1}{12D^2} \right)^{\ell-1} \le \frac{1}{2}\left(\frac{1}{4}\right)^{\ell-1} \le \left(\frac{1}{2}\right)^{2\ell-1}.$$

Note that $\ell = \lfloor L/(D+1) \rfloor$ and $L = \left\lceil 2D \log \frac{nD}{\delta} \right\rceil$. We have $\ell \ge \log \frac{nD}{\delta} - 1$. We may assume $nD \ge 16$. Otherwise, the sampling problem is trivial. The inequality (28) can be proved by

$$\Pr\left[\mathcal{B}_e\right] \le \left(\frac{1}{2}\right)^{2\log\frac{nD}{\delta} - 3} \le \frac{\delta}{n(D+1)}.$$

7.3. **Proof of Lemma 7.2.** We now prove (Lemma 7.2). We use the following lemma to prove it.

Lemma 7.7. *Let $\Phi = (V, \boldsymbol{Q}, C)$ be a CSP formula. Let $\boldsymbol{h} = (h_v)_{v\in V}$ be the projection scheme satisfying Condition 3.4 with parameters α and β. Let D denote the maximum degree of the dependency graph of Φ. Let $q_v = |Q_v|$. Suppose for any constraint $c \in C$, it holds that*

$$\sum_{v\in\mathsf{vbl}(c)} \log q_v \geq \frac{1}{\beta}\log(40\mathrm{e}D^2).$$

Fix a variable $u \in V$ and a partial projected configuration $\tau \in \Sigma_{V\setminus\{u\}}$. For any $y \in \Sigma_u$, it holds that

$$\nu_u^\tau(y) \leq \frac{1}{q_u}\left\lceil\frac{q_u}{s_u}\right\rceil\exp\left(\frac{1}{20D}\right).$$

Proof. Define a new CSP formula $\widehat{\Phi} = (V, \widehat{\boldsymbol{Q}} = (\widehat{Q}_v)_{v\in V}, C)$ by

$$\forall w \in V, \quad \widehat{Q}_w = \begin{cases} h_w^{-1}(\tau_w) & \text{if } w \neq u \\ Q_w & \text{if } w = u. \end{cases}$$

Let $\mathcal{D}$ denote the product distribution that each $w \in V$ takes a value from $\widehat{Q}_w$ uniformly and independently. For each constraint $c \in C$, define a bad event B_c as c is not satisfied. Let $\mathcal{B} = (B_c)_{c\in C}$ be the collection of bad events. Recall that $\Gamma(\cdot)$ is defined as in the Lovász local lemma (Theorem 2.1). It holds that $\max_{c\in C}|\Gamma(B_c)| \leq D$. For each B_c, let $x(B_c) = \frac{1}{40D^2}$. By Condition 3.4, it holds that

$$\begin{aligned}\Pr_{\mathcal{D}}[B_c \text{ is not satisfied}] &= \prod_{v\in\mathsf{vbl}(c)}\frac{1}{\left|\widehat{Q}_v\right|} \leq \prod_{v\in\mathsf{vbl}(c)}\frac{1}{\lfloor q_v/s_v\rfloor} \leq \left(\prod_{v\in\mathsf{vbl}(c)}\frac{1}{q_v}\right)^\beta \\ &\leq \frac{1}{40\mathrm{e}D^2} \leq \frac{1}{40D^2}\left(1-\frac{1}{40D^2}\right)^{40D^2-1} \\ &\leq \frac{1}{40D^2}\left(1-\frac{1}{40D^2}\right)^D \leq x(B_c)\prod_{B_{c'}\in\Gamma(B_c)}(1-x(B_{c'})).\end{aligned}$$

Fix $y \in \Sigma_u$. Let A denote the event that the value of u belongs to $h_u^{-1}(y)$, then $|\Gamma(A)| \leq D$, where $\Gamma(A) \subseteq \mathcal{B}$ is the set of bad events B such that $u \in \mathsf{vbl}(B)$. Let $\widehat{\mu}$ denote the uniform distribution of all satisfying assignments to $\widehat{\Phi}$. By Theorem 2.1, we have

$$\nu_u^\tau(y) = \Pr_{\widehat{\mu}}[A] = \Pr_{X\sim\widehat{\mu}}\left[X_u \in h_u^{-1}(y)\right] \leq \frac{1}{q_u}\left\lceil\frac{q_u}{s_u}\right\rceil\left(1-\frac{1}{40D^2}\right)^{-D} \leq \frac{1}{q_u}\left\lceil\frac{q_u}{s_u}\right\rceil\exp\left(\frac{1}{20D}\right). \qquad \square$$

Now we are ready to prove Lemma 7.2.

Proof of Lemma 7.2. Fix a subset $H \subseteq V$, and an projected configuration $\sigma \in \Sigma_H$. Recall $1 \leq t \leq T+1$ is a fixed integer. Recall $Y = Y_{t-1}(\Lambda)$, where $\Lambda = V \setminus \{v_t\}$ if $1 \leq t \leq T$, and $\Lambda = V$ if $t = T+1$. Recall that $v_1, v_2, \ldots, v_t \in V$ is a sequence such that v_i is the variable picked by Algorithm 1 in i-th iteration of the for-loop.

For any variable $u \in H$, let $t(u)$ denote the last step up to step t such that u is picked by Algorithm 1 of the for-loop. Formally, if u appears in the sequence $v_1, v_2, \ldots, v_t$, then $t(u)$ is the largest number such that $v_{t(u)} = u$; if u does not appear in the sequence $v_1, v_2, \ldots, v_t$, then $t(u) = 0$. We list all variables in H as $u_1, u_2, \ldots, u_{|H|}$ such that $t(u_1) \leq t(u_2) \leq \ldots \leq t(u_{|H|})$, where for these variables u satisfying $t(u) = 0$, we break tie arbitrarily. Thus, $Y_t(u) = Y_{t(u)}(u)$ for all $u \in H$. We have

$$\Pr[Y_H = \sigma] = \Pr\left[\forall u_i \in H, Y_{u_i} = \sigma_{u_i}\right] \leq \prod_{i=1}^{|H|}\Pr\left[Y_{t(u_i)}(u_i) = \sigma_{u_i} \mid \forall j < i, Y_{t(u_j)}(u_j) = \sigma_{u_j}\right].$$

We now only need to prove that, for any $1 \leq i \leq |H|$,

$$\Pr\left[Y_{t(u_i)}(u_i) = \sigma_{u_i} \mid \forall j < i, Y_{t(u_j)}(u_j) = \sigma_{u_j}\right] \leq \frac{1}{q_{u_i}}\left\lceil\frac{q_{u_i}}{s_{u_i}}\right\rceil\exp\left(\frac{1}{20D}\right). \tag{30}$$

Suppose $t(u_i) = 0$, then $Y_0(u_i) \in \Sigma_{u_i}$ is sampled independently with $\Pr\left[Y_0(u_i) = \sigma_{u_i}\right] = \frac{\left|h_{u_i}^{-1}(\sigma_{u_i})\right|}{q_{u_i}}$. Since $\boldsymbol{h}$ is balanced, we have $\left|h_{u_i}^{-1}(\sigma_{u_i})\right| \leq \left\lceil \frac{q_{u_i}}{s_{u_i}} \right\rceil$. Inequality (30) holds because

$$\Pr\left[Y_0(u_i) = \sigma_{u_i} \mid \forall j < i, Y_0(u_j) = \sigma_{u_j}\right] \leq \frac{1}{q_{u_i}} \left\lceil \frac{q_{u_i}}{s_{u_i}} \right\rceil.$$

Suppose $t(u_i) = \ell \neq 0$. Algorithm 1 uses the subroutine $\mathsf{InvSample}(\cdot)$ to sample a random $X_v \in Q_v$ in Line 4, then maps X_v into $Y_\ell(u_i)$ in Line 5. If X_v is returned in Line 4 or Line 10 in Algorithm 2, then X_v is uniformly distribution over Q_{u_i}. In this case, inequality (30) holds because

$$\Pr\left[Y_\ell(u_i) = \sigma_{u_i} \mid \forall j < i, Y_{t(u_j)}(u_j) = \sigma_{u_j}\right] = \sum_{X_v \in h_{u_i}^{-1}(\sigma_{u_i})} \frac{1}{q_{u_i}} \leq \frac{1}{q_{u_i}} \left\lceil \frac{q_{u_i}}{s_{u_i}} \right\rceil.$$

Otherwise, X_v is returned in Line 11 of Algorithm 2. In this case, $Y_\ell(u_i)$ is sampled from the distribution $\nu_{u_i}^{Y_{\ell-1}(V \setminus \{u_i\})}$. We can use Lemma 7.7 with $\tau = Y_{\ell-1}(V \setminus \{u_i\})$ and $u = u_i$. Note that Lemma 7.7 holds for any τ and u. We have

$$\Pr\left[Y_\ell(u_i) = \sigma_{u_i} \mid \forall j < i, Y_{t(u_j)}(u_j) = \sigma_{u_j}\right] = \nu_{u_i}^{Y_{\ell-1}(V \setminus \{u_i\})}(\sigma_{u_i}) \leq \frac{1}{q_{u_i}} \left\lceil \frac{q_{u_i}}{s_{u_i}} \right\rceil \exp\left(\frac{1}{20D}\right).$$

Thus, inequality (30) holds. □

8. Proof of rapid mixing

Let $\Phi = (V, \boldsymbol{Q}, C)$ be a CSP formula with atomic constraints and $\boldsymbol{h} = (h_v)_{v \in V}$ be a balanced projection scheme satisfying Condition 3.4 with parameter α and β, where $h_v : Q_v \to \Sigma_v$. Let $\nu = \nu_{\Phi, \boldsymbol{h}}$ be the projected distribution over $\Sigma = \bigotimes_{v \in V} \Sigma_v$ in Definition 3.2. Let $(Y_t)_{t \geq 0}$ denote the Glauber dynamics P_{Glauber} on ν. In this section, we show that the Glauber dynamics P_{Glauber} is rapid mixing, and prove Lemma 5.6 and Lemma 5.2.

8.1. The stationary distribution.

We first proves that ν is the unique stationary distribution.

Proposition 8.1. *Let $\Phi = (V, \boldsymbol{Q}, C)$ be a CSP formula with atomic constraints. Let $\boldsymbol{h} = (h_v)_{v \in V}$ be the projection scheme satisfying Condition 3.4 with parameters α and β. Let $q_v = |Q_v|$, $p = \max_{c \in C} \prod_{v \in \mathsf{vbl}(c)} \frac{1}{q_v}$ and D denote the maximum degree of the dependency graph of Φ. Suppose $\log \frac{1}{p} \geq \frac{1}{\beta} \log(2\mathrm{e}D)$. The Glauber dynamics P_{Glauber} is irreducible, aperiodic and reversible with respect to ν, thus it has the unique stationary distribution ν.*

Proof. By the transition rule of Glauber dynamics, it is easy to verify the Glauber dynamics is aperiodic and reversible with respect to ν. We prove the Markov chain is irreducible. We show that for any $\sigma \in \Sigma$, $\nu(\sigma) > 0$. This implies that the transition probability of Glauber dynamics is always well-defined and the Markov chain is connected. Fix a $\sigma \in \Sigma$. Define a new instance $\widehat{\Phi} = (V, \widehat{\boldsymbol{Q}} = (\widehat{Q}_v)_{v \in V}, C)$ as $\widehat{Q}_v = h_v^{-1}(\sigma_v)$ for all $v \in V$. It suffices to show that $\widehat{\Phi}$ is satisfiable, which implies $\nu(\sigma) > 0$. The maximum degree of dependency graph of $\widehat{\Phi}$ is at most D. Besides, if each variable picks a value from $\widehat{Q}_v$ uniformly and independently, then for each $c \in C$, the probability that c is not satisfied is at most

$$\prod_{v \in \mathsf{vbl}(c)} \frac{1}{|\widehat{Q}_v|} \leq \prod_{v \in \mathsf{vbl}(c)} \frac{1}{\lfloor q_v / s_v \rfloor} \leq \left(\prod_{v \in \mathsf{vbl}(c)} \frac{1}{q_v} \right)^{\beta} \leq \frac{1}{2\mathrm{e}D}.$$

By Lovász local lemma, $\widehat{\Phi}$ is satisfiable. □

8.2. **Path coupling analysis.** We use the path coupling [BD97] to show that the Markov chain is rapid mixing. Fix two projected configurations $\boldsymbol{X}, \boldsymbol{Y} \in \Sigma = \bigotimes_{v \in V} \Sigma_v$ such that $\boldsymbol{X}$ and $\boldsymbol{Y}$ disagree only at one variable $v_0 \in V$ (assume $s_{v_0} \geq 2$). We construct a coupling $(\boldsymbol{X}, \boldsymbol{Y}) \to (\boldsymbol{X}', \boldsymbol{Y}')$ such that $\boldsymbol{X} \to \boldsymbol{X}'$ and $\boldsymbol{Y} \to \boldsymbol{Y}'$ each individually follows the transition rule of P_{Glauber} such that

$$\mathrm{E}\left[d_{\text{ham}}(\boldsymbol{X}', \boldsymbol{Y}') \mid \boldsymbol{X}, \boldsymbol{Y}\right] \leq 1 - \frac{1}{2n}, \tag{31}$$

where $d_{\text{ham}}(\boldsymbol{X}', \boldsymbol{Y}') \triangleq |\{v \in V \mid X'_v \neq Y'_v\}|$ denotes the Hamming distance between $\boldsymbol{X}'$ and $\boldsymbol{Y}'$. Note that the Hamming distance is at most n. Thus, by path coupling lemma (Lemma 2.3), for any $0 < \varepsilon < 1$,

$$T_{\mathsf{mix}}(\varepsilon) \leq \left\lceil 2n \log \frac{n}{\varepsilon} \right\rceil,$$

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

The coupling $(\boldsymbol{X}, \boldsymbol{Y}) \to (\boldsymbol{X}', \boldsymbol{Y}')$ is constructed as follows.

- Pick the same variable $v \in V$ uniformly at random, set $X'_u \leftarrow X_u$ and $Y'_u \leftarrow Y_u$ for all $u \neq v$.
- Sample (X'_v, Y'_v) jointly from the optimal coupling between $\nu_v^{X_{V \setminus \{v\}}}$ and $\nu_v^{Y_{V \setminus \{v\}}}$.

By the linearity of expectation, we have

$$\begin{aligned} \mathrm{E}\left[d_{\text{ham}}(\boldsymbol{X}', \boldsymbol{Y}') \mid \boldsymbol{X}, \boldsymbol{Y}\right] &= \sum_{v \in V} \Pr\left[X'_v \neq Y'_v \mid \boldsymbol{X}, \boldsymbol{Y}\right] \\ \text{(by the optimal coupling)} \qquad &= \frac{1}{n} \sum_{v \in V \setminus \{v_0\}} d_{\mathrm{TV}}\left(\nu_v^{X_{V \setminus \{v\}}}, \nu_v^{Y_{V \setminus \{v\}}}\right) + \left(1 - \frac{1}{n}\right), \end{aligned}$$

where the last equation holds because $d_{\mathrm{TV}}\left(\nu_{v_0}^{X_{V \setminus \{v_0\}}}, \nu_{v_0}^{Y_{V \setminus \{v_0\}}}\right) = 0$. To prove (31), it suffices to prove

$$\sum_{v \in V \setminus \{v_0\}} d_{\mathrm{TV}}\left(\nu_v^{X_{V \setminus \{v\}}}, \nu_v^{Y_{V \setminus \{v\}}}\right) \leq \frac{1}{2}.$$

To prove the above inequality, we need to bound $d_{\mathrm{TV}}\left(\nu_v^{X_{V \setminus \{v\}}}, \nu_v^{Y_{V \setminus \{v\}}}\right)$ for each $v \in V \setminus \{v_0\}$. We use the coupling introduced by Moitra [Moi19] to do this task. For k-uniform CSP formula such that the domain of each variable is $[q]$, we construct an adaptive version [GLLZ19] of Moitra's coupling. Compared with the analysis in [GLLZ19, FGYZ20], this coupling is more refined and requires a more careful analysis. This part in given in Section 8.3. For general CSP formula, we use the original non-adaptive version of Moitra's coupling. The analysis for general case is much more involved, because we need to deal with arbitrary domain and arbitrary size of constraints. This part is given in Section 8.4.

8.3. **Adaptive coupling analysis.** We first analyze the simple case. Suppose the original input CSP formula of Algorithm 1 is a (k, d)-CSP formula $\Phi = (V, [q]^V, C)$ with atomic constraints, where $|\mathsf{vbl}(c)| = k$ for all $c \in C$ and each variable $v \in V$ appears in at most d constraints, on homogeneous domains $Q_v = [q]$ for all $v \in V$. Note that this case covers two applications: hypergraph coloring and k-CNF formula. We prove the following lemma.

Lemma 8.2. *Let $\Phi = (V, [q]^V, C)$ be a (k, d)-CSP formula with atomic constraints. Let $\boldsymbol{h} = (h_v)_{v \in V}$ be the projection scheme for Φ satisfying Condition 3.4 with parameters α and β. If*

$$k \log q \geq \frac{1}{\beta} \log\left(3000 q^2 d^6 k^6\right), \tag{32}$$

then it holds that $\sum_{v \in V \setminus \{v_0\}} d_{\mathrm{TV}}\left(\nu_v^{X_{V \setminus \{v\}}}, \nu_v^{Y_{V \setminus \{v\}}}\right) \leq \frac{1}{2}$.

Recall that for any $\sigma \in \Sigma_\Lambda$, where $\Lambda \subseteq V$, the distribution μ^σ is the distribution of $\boldsymbol{X} \in [q]^V$ such that $\boldsymbol{X}$ is sampled from μ conditional on $\boldsymbol{h}(\boldsymbol{X}_\Lambda) = (h_v(X_v))_{v \in \Lambda} = \sigma$, where μ is the uniform distribution over all satisfying assignments to Φ. We use μ_v^σ to denote the marginal distribution on v projected from μ^σ. For any $v \in V$ and $c \in \Sigma_v$, it holds that

$$\nu_v^{X_{V \setminus \{v\}}}(c) = \sum_{j \in h_v^{-1}(c)} \mu_v^{X_{V \setminus \{v\}}}(j) \quad \text{and} \quad \nu_v^{Y_{V \setminus \{v\}}}(c) = \sum_{j \in h_v^{-1}(c)} \mu_v^{Y_{V \setminus \{v\}}}(j).$$

Note that each h_v is a function from $[q]$ to Σ_v. By triangle inequality, it holds that

$$d_{\mathrm{TV}}\left(\nu_v^{X_{V\setminus\{v\}}}, \nu_v^{Y_{V\setminus\{v\}}}\right) = \frac{1}{2}\sum_{c\in\Sigma_v}\left|\nu_v^{X_{V\setminus\{v\}}}(c) - \nu_v^{Y_{V\setminus\{v\}}}(c)\right|$$

$$\left(\text{by } \biguplus_{c\in\Sigma_v} h_v^{-1}(c) = [q]\right) \quad \le \frac{1}{2}\sum_{j\in[q]}\left|\mu_v^{X_{V\setminus\{v\}}}(j) - \mu_v^{Y_{V\setminus\{v\}}}(j)\right| = d_{\mathrm{TV}}\left(\mu_v^{X_{V\setminus\{v\}}}, \mu_v^{Y_{V\setminus\{v\}}}\right).$$

For any variable $v \in V \setminus \{v_0\}$, define the influence on v caused by v_0 as

$$I_v \triangleq d_{\mathrm{TV}}\left(\mu_v^{X_{V\setminus\{v\}}}, \mu_v^{Y_{V\setminus\{v\}}}\right). \tag{33}$$

To prove the rapid mixing of Glauber dynamics, it suffices to prove that

$$\sum_{v\in V: v\neq v_0} I_v \le \frac{1}{2}. \tag{34}$$

Fix a variable $v_\star \in V$. We will use a coupling $\mathcal{C}_{\mathrm{apt}}$ to bound the influence $I_{v_\star}$. The coupling $\mathcal{C}_{\mathrm{apt}}$ draws two random samples $\boldsymbol{X}^{\mathcal{C}_{\mathrm{apt}}} \sim \mu^{X_{V\setminus\{v_\star\}}}$ and $\boldsymbol{Y}^{\mathcal{C}_{\mathrm{apt}}} \sim \mu^{Y_{V\setminus\{v_\star\}}}$. By coupling lemma (Lemma 2.2), the influence $I_{v_\star}$ can be bounded by

$$I_{v_\star} \le \Pr_{\mathcal{C}_{\mathrm{apt}}}\left[X_{v_\star}^{\mathcal{C}_{\mathrm{apt}}} \neq Y_{v_\star}^{\mathcal{C}_{\mathrm{apt}}}\right]. \tag{35}$$

To describe the coupling $\mathcal{C}_{\mathrm{apt}}$, we first introduce some definitions. Recall $\Phi = (V, [q]^V, C)$ is the original input CSP formula of Algorithm 1. Recall two projected configurations $\boldsymbol{X}, \boldsymbol{Y} \in \Sigma = \bigotimes_{v\in V}\Sigma_v$ differ only at v_0. Define two CSP formulas Φ^X and Φ^Y as follows:

- $\Phi^X = (V, \boldsymbol{Q}^X = (Q_u^X)_{u\in V}, C)$ is a CSP formula such that

$$Q_u^X = \begin{cases} h_u^{-1}(X_u) & \text{if } u \neq v_\star; \\ [q] & \text{if } u = v_\star. \end{cases} \tag{36}$$

- $\Phi^Y = (V, \boldsymbol{Q}^Y = (Q_u^Y)_{u\in V}, C)$ is a CSP formula such that

$$Q_u^Y = \begin{cases} h_u^{-1}(Y_u) & \text{if } u \neq v_\star; \\ [q] & \text{if } u = v_\star. \end{cases}$$

By definition, $(Q_u^X)_{u\in V}$ and $(Q_u^Y)_{u\in V}$ differ only at variable v_0. We then define two distributions

- μ_{Φ^X}: the uniform distribution over all satisfying assignment to Φ^X;
- μ_{Φ^Y}: the uniform distribution over all satisfying assignment to Φ^Y.

It is straightforward to verify $\mu_{\Phi^X} = \mu^{X_{V\setminus\{v_\star\}}}$ and $\mu_{\Phi^Y} = \mu^{Y_{V\setminus\{v_\star\}}}$. For any subset $S \subseteq V$, we use μ_{S,Φ^X} (and μ_{S,Φ^Y}) to denote the marginal distribution on S projected from μ_{Φ^X} (and μ_{Φ^Y}).

Recall that $\Phi = (V, [q]^V, C)$ is the original input CSP formula of Algorithm 1. Recall that $H = (V, \mathcal{E})$ denotes the (multi-)hypergraph that models Φ, where $\mathcal{E} \triangleq \{\mathsf{vbl}(c) \mid c \in C\}$. Note that H also models Φ^X and Φ^Y, because Φ, Φ^X, Φ^Y have the same sets of variables and constraints. We assume that given any hyperedge $e \in \mathcal{E}$, we can find the unique constraint in $c \in C$ represented by e. For each hyperedge $e \in \mathcal{E}$, define the *volume* of e with respect to Φ^X and Φ^Y as

$$\mathrm{Vol}_{\Phi^X}(e) \triangleq \prod_{u\in e}\left|Q_u^X\right| \quad \text{and} \quad \mathrm{Vol}_{\Phi^Y}(e) \triangleq \prod_{u\in e}\left|Q_u^Y\right|.$$

By Condition 3.4 and (32), initially, we have for any hyperedge $e \in \mathcal{E}$,

$$\mathrm{Vol}_{\Phi^X}(e) \ge 3000q^2d^6k^6 \quad \text{and} \quad \mathrm{Vol}_{\Phi^Y}(e) \ge 3000q^2d^6k^6. \tag{37}$$

Let γ be a threshold such that

$$\gamma \triangleq 32\mathrm{e}q^2d^3k^3 \le 3000q^2d^6k^6. \tag{38}$$

Consider an atomic constraint $c \in C$. Let $\sigma \in [q]^{\mathsf{vbl}(c)}$ denote the unique configuration forbidden by c, i.e. $c(\sigma) = \texttt{False}$. The constraint c is said to be satisfied by the value $x_u \in [q]$ of variable u if $u \in \mathsf{vbl}(c)$ and $\sigma_u \neq x_u$. In other words, given the condition that u takes the value x_u, the constraint c

must be satisfied. A constraint c is said to be satisfied by $\tau \in [q]^S$ for some subset $S \subseteq V$ if c is satisfied by some τ_u, where $u \in S \cap \mathsf{vbl}(c)$.

The coupling procedure $\mathcal{C}_{\text{apt}}$ is given in Algorithm 3.

Algorithm 3: The coupling procedure $\mathcal{C}_{\text{apt}}$

Input : CSP formulas $\Phi^X = (V, \boldsymbol{Q}^X = (Q_u^X)_{u\in V}, C)$ and $\Phi^Y = (V, \boldsymbol{Q}^Y = (Q_u^Y)_{u\in V}, C)$, a hypergraph $H = (V, \mathcal{E})$ modeling Φ^X and Φ^Y, two variables $v_0, v_\star \in V$, a threshold parameter γ in (38);

Output : a pair of assignments $\boldsymbol{X}^{\mathcal{C}_{\text{apt}}}, \boldsymbol{Y}^{\mathcal{C}_{\text{apt}}} \in [q]^V$.

1 $V_1 \leftarrow \{v_0\}$, $V_2 \leftarrow V \setminus V_1$, $V_{\text{set}} \leftarrow \emptyset$, $V_{\text{frozen}} \leftarrow \emptyset$ and $\mathcal{E}_{\text{frozen}} \leftarrow \emptyset$;

2 let $\boldsymbol{X}^{\mathcal{C}_{\text{apt}}}$ and $\boldsymbol{Y}^{\mathcal{C}_{\text{apt}}}$ be two empty assignments;

3 **while** $\exists e \in \mathcal{E}$ *s.t.* $e \cap V_1 \neq \emptyset, (e \cap V_2) \setminus (V_{\text{set}} \cup V_{\text{frozen}}) \neq \emptyset$ **do**

4   let e be the first such hyperedge and u be the first variable in $(e \cap V_2) \setminus (V_{\text{set}} \cup V_{\text{frozen}})$;

5   extend $\boldsymbol{X}^{\mathcal{C}_{\text{apt}}}$ and $\boldsymbol{Y}^{\mathcal{C}_{\text{apt}}}$ to variable u by sampling $(X_u^{\mathcal{C}_{\text{apt}}}, Y_u^{\mathcal{C}_{\text{apt}}})$ from the optimal coupling between μ_{u,Φ^X} and μ_{u,Φ^Y};

6   update Φ^X by setting $Q_u^X \leftarrow \{X_u^{\mathcal{C}_{\text{apt}}}\}$, update Φ^Y by setting $Q_u^Y \leftarrow \{Y_u^{\mathcal{C}_{\text{apt}}}\}$;

7     $V_{\text{set}} \leftarrow V_{\text{set}} \cup \{u\}$;

8   **if** $X_u^{\mathcal{C}_{\text{apt}}} \neq Y_u^{\mathcal{C}_{\text{apt}}}$ **then**

9         $V_1 \leftarrow V_1 \cup \{u\}$, $V_2 \leftarrow V \setminus V_1$;

10   **for** $e \in \mathcal{E}$ *s.t. the constraint c represented by e is satisfied by both* $X_u^{\mathcal{C}_{\text{apt}}}$ *and* $Y_u^{\mathcal{C}_{\text{apt}}}$ **do**

11         $\mathcal{E} \leftarrow \mathcal{E} \setminus \{e\}$, update Φ^X and Φ^Y by removing constraint c from C, i.e. $C \leftarrow C \setminus \{c\}$;

12   **for** $e \in \mathcal{E}$ *s.t.* $\mathrm{Vol}_{\Phi^X}(e) \leq \gamma$ *or* $\mathrm{Vol}_{\Phi^Y}(e) \leq \gamma$ **do**

13         $V_{\text{frozen}} \leftarrow V_{\text{frozen}} \cup ((e \cap V_2) \setminus V_{\text{set}})$;

14   **for** $e \in \mathcal{E}$ *s.t.* $(e \cap V_2) \setminus (V_{\text{set}} \cup V_{\text{frozen}}) = \emptyset$ **do**

15         $\mathcal{E}_{\text{frozen}} \leftarrow \mathcal{E}_{\text{frozen}} \cup \{e\}$;

16   **while** $\exists e \in \mathcal{E}_{\text{frozen}}$ *s.t.* $e \cap V_1 \neq \emptyset$ *and* $e \cap V_{\text{frozen}} \neq \emptyset$ **do**

17         $V_1 \leftarrow V_1 \cup (e \cap V_{\text{frozen}})$, $V_2 \leftarrow V \setminus V_1$, $V_{\text{frozen}} \leftarrow V_{\text{frozen}} \setminus e$;

18 extend $\boldsymbol{X}^{\mathcal{C}_{\text{apt}}}$ and $\boldsymbol{Y}^{\mathcal{C}_{\text{apt}}}$ to the set $V_2 \setminus V_{\text{set}}$ by sampling $(X_{V_2 \setminus V_{\text{set}}}^{\mathcal{C}_{\text{apt}}}, Y_{V_2 \setminus V_{\text{set}}}^{\mathcal{C}_{\text{apt}}})$ from the optimal coupling between $\mu_{V_2 \setminus V_{\text{set}}, \Phi^X}$ and $\mu_{V_2 \setminus V_{\text{set}}, \Phi^Y}$;

19 extend $\boldsymbol{X}^{\mathcal{C}_{\text{apt}}}$ and $\boldsymbol{Y}^{\mathcal{C}_{\text{apt}}}$ to the set $V_1 \setminus V_{\text{set}}$ by sampling $(X_{V_1 \setminus V_{\text{set}}}^{\mathcal{C}_{\text{apt}}}, Y_{V_1 \setminus V_{\text{set}}}^{\mathcal{C}_{\text{apt}}})$ from the optimal coupling between $\mu_{V_1 \setminus V_{\text{set}}, \Phi^X}(\cdot \mid \boldsymbol{X}^{\mathcal{C}_{\text{apt}}})$ and $\mu_{V_1 \setminus V_{\text{set}}, \Phi^Y}(\cdot \mid \boldsymbol{Y}^{\mathcal{C}_{\text{apt}}})$;

20 **return** $(\boldsymbol{X}^{\mathcal{C}_{\text{apt}}}, \boldsymbol{Y}^{\mathcal{C}_{\text{apt}}})$;

The coupling procedure $\mathcal{C}_{\text{apt}}$ starts from two empty assignments $\boldsymbol{X}^{\mathcal{C}_{\text{apt}}}$ and $\boldsymbol{Y}^{\mathcal{C}_{\text{apt}}}$, then gradually extends these assignments, finally outputs two full assignments on V. The following three basic sets of variables are maintained by the coupling.

- V_1/V_2: V_1 is a superset of *discrepancy* variables, which contains all variables w such that the coupling on w may be failed i.e. $X_w^{\mathcal{C}_{\text{apt}}} \neq Y_w^{\mathcal{C}_{\text{apt}}}$; $V_2 = V \setminus V_1$ is the complement of set V_1;
- V_{set}: the set of variables whose values are already assigned by the coupling procedure.

In addition, the coupling procedure $\mathcal{C}_{\text{apt}}$ also maintains two CSP formulas $\Phi^X = (V, \boldsymbol{Q}^X, C))$, $\Phi^Y = (V, \boldsymbol{Q}^Y, C)$ and a hypergraph $H = (V, \mathcal{E})$ modeling these two formulas. In each step, we pick a suitable variable u (Line 4), extend $\boldsymbol{X}^{\mathcal{C}_{\text{apt}}}$ and $\boldsymbol{Y}^{\mathcal{C}_{\text{apt}}}$ to variable u (Line 5). We then remove all the constraints (together with corresponding hyperedges[2]) satisfied by both $X_u^{\mathcal{C}_{\text{apt}}}$ and $Y_u^{\mathcal{C}_{\text{apt}}}$ (Line 11), update Φ^X and Φ^Y by setting $Q_u^X \leftarrow \{X_u^{\mathcal{C}_{\text{apt}}}\}$ and $Q_u^Y \leftarrow \{Y_u^{\mathcal{C}_{\text{apt}}}\}$ (Line 6). In other words, we force u in Φ^X to take the value $X_u^{\mathcal{C}_{\text{apt}}}$, and force u in Φ^Y to take the value $Y_u^{\mathcal{C}_{\text{apt}}}$.

[2]Remark that $\mathcal{E}$ is a multi-set of hyperedges. Once a hyperedge e is removed from $\mathcal{E}$ in Line 11, we only remove a single copy of e representing the constraint c.

The coupling procedure C_{apt} guarantees that the volume of all hyperedges $e \in \mathcal{E}$ cannot be too small in the whole procedure. This property is controlled by the parameter γ. Thus, the coupling procedure C_{apt} is adaptive with respect to the current volumes of hyperedges. Specifically, the following two sets are maintained during the coupling.

- V_{frozen}: the set of *frozen* variables, which is a set of unassigned variables in V_2, where each $w \in V_{\text{frozen}}$ is incident to a hyperedge e such that the volume of e is below the threshold γ.
- $\mathcal{E}_{\text{frozen}}$: the multi-set of *frozen* hyperedges such that for each hyperedge $e \in \mathcal{E}_{\text{frozen}}$, all unassigned variables in $e \cap V_2$ are frozen.

Once the volume of some hyperedge e is below the threshold γ (Line 12), we froze all unassigned variables in $e \cap V_2$ (Line 13). Once a variable becomes frozen, the coupling cannot assign values to this variable. If in a hyperedge e, all unassigned variables in $e \cap V_2$ are frozen, then the coupling cannot assign values to any unassigned variables e, the hyperedge e becomes frozen (Line 14 and Line 15). Finally, once a frozen hyperedge both contains frozen variables and variables in V_1, we put all frozen variables in this hyperedge into V_1 (Line 16 and Line 17).

Once the while-loop in Algorithm 3 terminates, we then sample assignments for variables in $V_2 \setminus V_{\text{set}}$ and $V_1 \setminus V_{\text{set}}$ from the conditional distributions (Line 18 and Line 19).

Lemma 8.3. *The coupling procedure C_{apt} satisfies the following properties:*

- *the coupling procedure will terminate eventually;*
- *the output $X^{C_{\text{apt}}} \in [q]^V$ follows $\mu^{X_{V\setminus\{v\}}}$ and the output $Y^{C_{\text{apt}}} \in [q]^V$ follows $\mu^{Y_{V\setminus\{v\}}}$;*
- *for any time of the coupling procedure and any e in the current set $\mathcal{E}$, it holds that*

$$\mathrm{Vol}_{\Phi^X}(e) \geq \frac{\gamma}{q} \quad \text{and} \quad \mathrm{Vol}_{\Phi^Y}(e) \geq \frac{\gamma}{q};$$

- *for any variable $u \in V$, if $X_u^{C_{\text{apt}}} \neq Y_u^{C_{\text{apt}}}$ in the final output, then $u \in V_1$.*

Proof. We prove that the coupling C_{apt} must terminate. Consider the while-loop in Line 16 and Line 17. After the Line 17, the hyperedge e cannot satisfy the condition in Line 16 (because $e \cap V_{\text{frozen}} = \emptyset$), thus the while-loop in Line 16 and Line 17 will terminate eventually. Consider the main while-loop (Line 3). After each loop, the size of V_{set} will increase by 1. Note that the size of V_{set} cannot be greater than n. Hence, the coupling C_{apt} will terminate eventually.

We prove that the output $X^{C_{\text{apt}}} \in [q]^V$ follows the distribution $\mu^{X_{V\setminus\{v\}}}$. The result for the output $Y^{C_{\text{apt}}} \in [q]^V$ can be proved in a similar way. Consider the input CSP formula $\Phi^X = (V, C, (Q_u^X)_{u \in V})$ defined in (36). It holds that the uniform distribution μ_{Φ^X} of all satisfying assignments to Φ^X is precisely the distribution $\mu^{X_{V\setminus\{v\}}}$. Suppose $V_{\text{set}} = \{u_1, u_2, \ldots, u_\ell\}$, where u_i is the i-th variable whose value is assigned by the coupling C_{apt}. The following properties holds:

- the value of u_1 is sampled from the marginal distribution μ_{u_1,Φ^X};
- for each $1 \leq i < \ell$, once u_i gets the value $X_{u_i}^{C_{\text{apt}}}$, we fix $Q_{u_i}^X$ as $\{X_{u_i}^{C_{\text{apt}}}\}$ (Line 6) and remove a subset of constraints satisfied by current $X_u^{C_{\text{apt}}}$ (Line 11); after updated Φ^X, we sample the value of u_{i+1} from the marginal distribution μ_{u_{i+1},Φ^X};
- given the assignment of V_{set}, the assignments of $V_2 \setminus V_{\text{set}}$ and $V_1 \setminus V_{\text{set}}$ are sampled from the conditional distributions in Line 18 and Line 19.

Note that for each u_i, the marginal distribution μ_{u_i,Φ^X} is precisely the distribution $\mu^{X_{V\setminus\{v\}}}$ projected on u_i conditional on the value of u_j is fixed as $X_{u_j}^{C_{\text{apt}}}$ for all $j < i$. By the chain rule, the output $X^{C_{\text{apt}}} \in [q]^V$ follows the distribution $\mu^{X_{V\setminus\{v\}}}$.

We now prove the third property. By (37) and (38), initially, for all $e \in \mathcal{E}$, it holds that $\mathrm{Vol}_{\Phi^X}(e) > \frac{\gamma}{q}$ and $\mathrm{Vol}_{\Phi^Y}(e) > \frac{\gamma}{q}$. Suppose during the coupling procedure, there is a time such that some hyperedge e in the current set $\mathcal{E}$ satisfies $\mathrm{Vol}_{\Phi^X}(e) < \frac{\gamma}{q}$ or $\mathrm{Vol}_{\Phi^Y}(e) < \frac{\gamma}{q}$. Without loss generality, we assume $\mathrm{Vol}_{\Phi^X}(e) < \frac{\gamma}{q}$. The case $\mathrm{Vol}_{\Phi^Y}(e) < \frac{\gamma}{q}$ follows from symmetry. Recall

$$\mathrm{Vol}_{\Phi^X}(e) \triangleq \prod_{u \in e} \left|Q_u^X\right|.$$

Note that the volume $\mathrm{Vol}_{\Phi^X}(e)$ decreases only if we update Q_u^X for some $u \in e$ in Line 6. Note that for any $u \in V$, it holds that $|Q_u^X| \le q$. In Line 6, once the coupling sets $Q_u^X \leftarrow \{X_u^{C_{\mathrm{apt}}}\}$, the volume $\mathrm{Vol}_{\Phi^X}(e)$ decreases by at most a factor q. If $\mathrm{Vol}_{\Phi^X}(e) < \frac{\gamma}{q}$, the following event must occur

- event $\mathcal{B}$: the main while-loop pick a variable $u \in e$ after $\mathrm{Vol}_{\Phi^X}(e) < \gamma$.

We show that the event $\mathcal{B}$ cannot occur. Consider the first time that $\mathrm{Vol}_{\Phi^X}(e) < \gamma$. After Line 12 and Line 13, it must hold that

$$e \subseteq V_1 \cup V_{\mathrm{set}} \cup V_{\mathrm{frozen}}. \tag{39}$$

Note that the coupling C_{apt} only adds variables into V_1 and V_{set}, but never deletes variables from V_1 and V_{set}. Also note that if a variable is removed from V_{frozen}, it must be added into V_1(Line 17). Thus, (39) holds up to the end of the coupling. Consider the variable u in event $\mathcal{B}$, u must satisfy $u \in V_2 \setminus (V_{\mathrm{set}} \cup V_{\mathrm{frozen}})$. However, by (39), there is no such variable u in hyperedge e. Contradiction.

Finally, we prove the last property. In this proof, we consider $V_1, V_2, V_{\mathrm{set}}, V_{\mathrm{frozen}}, \mathcal{E}, \mathcal{E}_{\mathrm{frozen}}$ when the main while-loop in C_{apt} terminates. We claim that the following properties holds:

- (I) for any $u \in V_2 \cap V_{\mathrm{set}}$, $X_u^{C_{\mathrm{apt}}} = Y_u^{C_{\mathrm{apt}}}$;
- (II) for any $e \in \mathcal{E}$ such that $e \cap V_1 \neq \emptyset$ and $e \cap V_2 \neq \emptyset$, $e \cap V_2 \subseteq V_{\mathrm{set}}$.

Consider the CSP formulas Φ^X and Φ^Y in Line 18. Note that both Φ^X and Φ^Y are modeled by hypergraph $H = (V, \mathcal{E})$. Define a set of variables

$$R = \bigcup_{\substack{e \in \mathcal{E} \\ e \cap V_1 \neq \emptyset, e \cap V_2 \neq \emptyset}} (e \cap V_2).$$

Recall μ_{Φ^X} and μ_{Φ^Y} are the uniform distributions of satisfying assignments to Φ^X and Φ^Y. By the definition of R, conditional on any assignment $\sigma \in [q]^R$ on set R, the assignment on $V_2 \setminus R$ is independent with the assignment on V_1. By property (I) and (II), it holds that $R \subseteq V_2 \cap V_{\mathrm{set}}$ and $X_R^{C_{\mathrm{apt}}} = Y_R^{C_{\mathrm{apt}}}$. Since $R \subseteq V_{\mathrm{set}}$ and $X_R^{C_{\mathrm{apt}}} = Y_R^{C_{\mathrm{apt}}}$, for any $u \in R$, $|Q_u^X| = |Q_u^Y| = 1$ and $Q_u^X = Q_u^Y$. Hence, in Φ^X and Φ^Y, variables in R are fixed as a same value in $[q]$. Thus, $\mu_{V_2 \setminus V_{\mathrm{set}}, \Phi^X}$ and $\mu_{V_2 \setminus V_{\mathrm{set}}, \Phi^Y}$ are identical distributions. By Line 18,

$$X_{V_2 \setminus V_{\mathrm{set}}}^{C_{\mathrm{apt}}} = Y_{V_2 \setminus V_{\mathrm{set}}}^{C_{\mathrm{apt}}} \tag{40}$$

Combining property (I) and (40) proves that $X_{V_2}^{C_{\mathrm{apt}}} = Y_{V_2}^{C_{\mathrm{apt}}}$. This proves the last property.

We finish the prove by proving properties (I) and (II). The property (I) is trivial, because for any $u \in V_{\mathrm{set}}$, if $X_u^{C_{\mathrm{apt}}} \neq Y_u^{C_{\mathrm{apt}}}$, then by Line 9, it must hold that $u \in V_1$. We then prove property (II). Suppose there is an hyperedge e such that $e \cap V_1 \neq \emptyset$, $e \cap V_2 \neq \emptyset$ and e violates property (II). We define a set

$$S(e) = (e \cap V_2) \setminus V_{\mathrm{set}} = (e \setminus V_1) \setminus V_{\mathrm{set}} \neq \emptyset.$$

There are only two possibilities for the set $S(e)$, we show neither of them is possible.

- $S(e) \not\subseteq V_{\mathrm{frozen}}$: in this case, e satisfies the condition in the main while-loop (Line 3), the main while-loop cannot terminate; contradiction.
- $S(e) \subseteq V_{\mathrm{frozen}}$: in this case, by Line 14 and Line 15, $e \in \mathcal{E}_{\mathrm{frozen}}$; hence, e satisfies the condition in Line 16, then by Line 17, all variables in $e \cap V_{\mathrm{frozen}}$ are removed from V_{frozen} and added into V_1, thus there is no such non-empty subset $S(e) \subseteq e$ such that $S(e) \subseteq V_{\mathrm{frozen}}$; contradiction.

Hence, such non-empty subset $S(e)$ does not exist, which implies property (II) holds. □

By Lemma 8.3 and the coupling lemma (Lemma 2.2), to bound the $I_{v_\star}$ in (33), we can bound

$$I_{v_\star} = d_{\mathrm{TV}}\left(\mu_{v_\star}^{X_{V \setminus \{v_\star\}}}, \mu_{v_\star}^{Y_{V \setminus \{v_\star\}}}\right) \le \Pr_{C_{\mathrm{apt}}}\left[v_\star \in V_1\right], \tag{41}$$

where V_1 denotes the set V_1 at the end of the coupling C_{apt}.

In the rest of the proof, our task is to bounding the RHS of (41). From now, we use hypergraph $H = (V, \mathcal{E})$ to model the *input* CSP formulas Φ^X and Φ^Y in Algorithm 3. For any $v \in V$, define

$$N_{\mathrm{vtx}}(v) \triangleq \{u \neq v \mid \exists e \in \mathcal{E} \text{ s.t. } u, v \in e\}.$$

We say a variable u is incident to a hyperedge e if $u \in e$; a sequence of variables $v_0, v_1, \dots, v_\ell$ is a path in hypergraph H if $v_i \in N_{\mathrm{vtx}}(v_{i-1})$ for all $1 \leq i \leq \ell$. We define the failed variables and failed edges.

Definition 8.4. Consider the time when the main while-loop in coupling procedure C_{apt} terminates.

- A variable $u \in V$ is said to be failed if $u \in V_{\mathrm{set}}$ and $X_u^{C_{\mathrm{apt}}} \neq Y_u^{C_{\mathrm{apt}}}$.
- A hyperedge $e \in \mathcal{E}$ is said to be failed if both of the following two properties hold:
  1. the constraint represented by e is not satisfied by both $\boldsymbol{X}^{C_{\mathrm{apt}}}$ and $\boldsymbol{Y}^{C_{\mathrm{apt}}}$;
  2. $\mathrm{Vol}_{\Phi^X}(e) < \gamma$ or $\mathrm{Vol}_{\Phi^Y}(e) < \gamma$.

Lemma 8.5. *For any $u \in V_1$, there exists a path $u_0, u_1, \dots, u_\ell \in V$ in H such that*

- *$u_0 = v_0$ is the initial disagreement variable, $u_\ell = u$ and $u_i \in V_1$ for all $0 \leq i \leq \ell$;*
- *for any $1 \leq i \leq \ell$, either u_i is failed or u_i is incident to a failed hyperedge e_i.*

Proof. Suppose $V_1 = \{v_0, v_1, v_2, \dots, v_m\}$, where v_0 is the initial disagreement variable and v_i is the i-th variable added into set V_1. If a set of variables are added into V_1 at the same time (Line 17), we break tie arbitrarily. We prove the first part of the lemma by induction on the index i.

The base case is $i = 0$, the first part of the lemma holds for the path that only contains v_0.

Assuming the lemma holds up to index i, we prove the lemma for index $i + 1$. Consider the time when v_{i+1} is added into the set V_1. There are following two possibilities.

- v_{i+1} is added in Line 9. Consider the hyperedge e in Line 4. It holds that $v_{i+1} \in e$ and $e \cap V_1 \neq \emptyset$, where $V_1 = \{v_0, v_1, \dots, v_i\}$. Pick an arbitrary $v_j \in e \cap V_1$. By induction hypothesis, since $j < i$, there exists a path $u_0 = v_0, u_1, u_2, \dots, u_\ell = v_j$ for v_j. Note that $v_{i+1} \in e$ and $v_j \in e$. We can find the path $u_0 = v_0, u_1, u_2, \dots, u_\ell = v_j, u_{\ell+1} = v_{i+1}$ for v_{i+1}.
- v_{i+1} is added in Line 17. Consider the hyperedge e satisfying the condition in Line 16. It holds that $v_{i+1} \in e$ and $e \cap V_1 \neq \emptyset$, where $V_1 = \{v_0, v_1, \dots, v_i\}$. Pick an arbitrary $v_j \in e \cap V_1$. By induction hypothesis, since $j < i$, there exists a path $u_0 = v_0, u_1, u_2, \dots, u_\ell = v_j$ for v_j. Note that $v_{i+1} \in e$ and $v_j \in e$. We can find the path $u_0 = v_0, u_1, u_2, \dots, u_\ell = v_j, u_{\ell+1} = v_{i+1}$ for v_{i+1}.

We now prove the second part of the lemma. It suffices to show that for any $u \in V_1 \setminus \{v_0\}$, either u is failed or u is incident to a failed hyperedge e. Note that a variable u is added into V_1 in either Line 9 or Line 17. If u is added in Line 9, then it holds that $X_u^{C_{\mathrm{apt}}} \neq Y_u^{C_{\mathrm{apt}}}$, thus u is a failed variable. Suppose u is added in Line 17. Before the execution of Line 17, $u \in V_{\mathrm{frozen}}$ must be a frozen variable. Consider the moment that u becomes frozen. By Line 13, u must belong to a hyperedge e such that e is not satisfied by both $\boldsymbol{X}^{C_{\mathrm{apt}}}$ and $\boldsymbol{Y}^{C_{\mathrm{apt}}}$ (otherwise, e is deleted in Line 11) and $\min\{\mathrm{Vol}_{\Phi^X}(e), \mathrm{Vol}_{\Phi^Y}(e)\} < \gamma$. Note that after Line 13, $e \subseteq V_1 \cup V_{\mathrm{set}} \cup V_{\mathrm{frozen}}$. After that, in the main while-loop, the coupling C_{apt} cannot assign values to any unassigned variables in e. Thus, this hyperedge e is not satisfied by both $\boldsymbol{X}^{C_{\mathrm{apt}}}$ and $\boldsymbol{Y}^{C_{\mathrm{apt}}}$ up to the main while-loop in C_{apt} terminates. Hence, e is a failed hyperedge and u is incident to e. □

Lemma 8.5 says if a variable belongs to V_1, there exists a path satisfying the condition in Lemma 8.5. However, the failure probability of such path is not easy to bound. We next modify such path into a sequence whose failure probability is easy to bound.

Define the length of a path by the number of variables in this path minus 1, e.g. the length of the path $v_1, v_2, \dots, v_\ell$ is $\ell - 1$. For any two variables $u, w \in V$, the distance between u and w in H, denoted as $\mathrm{dist}_H(u, w)$, is the length of the shortest path between u and w in H. We extend the notion of distance to subsets of variables. For any variable $u \in V$ and subsets $S, T \subseteq V$, define

$$\mathrm{dist}_H(u, S) \triangleq \min_{w \in S} \mathrm{dist}_H(u, w);$$

$$\mathrm{dist}_H(S, T) \triangleq \min_{w \in S, w' \in T} \mathrm{dist}_H(w, w').$$

For such distance function $\mathrm{dist}_H(\cdot,\cdot)$, the triangle inequality may not hold for any subsets. But we will use the following two specific triangle inequalities.

$$\forall u_1, u_2, u_3 \in V, \qquad \mathrm{dist}_H(u_1,u_2) \le \mathrm{dist}_H(u_1,u_3) + \mathrm{dist}_H(u_3,u_2) \tag{42}$$

$$\forall u \in V, S, T \subseteq V \quad \mathrm{dist}_H(S,T) \le \mathrm{dist}_H(S,u) + \mathrm{dist}_H(u,T). \tag{43}$$

The inequality (42) holds trivially. Suppose $\mathrm{dist}_H(S,u) = \mathrm{dist}_H(u_S,u)$ for $u_S \in S$ and $\mathrm{dist}_H(u,T) = \mathrm{dist}_H(u,u_T)$ for $u_T \in T$. By (42), we have

$$\mathrm{dist}_H(S,T) \overset{(\star)}{\le} \mathrm{dist}_H(u_S,u_T) \le \mathrm{dist}_H(u_S,u) + \mathrm{dist}_H(u,u_T) = \mathrm{dist}_H(S,u) + \mathrm{dist}_H(u,T),$$

where $(\star)$ holds because $u_S \in S$ and $u_T \in T$. Remark that (43) covers (42), because S and T may only contain a single variable.

We have the following lemma.

Lemma 8.6. *For any $u \in V_1 \setminus \{v_0\}$, there exists a sequence of sets $S_1, S_2, \dots, S_\ell$, where each S_i is either a hyperedge or a set containing a single variable, such that*

- *$S_1, S_2, \dots, S_\ell$ are mutually disjoint;*
- *$\mathrm{dist}_H(v_0, S_1) \le 2$ and $\mathrm{dist}_H(u, S_\ell) = 0$;*
- *for any $1 \le i \le \ell - 1$, $\mathrm{dist}_H(S_i, S_{i+1}) \le 2$.*
- *for each $1 \le i \le \ell$, S_i either contains a failed variable or S_i is a failed hyperedge.*

Proof. Fix a variable $u \in V_1 \setminus \{v_0\}$. Let $v_0, v_1, \dots, v_m$ where $v_m = u$ denote the path in Lemma 8.5. For each $1 \le i \le m$ if v_i is not a failed variable, we use e_i to denote the failed hyperedge incident to v_i; if v_i is a failed variable, we let $e_i = \{v_i\}$. We first show that how to construct the sequence $S_1, S_2, \dots, S_\ell$, then we show that such sequence satisfies the properties in the lemma.

Let $\mathcal{S}$ be an empty stack. Let P denote the path $(v_1, v_2, \dots, v_m)$. Remark that P does not contain variable v_0. We repeat the following procedure until P becomes an empty path. We pick the last variable in the path P, denote this variable as v_i. We search for the minimum index j such that $j < i$ and $e_i \cap e_j \ne \emptyset$. Here are two cases depending on whether such index j exists.

- If such index j does not exist, then push e_i into the stack $\mathcal{S}$, remove v_i from the path P.
- If such index j exists, then push e_i into the stack $\mathcal{S}$, remove all v_t for $j \le t \le i$ from the path P.

Let $S_1, S_2, \dots, S_\ell$ be the elements in stack $\mathcal{S}$ from top to bottom.

We now prove that all S_i are disjoint. Suppose there are two indices $j < i$ such that $S_i \cap S_j \ne \emptyset$. Suppose $S_i = e_{i*}$ and $S_j = e_{j^*}$. It holds that $i^* > j^*$. e_{j^*} must be removed when processing e_{i^*}, thus e_{j^*} cannot be added into stack $\mathcal{S}$. Contradiction. This proves the first property.

We now prove the second property. Note that $u \in e_m$ and $S_\ell = e_m$, thus $\mathrm{dist}(u, S_\ell) = 0$. To bound $\mathrm{dist}_H(v_0, S_1)$, we consider two cases.

- Case $S_1 = e_1$. Note that v_0 and v_1 are adjacent in H, i.e. $\mathrm{dist}_H(v_0, v_1) = 1$. It holds that $v_1 \in S_1 = e_1$. Hence, $\mathrm{dist}_H(v_0, S_1) \le \mathrm{dist}_H(v_0, v_1) = 1$;
- Case $S_1 \ne e_1$. Suppose $S_1 = e_t$. In this case, it must hold that $e_1 \cap e_t \ne \emptyset$, thus $\mathrm{dist}_H(v_1, e_t) \le \mathrm{dist}_H(v_1, v^*) = 1$. where $v^* \in e_1 \cap e_t$ is an arbitrary variable. Note that $\mathrm{dist}_H(v_0, v_1) = 1$. By triangle inequality in (43), we have $\mathrm{dist}_H(v_0, e_t) \le \mathrm{dist}_H(v_0, v_1) + \mathrm{dist}_H(v_1, e_t) \le 2$.

Finally, we bound the distance $\mathrm{dist}_H(S_i, S_{i+1})$. Suppose $S_{i+1} = e_j$ and $S_i = e_{j'}$. Here are two cases.

- Case $j' = j - 1$: Note that $\mathrm{dist}_H(v_j, v_{j'}) = 1$, $v_j \in e_j$ and $v_{j'} \in e_{j'}$. We have $\mathrm{dist}_H(e_j, e_{j'}) \le \mathrm{dist}_H(v_j, v_{j'}) \le 1$. Hence, $\mathrm{dist}_H(S_i, S_{i+1}) = \mathrm{dist}_H(e_j, e_{j'}) \le 1$.
- Case $j' < j - 1$: Consider the moment when $S_{i+1} = e_j$ is added into $\mathcal{S}$. It must hold that $e_{j'+1} \cap e_j \ne \emptyset$. Note that $v_{j'} \in e_{j'}$ and $\mathrm{dist}_H(v_{j'}, v_{j'+1}) = 1$. We have $\mathrm{dist}_H(e_{j'}, v_{j'+1}) \le \mathrm{dist}_H(v_{j'}, v_{j'+1}) = 1$. Note that $v_{j'+1} \in e_{j'+1}$ and $e_{j'+1} \cap e_j \ne \emptyset$. It holds that $\mathrm{dist}_H(v_{j'+1}, e_j) \le \mathrm{dist}_H(v_{j'+1}, v^*) = 1$, where $v^* \in e_{j'+1} \cap e_j$ is an arbitrary variable. By triangle inequality in (43), $\mathrm{dist}_H(e_{j'}, e_j) \le \mathrm{dist}(e_{j'}, v_{j'+1}) + \mathrm{dist}_H(v_{j'+1}, e_j) \le 2$.

Combining two cases proves the third property.

For the last property, by Lemma 8.5, it is easy to see that each S_i is either a failed hyperedge or a set containing a single failed variable. □

We say a sequence of sets $S_1, S_2, \dots, S_\ell$ is a *percolation sequence (PS)* if the following three properties are satisfied:

- $S_1, S_2, \dots, S_\ell$ are mutually disjoint;
- $\mathrm{dist}_H(v_0, S_1) \le 2$;
- for any $1 \le i \le \ell - 1$, $\mathrm{dist}_H(S_i, S_{i+1}) \le 2$.

We say a percolation sequence $S_1, S_2, \dots, S_\ell$ is a percolation sequence for $v_\star$ if $\mathrm{dist}_H(v_\star, e_\ell) = 0$, i.e. $v_\star \in e_\ell$. For any S_i in sequence, we say S_i fails if either S_i contains a failed variable or S_i is a failed hyperedge. By (35) and Lemma 8.6, we have

$$I_{v_\star} \le \Pr_{\mathcal{C}_{\mathrm{apt}}}\left[X_{v_\star}^{\mathcal{C}_{\mathrm{apt}}} \neq Y_{v_\star}^{\mathcal{C}_{\mathrm{apt}}}\right] \le \sum_{\text{PS for } v_\star: S_1, S_2, \dots, S_\ell} \Pr_{\mathcal{C}_{\mathrm{apt}}}\left[\forall 1 \le i \le \ell, S_i \text{ fails}\right]. \tag{44}$$

The following lemma bounds the probability that all elements in a PS fail.

Lemma 8.7. *Fix a percolation sequence (PS) $S_1, S_2, \dots, S_\ell$ to $v_\star$. It holds that*

$$\Pr_{\mathcal{C}_{\mathrm{apt}}}\left[\forall 1 \le i \le \ell, S_i \text{ fails}\right] \le \prod_{\substack{1 \le i \le \ell \\ S_i \text{ contains a single variable}}} \frac{1}{8k^3d^2} \prod_{\substack{1 \le i \le \ell \\ S_i \text{ is a hyperedge}}} \frac{1}{8k^3d^3}.$$

We need the following technical lemma to prove Lemma 8.7. We introduce a parameter s to write γ defined in (38) as

$$\gamma = seq^2dk, \quad \text{where } s \triangleq 32k^2d^2. \tag{45}$$

Lemma 8.8. *During the coupling procedure $\mathcal{C}_{\mathrm{apt}}$, the CSP formulas $\Phi^X = (V, (Q_u^X)_{u \in V}, \mathbf{C})$ and $\Phi^Y = (V, (Q_u^Y)_{u \in V}, \mathbf{C})$ always satisfies that for any $u \in V \setminus (V_{\mathrm{set}} \cup \{v_0\})$, $Q_u^Y = Q_u^X$ and for any $j \in Q_u^X = Q_u^Y$,*

$$\begin{aligned} \frac{1}{q_u}\left(1 - \frac{4}{sk}\right) &\le \mu_{u,\Phi^X}(j) \le \frac{1}{q_u}\left(1 + \frac{4}{sk}\right) \\ \frac{1}{q_u}\left(1 - \frac{4}{sk}\right) &\le \mu_{u,\Phi^Y}(j) \le \frac{1}{q_u}\left(1 + \frac{4}{sk}\right), \end{aligned} \tag{46}$$

where $q_u = \left|Q_u^X\right| = \left|Q_u^Y\right|$, thus $d_{\mathrm{TV}}\left(\mu_{u,\Phi^X}, \mu_{u,\Phi^Y}\right) \le \frac{4}{sk}$.

Furthermore, for any optimal coupling $(x, y) \in Q_u^X \times Q_u^Y$ between μ_{u,Φ^X} and μ_{u,Φ^Y}, it holds that

$$\forall j \in Q_u^X = Q_u^Y \quad \Pr\left[x = j \lor y = j\right] = \max\left\{\mu_{u,\Phi^X}(j), \mu_{u,\Phi^Y}(j)\right\} \le \frac{1}{q_u}\left(1 + \frac{4}{sk}\right).$$

Proof. Initially, the input Φ^X and Φ^Y satisfy $Q_u^X = Q_u^Y$ for any $u \in V \setminus \{v_0\}$. Consider each update step in Line 6. After the value of u is assigned, we put the variable u into V_{set} in Line 7. It still holds that $Q_v^Y = Q_v^X$ for any $v \in V \setminus (V_{\mathrm{set}} \cup \{v_0\})$. By Lemma 8.3, at any time, for any e in current $\mathcal{E}$, it holds that

$$\begin{aligned} \mathrm{Vol}_{\Phi^X}(e) &= \prod_{u \in e} q_u \ge \frac{\gamma}{q} = seqdk \\ \mathrm{Vol}_{\Phi^Y}(e) &= \prod_{u \in e} q_u \ge \frac{\gamma}{q} = seqdk. \end{aligned}$$

We now prove (46) for Φ^X. The result for Φ^Y can be proved in a similar way. Let $\mathcal{D}$ denote the product distribution such that each variable $v \in V$ takes a value from Q_v^X uniformly at random. Let B_c to denote the bad event that the constraint c is not satisfied. Let $\mathcal{B} = (B_c)_{c \in C}$ denote the collection of bad events. Let $\Gamma(\cdot)$ be defined as in the Lovász local lemma (Theorem 2.1). For each $c \in C$, let $x(B_c) = \frac{1}{sqdk}$. For each constraint $c \in C$,

$$\begin{aligned} \Pr_{\mathcal{D}}\left[B_c\right] &= \prod_{u \in \mathsf{vbl}(c)} \frac{1}{q_u} \le \frac{1}{seqdk} \le \frac{1}{sqdk}\left(1 - \frac{1}{sqdk}\right)^{sqdk-1} \le \frac{1}{sqdk}\left(1 - \frac{1}{sqdk}\right)^{dk-1} \\ &\le x(B_c) \prod_{B_{c'} \in \Gamma(B_c)} (1 - x(B_{c'})), \end{aligned}$$

where the last inequality holds because the maximum degree of the dependency graph is at most $k(d-1) \le dk - 1$. Fix a $j \in Q_u^X = Q_u^Y$. Let A denote the event that v takes the value j. Note that $|\Gamma(A)| \le d$. By Lovász local lemma (Theorem 2.1), we have

$$\mu_{u,\Phi^X}(j) = \Pr_{\mu_{\Phi^X}}[A] \le \frac{1}{q_u}\left(1 - \frac{1}{sqdk}\right)^{-d} \le \frac{1}{q_u}\exp\left(\frac{2}{sqk}\right) \le \frac{1}{q_u}\left(1 + \frac{4}{sqk}\right),$$

which implies the upper bound in (46). Let A' denote the event that v does not take the value j. Note that $|\Gamma(A')| \le d$. By Lovász local lemma (Theorem 2.1), we have

$$\Pr_{\mu_{\Phi^X}}[A'] \le \left(1 - \frac{1}{q_u}\right)\left(1 - \frac{1}{sqdk}\right)^{-d} \le \left(1 - \frac{1}{q_u}\right)\exp\left(\frac{2}{sqk}\right) \le \left(1 - \frac{1}{q_u}\right)\left(1 + \frac{4}{sqk}\right).$$

We have

$$\mu_{u,\Phi^X}(j) = 1 - \Pr_{\mu_{\Phi^X}}[A'] \ge 1 - \left(1 - \frac{1}{q_u}\right)\left(1 + \frac{4}{sqk}\right) = \frac{1}{q_u}\left(1 - \frac{4q_u}{sqk} + \frac{4}{sqk}\right) \ge \frac{1}{q_u}\left(1 - \frac{4}{sk}\right),$$

where the last inequality holds because $q_u \le q$. This proves the lower bound in (46). The inequalities in (46) imply

$$d_{\mathrm{TV}}\left(\mu_{u,\Phi^X}, \mu_{u,\Phi^Y}\right) \le \frac{1}{2}\sum_{j \in Q_u^X = Q_u^Y}\left|\mu_{u,\Phi^X}(j) - \mu_{u,\Phi^Y}(j)\right| = \frac{4}{sk}.$$

Let $(x,y) \in Q_u^X \times Q_u^Y$ be the optimal coupling between μ_{u,Φ^X} and μ_{u,Φ^Y}. It holds that

$$\Pr[x = y] = 1 - d_{\mathrm{TV}}\left(\mu_{u,\Phi^X}, \mu_{u,\Phi^Y}\right)$$

Define a set $S = \{j \in Q_u^X = Q_u^Y \mid \mu_{u,\Phi^X}(j) \ge \mu_{u,\Phi^Y}(j)\}$. Note that $\sum_{j \in Q_u^X}\mu_{u,\Phi^X}(j) = \sum_{j \in Q_u^Y}\mu_{u,\Phi^Y}(j) = 1$. We have $d_{\mathrm{TV}}\left(\mu_{u,\Phi^X}, \mu_{u,\Phi^Y}\right) = \sum_{j \in S}(\mu_{u,\Phi^X}(j) - \mu_{u,\Phi^Y}(j))$, which implies

$$\begin{aligned}
\Pr[x = y] &= 1 - \sum_{j \in S}(\mu_{u,\Phi^X}(j) - \mu_{u,\Phi^Y}(j)) = \left(1 - \sum_{j \in S}\mu_{u,\Phi^X}(j)\right) + \sum_{j \in S}\mu_{u,\Phi^Y}(j) \\
&= \sum_{j \in Q_u^X \setminus S}\mu_{u,\Phi^X}(j) + \sum_{j \in S}\mu_{u,\Phi^Y}(j) \\
&= \sum_{j \in Q_u^X}\min\{\mu_{u,\Phi^X}(j), \mu_{u,\Phi^Y}(j)\}. \qquad (47)
\end{aligned}$$

On the other hand, since $(x,y) \in Q_u^X \times Q_u^Y$ as a valid coupling, we have

$$\forall j \in Q_u^X, \quad \Pr[x = y = j] \le \min\{\mu_{u,\Phi^X}(j), \mu_{u,\Phi^Y}(j)\}.$$

This implies that

$$\forall j \in Q_u^X \quad \Pr[x = y = j] = \min\{\mu_{u,\Phi^X}(j), \mu_{u,\Phi^Y}(j)\}. \qquad (48)$$

Fix a $j \in Q_u^X$. Without loss of generality, assume $\mu_{u,\Phi^X}(j) \ge \mu_{u,\Phi^Y}(j)$ (the case $\mu_{u,\Phi^X}(j) < \mu_{u,\Phi^Y}(j)$ follows from symmetry). By (48), $y = j$ implies $x = j$. Thus $x = j \vee y = j$ if and only if $x = j$. Thus,

$$\Pr[x = j \vee y = j] = \max\left\{\mu_{u,\Phi^X}(j), \mu_{u,\Phi^Y}(j)\right\} \le \frac{1}{q_u}\left(1 + \frac{4}{sk}\right).$$

□

Now, we are ready to prove Lemma 8.7.

Proof of Lemma 8.7. Given $\mathcal{S} = S_1, S_2, \ldots, S_\ell$, we define a set of variables $\mathsf{vbl}(\mathcal{S}) = \cup_{i=1}^{\ell} S_i$. For each $1 \le i \le \ell$, sample a random real number $r_i \in [0,1]$ uniformly and independently.

Consider the following implementation of coupling C_{apt}. In Line 5, we need to sample $X_u^{C_{\mathrm{apt}}}$ and $Y_u^{C_{\mathrm{apt}}}$ from the optimal coupling between marginal distributions μ_{u,Φ^X} and μ_{u,Φ^Y}. If $u \in \mathsf{vbl}(\mathcal{S})$, then we use the following implementation. We can find a unique S_i such that $u \in S_i$, because all S_i are mutually disjoint. We use random number r_i to implement the optimal coupling between μ_{u,Φ^X} and μ_{u,Φ^Y}. Here are two case for S_i: (1) $S_i = \{u\}$; (2) S_i is a hyperedge and $u \in S_i$. We handle two cases separately.

Suppose $S_i = \{u\}$. The optimal coupling satisfies $\Pr_{C_{\text{apt}}}\left[X_u^{C_{\text{apt}}} \neq Y_u^{C_{\text{apt}}}\right] = d_{\mathrm{TV}}\left(\mu_{u,\Phi^X}, \mu_{u,\Phi^Y}\right)$. The optimal coupling can be implemented as follows.

- If $r_i \leq d_{\mathrm{TV}}\left(\mu_{u,\Phi^X}, \mu_{u,\Phi^Y}\right)$, then sample a pair $(X_u^{C_{\text{apt}}}, Y_u^{C_{\text{apt}}})$ from the optimal coupling conditional on $X_u^{C_{\text{apt}}} \neq Y_u^{C_{\text{apt}}}$;
- If $r_i > d_{\mathrm{TV}}\left(\mu_{u,\Phi^X}, \mu_{u,\Phi^Y}\right)$, then sample a pair $(X_u^{C_{\text{apt}}}, Y_u^{C_{\text{apt}}})$ from the optimal coupling conditional on $X_u^{C_{\text{apt}}} = Y_u^{C_{\text{apt}}}$.

By Lemma 8.8, it holds that $d_{\mathrm{TV}}\left(\mu_{u,\Phi^X}, \mu_{u,\Phi^Y}\right) \leq \frac{4}{sk} = \frac{1}{8k^3d^2}$. Define the following event for S_i:

$$\mathcal{B}_i : \quad r_i \leq \frac{4}{sk} = \frac{1}{8k^3d^2}. \tag{49}$$

According to the implementation, if variable u fails in C_{apt}, then event $\mathcal{B}_i$ must occur.

Suppose $S_i = e$ is a hyperedge. Suppose e represents the constraint c such that c forbids a unique configuration $\sigma \in [q]^{\mathsf{vbl}(c)}$, i.e. $c(\sigma) = \texttt{False}$. In addition to r_i, we maintain two variables M_i and D_i for S_i, where $M_i \in [0,1]$ is a real number, $D_i \in \{0,1\}$ is a Boolean variable. Initially, $M_i = 1$ and $D_i = 0$. Suppose the coupling C_{apt} pick a variable $u \in e$. We sample $X_u^{C_{\text{apt}}}$ and $Y_u^{C_{\text{apt}}}$ via following procedure $\mathsf{Couple}(u)$.

- If $D_i = 1$, sample $X_u^{C_{\text{apt}}}$ and $Y_u^{C_{\text{apt}}}$ from the optimal coupling between μ_{u,Φ^X} and μ_{u,Φ^Y}. We does not need to use r_i to implement this sampling step.
- If $D_i = 0$, let $p_u = \max\{\mu_{u,\Phi^X}(\sigma_u), \mu_{u,\Phi^Y}(\sigma_u)\}$, then check whether $r_i \leq M_i p_u$.
  (1) if $r_i > M_i p_u$, sample $X_u^{C_{\text{apt}}}$ and $Y_u^{C_{\text{apt}}}$ from the optimal coupling between μ_{u,Φ^X} and μ_{u,Φ^Y} conditional on $X_u^{C_{\text{apt}}} \neq \sigma_u \wedge Y_u^{C_{\text{apt}}} \neq \sigma_u$; then set $D_i \leftarrow 1$;
  (2) if $r_i \leq M_i p_u$, sample $X_u^{C_{\text{apt}}}$ and $Y_u^{C_{\text{apt}}}$ from the optimal coupling between μ_{u,Φ^X} and μ_{u,Φ^Y} conditional on $X_u^{C_{\text{apt}}} = \sigma_u \vee Y_u^{C_{\text{apt}}} = \sigma_u$; then set $M_i \leftarrow M_i p_u$.

We first prove that above implementation is a valid coupling between μ_{u,Φ^X} and μ_{u,Φ^Y}. Note that if $D_i = 1$, then there is a variable $u \in e = S_i$ such that e is satisfied by both $X_u^{C_{\text{apt}}}$ and $Y_u^{C_{\text{apt}}}$, thus D_i indicates whether e is removed by the coupling. We claim

$$\text{conditional on } D_i = 0 \text{ and } M_i = m_i,\ r_i \text{ is a uniform random real number in } [0, m_i]\,. \tag{50}$$

Let $\mathcal{R}$ denote all the randomness of the coupling C_{apt} except the randomness of r_i. We first fix $\mathcal{R}$, then prove (50) by induction. Initially, r_i is sampled from $[0,1]$, $M_i = 1$, $D_i = 0$, the property holds. Consider one execution of $\mathsf{Couple}(u)$. Suppose $D_i = 0$ and $M_i = m_i$ before the execution. We show that (50) still holds after we sampled $X_u^{C_{\text{apt}}}$ and $Y_u^{C_{\text{apt}}}$ according to $\mathsf{Couple}(u)$. By induction hypothesis, r_i is a uniform random real number in $[0, m_i]$. Note that conditional on $\mathcal{R}$ and $D_i = 0$, the value of p_u is fixed.[3] After the procedure $\mathsf{Couple}(u)$, $D_i = 0$ if and only if $r_i \leq m_i p_u$. Since r_i is a uniform random real number in $[0, m_i]$, conditional on $r_i \leq m_i p_u$, r_i is a uniform random real number in $[0, m_i p_u]$. Since we set $m_i \leftarrow m_i p_u$ at the end of the procedure, thus r_i is a uniform random real number in $[0, m_i]$ after the procedure $\mathsf{Couple}(u)$, and (50) still holds.

To prove the validity of the implementation. First note that if $D_i = 1$, the validity holds trivially. If $D_i = 0$, by (50), r_i is a uniform random real number in $[0, M_i]$. Thus $r_i > M_i p_u$ with probability $1 - p_u$, and $r_i \leq M_i p_u$ with probability p_u. By Lemma 8.8, in the optimal coupling, the event $X_u^{C_{\text{apt}}} = c_u \vee Y_u^{C_{\text{apt}}} = c_u$ has probability p_u. Thus, the validity holds due to the chain rule.

Next, for hyperedge $S_i = e$, we define the following bad event

$$\mathcal{B}_i : \quad r_i \leq \frac{1}{8d^3k^3}. \tag{51}$$

We show that if the hyperedge $S_i = e$ fails, then $\mathcal{B}_i$ must occur.

[3] This is because $\mathcal{R}$ fixes all the randomness except the randomness of r_i. In our implementation, we only use r_i to compare with a threshold $M_i p_u$ when we couple $X_u^{C_{\text{apt}}}$ and $Y_u^{C_{\text{apt}}}$ in Line 5 for some $u \in e = S_i$. Conditional further on $D_i = 0$, the results of all previous comparisons are fixed, namely, r_i is smaller or equal to all the thresholds $m_i p_u$. Hence, given $\mathcal{R}$ and $D_i = 0$, the previous procedure of C_{apt} is fully determined, which implies p_u is fixed.

Suppose $S_i = e$ is a hyperedge. Consider the input CSP formulas $\Phi^X = (V, (Q_u^X)_{u\in V}, C)$ and $\Phi^Y = (V, (Q_u^Y)_{u\in V}, C)$. For any $u \neq v_0$, let $q_u = \left|Q_u^X\right| = \left|Q_u^Y\right|$. Suppose e represents the atomic constraint c such that $c(\sigma) = \texttt{False}$ for some unique $\sigma \in [q]^e$. Suppose after the coupling procedure C_{apt}, variables $u_1, u_2, \ldots, u_m \in V_{\text{set}} \cap e$. Since the hyperedge S_i fails, it holds that

- after the coupling procedure, $\text{Vol}_{\Phi^X}(e) < \gamma$ or $\text{Vol}_{\Phi^Y}(e) < \gamma$;
- for any $1 \le i \le m$, $X_{u_i}^{C_{\text{apt}}} = \sigma_{u_i}$ or $Y_{u_i}^{C_{\text{apt}}} = \sigma_{u_i}$.

The second property holds because otherwise e is satisfied by both $\boldsymbol{X}^{C_{\text{apt}}}$ and $\boldsymbol{Y}^{C_{\text{apt}}}$, thus must be removed by the coupling. According to our implementation, at the end of the coupling, we have

$$D_i = 0 \quad \text{and} \quad r_i \le M_i = \prod_{j=1}^{m} p_{u_j}.$$

Note that $s > 32$ (s is defined in (45)), $m \le k$ because $|e| = k$. By Lemma 8.8, we have

$$\prod_{j=1}^{m} p_{u_j} \le \prod_{j=1}^{m} \frac{1}{q_{u_j}}\left(1 + \frac{4}{sk}\right) \le \exp\left(\frac{4m}{sk}\right)\prod_{j=1}^{m} \frac{1}{q_{u_j}} \le \mathrm{e}\prod_{j=1}^{m} \frac{1}{q_{u_j}}.$$

At the end of the coupling, we have $\text{Vol}_{\Phi^X}(e) < \gamma$ or $\text{Vol}_{\Phi^Y}(e) < \gamma$. But in the beginning of the coupling, by (37), we have $\text{Vol}_{\Phi^X}(e) \ge 3000q^2d^6k^6$ and $\text{Vol}_{\Phi^Y}(e) \ge 3000q^2d^6k^6$. The volume of e decreases because we update Φ^X and Φ^Y in Line 6 for $u = u_1, u_2, \ldots, u_m$. Note that $v_0 \notin V_{\text{set}}$, thus $u_j \neq v_0$ for all $1 \le j \le m$. We have

$$\prod_{j=1}^{m} q_{u_j} \ge \frac{3000q^2d^6k^6}{\gamma} = \frac{3000q^2d^6k^6}{32\mathrm{e}q^2d^3k^3} = \frac{3000d^3k^3}{32\mathrm{e}}.$$

If the hyperedge S_i fails, then it holds that

$$r_i \le \prod_{j=1}^{m} p_{u_j} \le \mathrm{e}\prod_{j=1}^{m} \frac{1}{q_{u_j}} \le \frac{32\mathrm{e}^2}{3000d^3k^3} \le \frac{1}{8d^3k^3}.$$

Thus the event $\mathcal{B}_i$ must occur.

Combining two cases together, we have

$$\begin{aligned}
&\Pr_{C_{\text{apt}}}\left[\forall 1 \le i \le \ell, S_i \text{ fails}\right] \le \Pr\left[\forall 1 \le i \le \ell, \mathcal{B}_i\right] \\
\text{(all } r_i \text{ are mutually independent)} \quad &\le \prod_{i=1}^{\ell} \Pr\left[\mathcal{B}_i\right] \\
\text{(by (49) and (51))} \quad &\le \prod_{\substack{1\le i\le \ell \\ S_i \text{ contains a single variable}}} \frac{1}{8d^2k^3} \prod_{\substack{1\le i\le \ell \\ S_i \text{ is a hyperedge}}} \frac{1}{8d^3k^3}. \qquad \square
\end{aligned}$$

Recall a sequence of sets $S_1, S_2, \ldots, S_\ell$ is called a *percolation sequence (PS)* to $u \in V$ if it satisfies first three properties in Lemma 8.6. We call a sequence of sets $S_1, S_2, \ldots, S_\ell$ a *percolation sequence (PS)* if it satisfies first three properties in Lemma 8.6 except $\text{dist}_H(u, s_\ell) = 0$. For any S_i, let

$$p_{\text{fail}}(S_i) = \begin{cases} \frac{1}{8d^2k^3} & \text{if } S_i \text{ contains a single variable;} \\ \frac{1}{8d^3k^3} & \text{if } S_i \text{ is a hyperedge.} \end{cases} \tag{52}$$

Combining (44) and Lemma 8.7, we have

$$I_{v_\star} \le \sum_{\text{PS for } v_\star : e_1, e_2, \ldots, e_\ell} \Pr_{C_{\text{apt}}}\left[\forall 1 \le i \le \ell, S_i \text{ fails}\right] \le \sum_{\text{PS for } v_\star : e_1, e_2, \ldots, e_\ell} \prod_{i=1}^{\ell} p_{\text{fail}}(S_i).$$

Note that the hypergraph H is same for any $v_\star \in V \setminus \{v_0\}$. We can use the above inequality with $v_\star = v$ for all $v \in V \setminus \{v_0\}$. This implies

$$\sum_{v\in V: v\neq v_0} I_v \le \sum_{v\in V: v\neq v_0} \sum_{\text{PS to } v: S_1, S_2, \ldots, S_\ell} \prod_{1\le i\le \ell} p_{\text{fail}}(S_i) \le k \sum_{\text{PS}: S_1, S_2, \ldots, S_\ell} \prod_{1\le i\le \ell} p_{\text{fail}}(S_i),$$

where the last inequality holds because there are at most k variables v that satisfies $\mathrm{dist}(v, S_\ell) = 0$ (if S_ℓ contains a single variable, there are only one variable v; if S_ℓ is a hyperedge, there are k variables v). We can enumerate all the PSs according the length. We have

$$\sum_{v \in V: v \neq v_0} I_v \leq k \sum_{\ell=1}^{\infty} \sum_{\substack{\text{PS of length } \ell \\ S_1, S_2, \ldots, S_\ell}} \prod_{1 \leq i \leq \ell} p_{\text{fail}}(S_i) = k \sum_{\ell=1}^{\infty} N(\ell),$$

where

$$N(\ell) \triangleq \sum_{\substack{\text{PS of length } \ell \\ S_1, S_2, \ldots, S_\ell}} \prod_{1 \leq i \leq \ell} p_{\text{fail}}(S_i).$$

We then show that

$$N(\ell) \leq \left(k^2 d^2 \frac{1}{8d^2k^3} + k^2 d^3 \frac{1}{8d^3k^3}\right) \left(k^3 d^2 \frac{1}{8d^2k^3} + k^3 d^3 \frac{1}{8d^3k^3}\right)^{\ell - 1}. \tag{53}$$

We need the following basic facts to prove (53). We may assume $d, k \geq 2$, otherwise the sampling problem is trivial. Fix a variable $v \in V$. The number of variables u satisfying $\mathrm{dist}_H(v, u) \leq 2$ is at most

$$1 + d(k-1) + d(d-1)(k-1)^2 \leq k^2 d^2.$$

The number of hyperedges e' satisfying $\mathrm{dist}_H(v, e') \leq 2$ is at most

$$d + d(k-1)(d-1) + d(d-1)^2(k-1)^2 \leq k^2 d^3.$$

Fix a hyperedge $e \in \mathcal{E}$. The number of variables u satisfying $\mathrm{dist}_H(e, u) \leq 2$ is at most

$$k + k(d-1)(k-1) + k(d-1)^2(k-1)^2 \leq k^3 d^2.$$

The number of hyperedges e' satisfying $\mathrm{dist}_H(e, e') \leq 2$ is at most

$$(1 + k(d-1)) + k(k-1)(d-1)^2 + k(k-1)^2(d-1)^3 \leq k^3 d^3.$$

We prove (53) by induction on ℓ. Suppose $\ell = 1$. It holds that $\mathrm{dist}_H(v_0, S_1) \leq 2$. By (52), we have

$$N(1) \leq k^2 d^2 \frac{1}{8d^2k^3} + k^2 d^3 \frac{1}{8d^3k^3}.$$

Suppose (53) holds for all $\ell \leq k$. We prove (53) for $\ell = k+1$. For PS $S_1, S_2, \ldots, S_{k+1}$ of length $k+1$, $S_1, S_2, \ldots, S_k$ is a PS of length k and $\mathrm{dist}_H(S_k, S_{k+1}) \leq 2$. For any S_k, there are at most $k^3 d^2$ ways to choose S_{k+1} as a variable, and at most $k^3 d^3$ ways to choose S_{k+1} as a hyperedge. This implies

$$\begin{aligned} N(k+1) &\leq N(k) \left(k^3 d^2 \frac{1}{8d^2k^3} + k^3 d^3 \frac{1}{8d^3k^3}\right) \\ &\overset{\text{by I.H.}}{\leq} \left(k^2 d^2 \frac{1}{8d^2k^3} + k^2 d^3 \frac{1}{8d^3k^3}\right) \left(k^3 d^2 \frac{1}{8d^2k^3} + k^3 d^3 \frac{1}{8d^3k^3}\right)^k. \end{aligned}$$

This proves (53). Now, we have

$$\sum_{v \in V: v \neq v_0} I_v \leq k \sum_{\ell=1}^{\infty} N(\ell) \leq \sum_{\ell=1}^{\infty} \left(k^3 d^2 \frac{1}{8d^2k^3} + k^3 d^3 \frac{1}{8d^3k^3}\right)^\ell = \sum_{\ell=1}^{\infty} \left(\frac{1}{4}\right)^\ell \leq \frac{1}{2}.$$

8.4. **Non-adaptive coupling analysis.** We now analyze the general CSP formula $\Phi = (V, \mathcal{Q}, C)$ with atomic constraints, where each variable $v \in V$ has an arbitrary domain Q_v and each constraint contains arbitrary number of variables. We will prove the following lemma is this section.

Lemma 8.9. *Let $\Phi = (V, \mathcal{Q}, C)$ be the input CSP formula with atomic constraints in Algorithm 1. Let $\boldsymbol{h} = (h_v)_{v \in V}$ be the projection scheme for Φ satisfying Condition 3.4 with parameters α and β. Let $q_v = |Q_v|$, $p = \max_{c \in C} \prod_{v \in \mathsf{vbl}(c)} \frac{1}{q_v}$ and D denote the maximum degree of the dependency graph of Φ. If*

$$\log \frac{1}{p} \geq \frac{50}{\beta} \log \left(\frac{2000 D^4}{\beta}\right),$$

then it holds that $\sum_{v \in V \setminus \{v_0\}} d_{\mathrm{TV}}\left(\nu_v^{X_{V\setminus\{v\}}}, \nu_v^{Y_{V\setminus\{v\}}}\right) \le \frac{1}{2}$.

Fix a variable $v_\star \in V \setminus \{v_0\}$. The goal of this section is to construct a non-adaptive coupling C_{non} to bound the total variation distance $d_{\mathrm{TV}}\left(\nu_{v_\star}^{X_{V\setminus\{v_\star\}}}, \nu_{v_\star}^{Y_{V\setminus\{v_\star\}}}\right)$.

Recall that $\Phi = (V, \boldsymbol{Q}, C)$ is the original input CSP formula. Recall that two CSP formulas $\Phi^X = (V, \boldsymbol{Q}^X = (Q_u^X)_{u\in V}, C)$ and $\Phi^Y = (V, \boldsymbol{Q}^Y = (Q_v^Y)_{v\in V}, C)$ are defined by

$$Q_u^X = \begin{cases} h_u^{-1}(X_u) & \text{if } u \neq v_\star; \\ Q_u & \text{if } u = v_\star, \end{cases} \qquad Q_u^Y = \begin{cases} h_u^{-1}(Y_u) & \text{if } u \neq v_\star; \\ Q_u & \text{if } u = v_\star. \end{cases} \tag{54}$$

By definition, $(Q_u^X)_{u\in V}$ and $(Q_u^Y)_{u\in V}$ differ only at variable v_0. Let μ_{Φ^X} denote the uniform distribution over all satisfying assignments to Φ^X, and μ_{Φ^Y} denote the uniform distribution over all satisfying assignments to Φ^Y. The first step for non-adaptive coupling analysis is to construct another projection schemes on instances Φ^X and Φ^Y. Let $\boldsymbol{h}^X = (h_v^X)_{v\in V}$ denote the projection scheme for Φ^X and $\boldsymbol{h}^Y = (h_v^Y)_{v\in V}$ denote the projection scheme for Φ^Y, where $h_v^X : Q_v^X \to \Sigma_v^X$ and $h_v^Y : Q_v^Y \to \Sigma_v^Y$. For each $v \in V$, define

$$s_v^X \triangleq \left|\Sigma_v^X\right|, \quad s_v^Y \triangleq \left|\Sigma_v^Y\right|, \quad q_v^X = \left|Q_v^X\right|, \quad q_v^Y = \left|Q_v^Y\right|.$$

In our analysis, we construct a pair of projection schemes $\boldsymbol{h}^X, \boldsymbol{h}^Y$ satisfying the following condition.

Condition 8.10. *Let $\Phi = (V, Q, C)$ be the original input CSP formula of Algorithm 1 and $\boldsymbol{h} = (h_v)_{v\in V}$ be the original projection scheme for Φ satisfying Condition 3.4 with parameters α and β. The projection scheme $\boldsymbol{h}^X$ for Φ^X and the projection scheme $\boldsymbol{h}^Y$ for Φ^Y satisfy the following conditions:*

- *both $\boldsymbol{h}^X$ and $\boldsymbol{h}^Y$ are balanced, i.e. for each $v \in V$ and $c_v^X \in \Sigma_v^X$, $\lfloor q_v^X/s_v^X \rfloor \le \left|(h_v^X)^{-1}(c_v^X)\right| \le \lceil q_v^X/s_v^X \rceil$; for each $v \in V$ and $c_v^Y \in \Sigma_v^Y$, $\lfloor q_v^Y/s_v^Y \rfloor \le \left|(h_v^Y)^{-1}(c_v^Y)\right| \le \lceil q_v^Y/s_v^Y \rceil$;*
- *$\Sigma_{v_0}^X = \Sigma_{v_0}^Y$; and $h_u^X = h_u^Y$ for all $u \in V \setminus \{v_0\}$;*
- *$h_{v_\star}^X = h_{v_\star}^Y = h_{v_\star}$, where $h_{v_\star}$ is the original projection scheme $\boldsymbol{h}$ restricted on variable $v_\star$;*
- *for any constraint $c \in C$,*

$$\min\left(\sum_{v\in\mathsf{vbl}(c)} \log\left\lfloor \frac{q_v^X}{s_v^X}\right\rfloor, \sum_{v\in\mathsf{vbl}(c)} \log\left\lfloor \frac{q_v^Y}{s_v^Y}\right\rfloor\right) \ge \frac{\beta}{10}\left(\sum_{v\in\mathsf{vbl}(c)} \log q_v\right); \tag{55}$$

for any constraint $c \in C$ satisfying $v_\star \notin \mathsf{vbl}(c)$,

$$\min\left(\sum_{v\in\mathsf{vbl}(c)} \log \frac{q_v^X}{\lceil q_v^X/s_v^X\rceil}, \sum_{v\in\mathsf{vbl}(c)} \log \frac{q_v^Y}{\lceil q_v^Y/s_v^Y\rceil}\right) \ge \frac{\beta}{10}\left(\sum_{v\in\mathsf{vbl}(c)} \log q_v\right); \tag{56}$$

for any constraint $c \in C$ satisfying $v_\star \in \mathsf{vbl}(c)$,

$$\min\left(\log\left\lfloor\frac{q_{v_\star}^X}{s_{v_\star}^X}\right\rfloor + \sum_{v\in\mathsf{vbl}(c)\setminus\{v_\star\}} \log\frac{q_v^X}{\lceil q_v^X/s_v^X\rceil}, \log\left\lfloor\frac{q_{v_\star}^Y}{s_{v_\star}^Y}\right\rfloor + \sum_{v\in\mathsf{vbl}(c)\setminus\{v_\star\}} \log\frac{q_v^Y}{\lceil q_v^Y/s_v^Y\rceil}\right)$$
$$\ge \frac{\beta}{10}\left(\sum_{v\in\mathsf{vbl}(c)} \log q_v\right), \tag{57}$$

where $q_v^X = \left|Q_v^X\right|$, $q_v^Y = \left|Q_v^Y\right|$ and $q_v = |Q_v|$ for all $v \in V$.

Condition 8.10 is a variation of Condition 3.4. The lower bound in (56) can be transformed to the upper bounds on $\sum_{v\in\mathsf{vbl}(c)} \lceil q_v^X/s_v^X\rceil$ and $\sum_{v\in\mathsf{vbl}(c)} \lceil q_v^Y/s_v^Y\rceil$. Thus, (56) and (55) are similar to (6) and (7) in Condition 3.4. Moreover, for constraint $c \in C$ satisfying $v_\star \in \mathsf{vbl}(c)$, we need an extra condition in (57). The purpose of this extra condition is to handle the case that $|\mathsf{vbl}(c)|$ can be very large.

The following lemma shows that the projection schemes satisfying Condition 8.10 exist under a Lovász local lemma condition. Since we only use $\boldsymbol{h}^X$ and $\boldsymbol{h}^Y$ for analysis, we only need to show such projection schemes exist, we do not need an algorithm to construct specific projection schemes.

Lemma 8.11. *Let $\Phi = (V, \boldsymbol{Q}, C)$ be the original input CSP formula of Algorithm 1 and $\boldsymbol{h} = (h_v)_{v\in V}$ be the original projection scheme for Φ satisfying Condition 3.4 with parameters α and β. Let $q_v = |Q_v|$ and D denote the maximum degree of the dependency graph of Φ. Let $p \triangleq \max_{c\in C}\prod_{v\in\mathsf{vbl}(c)}\frac{1}{q_v}$. Suppose*

$$\log\frac{1}{p} \geq \frac{55}{\beta}(\log D + 3).$$

There exist projection schemes $\boldsymbol{h}^X, \boldsymbol{h}^Y$ for Φ^X, Φ^Y satisfying Condition 8.10.

The proof of Lemma 8.11 is deferred to Section 8.4.2.

Let $\boldsymbol{h}^X = (h_v^X)_{v\in V}$ and $\boldsymbol{h}^Y = (h_v^Y)_{v\in V}$ denote the projection schemes for Φ^X and Φ^Y, where $h_v^X : Q_v^X \to \Sigma_v^X$ and $h_v^Y : Q_v^Y \to \Sigma_v^Y$. Suppose $\boldsymbol{h}^X$ and $\boldsymbol{h}^Y$ satisfy Condition 8.10. By Condition 8.10, for any variable $v \in V$, $\Sigma_v^X = \Sigma_v^Y$ and $s_v^X = s_v^Y = |\Sigma_v^X| = |\Sigma_v^Y|$. Denote

$$\forall v \in V,\quad s'_v \triangleq s_v^X = s_v^Y \text{ and } \Sigma'_v \triangleq \Sigma_v^X = \Sigma_v^Y;$$
$$\Sigma' \triangleq \bigotimes_{v\in V}\Sigma'_v.$$

Recall μ_{Φ^X} and μ_{Φ^Y} are the uniform distributions over all satisfying assignments to Φ^X and Φ^Y. We define the following two projected distributions:

- ν_X: the projected distribution (defined in Definition 3.2) over $\Sigma' = \bigotimes_{v\in V}\Sigma'_v$ induced from the instance Φ^X and the projection scheme $\boldsymbol{h}^X$;
- ν_Y: the projected distribution (defined in Definition 3.2) over $\Sigma' = \bigotimes_{v\in V}\Sigma'_v$ induced from the instance Φ^Y and the projection scheme $\boldsymbol{h}^Y$.

For any variable $v \in V$, let $\nu_{v,X}$ and $\nu_{v,Y}$ denote the marginal distributions on v projected from ν_X and ν_Y. Recall the goal of this section is to bound $d_{\mathrm{TV}}\left(\nu_{v_\star}^{X_{V\setminus\{v_\star\}}}, \nu_{v_\star}^{Y_{V\setminus\{v_\star\}}}\right)$. By Condition 8.10, $h_{v_\star}^X = h_{v_\star}^Y = h_{v_\star}$. By the definitions Φ^X, Φ^Y and the projected distribution in Definition 3.2,

$$\nu_{v_\star}^{X_{V\setminus\{v_\star\}}} = \nu_{v_\star,X} \quad\text{and}\quad \nu_{v_\star}^{Y_{V\setminus\{v_\star\}}} = \nu_{v_\star,Y}.$$

Recall that $\Phi = (V, \boldsymbol{Q}, C)$ is the original input CSP formula of Algorithm 1. Recall that $H = (V, \mathcal{E})$ denotes the (multi-)hypergraph that models Φ, where $\mathcal{E} \triangleq \{\mathsf{vbl}(c) \mid c \in C\}$. Note that H also models Φ^X and Φ^Y, because Φ, Φ^X, Φ^Y have the same sets of variables and constraints. Let $e \in \mathcal{E}$ be a hyperedge and $u \in e$ a variable in e. Let $X_u^{C_{\mathrm{non}}}, Y_u^{C_{\mathrm{non}}} \in \Sigma'_u$ be two values. Let $c_e \in C$ denote the atomic constraint represented by e. Let $\sigma \in \boldsymbol{Q}_e$ denote the unique configuration forbidden by c_e, i.e. $c_e(\sigma) = \mathtt{False}$. We say e is satisfied by $X_u^{C_{\mathrm{non}}}$ if $\sigma_u \notin (h_u^X)^{-1}(X_u^{C_{\mathrm{non}}})$, because in the projected distribution ν_X, conditional on the value of u is $X_u^{C_{\mathrm{non}}}$, the constraint c_e must be satisfied. Similarly, We say e is satisfied by $Y_u^{C_{\mathrm{non}}}$ if $\sigma_u \notin (h_u^Y)^{-1}(Y_u^{C_{\mathrm{non}}})$. The coupling procedure C_{non} is given in Algorithm 4.

The input of the coupling C_{non} contains CSP formulas Φ^X and Φ^Y, together with projection schemes $\boldsymbol{h}^X$ and $\boldsymbol{h}^Y$ satisfying Condition 8.10. We also give an index function $\mathrm{ID} : V \to [n]$ such that each variable has a distinct index and the variable $v_\star$ has the largest index. The coupling will use this index to pick the variable in Line 5. Compared with the adaptive coupling in Algorithm 3, the coupling C_{non} is non-adaptive, i.e. it does not need to maintain the current volume of each hyperedge. Instead, the coupling C_{non} is given projection schemes $\boldsymbol{h}^X$ and $\boldsymbol{h}^Y$ in advance. Once the coupling C_{non} picks a variable u, it assigns the values in Σ'_u to variable u, where the domain Σ'_u is determined by $\boldsymbol{h}^X$ and $\boldsymbol{h}^Y$. The coupling C_{non} will put u into V_1 if the coupling on u fails. After that, the coupling will remove all the hyperedges satisfied by both $X_u^{C_{\mathrm{non}}}$ and $Y_u^{C_{\mathrm{non}}}$ in Line 11. If all variables in a hyperedge e are assigned values and e is still not satisfied, the coupling C_{non} puts e into V_1 in Line 13. Remark that after the while-loop, C_{non} only samples the value for $V_2 \setminus V_{\mathrm{set}}$ because $V_1 \subseteq V_{\mathrm{set}}$.

Lemma 8.12. *The coupling procedure C_{non} satisfies the following properties:*

- *the coupling procedure will terminate eventually;*
- *the output $\boldsymbol{X}^{C_{\mathrm{non}}} \in \Sigma'$ follows ν_X and the output $\boldsymbol{Y}^{C_{\mathrm{non}}} \in \Sigma'$ follows ν_Y;*
- *for any variable $u \in V$, if $X_u^{C_{\mathrm{non}}} \neq Y_u^{C_{\mathrm{non}}}$ in the final output, then $u \in V_1$.*

Algorithm 4: The coupling procedure C_{non}

Input : CSP formulas $\Phi^X = (V, \boldsymbol{Q}^X = (Q_u^X)_{u\in V}, C)$ and $\Phi^Y = (V, \boldsymbol{Q}^Y = (Q_v^Y)_{v\in V}, C)$, the hypergraph $H = (V, \mathcal{E})$ modeling Φ^X and Φ^Y, projection schemes $\boldsymbol{h}^X$ and $\boldsymbol{h}^Y$ satisfying Condition 8.10, variables $v_0, v_\star \in V$, an index function $\mathrm{ID} : V \to [n]$ such that $\mathrm{ID}(u) \neq \mathrm{ID}(v)$ for all $u \neq v$ and $\mathrm{ID}(v_\star) = n$.

Output : a pair of assignments $\boldsymbol{X}^{C_{\mathrm{non}}}, \boldsymbol{Y}^{C_{\mathrm{non}}} \in \Sigma'$.

1 sample $X_{v_0}^{C_{\mathrm{non}}} \sim \nu_{v_0,X}$ and $Y_{v_0}^{C_{\mathrm{non}}} \sim \nu_{v_0,Y}$ independently;
2 $V_1 \leftarrow \{v_0\}$, $V_2 \leftarrow V \setminus V_1$, $V_{\mathrm{set}} \leftarrow \{v_0\}$;
3 remove all e from $\mathcal{E}$ s.t. the constraint c represented by e is satisfied by both $X_{v_0}^{C_{\mathrm{non}}}$ and $Y_{v_0}^{C_{\mathrm{non}}}$;
4 **while** $\exists e \in \mathcal{E}$ *s.t.* $e \cap V_1 \neq \varnothing, (e \cap V_2) \setminus V_{\mathrm{set}} \neq \varnothing$ **do**
5   let e be the first such hyperedge and u the variable in $(e \cap V_2) \setminus V_{\mathrm{set}}$ with lowest ID;
6   sample $(c_X, c_Y) \in \Sigma'_u \times \Sigma'_u$ from the optimal coupling between $\nu_{u,X}(\cdot \mid \boldsymbol{X}^{C_{\mathrm{non}}})$ and $\nu_{u,Y}(\cdot \mid \boldsymbol{Y}^{C_{\mathrm{non}}})$ and extend $\boldsymbol{X}^{C_{\mathrm{non}}}$ and $\boldsymbol{Y}^{C_{\mathrm{non}}}$ to u by setting $(X_u^{C_{\mathrm{non}}}, Y_u^{C_{\mathrm{non}}}) \leftarrow (c_X, c_Y)$;
7     $V_{\mathrm{set}} \leftarrow V_{\mathrm{set}} \cup \{u\}$;
8   **if** $X_u^{C_{\mathrm{non}}} \neq Y_u^{C_{\mathrm{non}}}$ **then**
9         $V_1 \leftarrow V_1 \cup \{u\}$, $V_2 \leftarrow V \setminus V_1$;
10   **for** $e \in \mathcal{E}$ *s.t. the constraint c represented by e is satisfied by both* $X_u^{C_{\mathrm{non}}}$ *and* $Y_u^{C_{\mathrm{non}}}$ **do**
11         $\mathcal{E} \leftarrow \mathcal{E} \setminus \{e\}$
12   **for** $e \in \mathcal{E}$ *s.t.* $e \subseteq V_{\mathrm{set}}$ **do**
13         $V_1 \leftarrow V_1 \cup \{e\}$, $V_2 \leftarrow V \setminus V_1$;
14 extend $\boldsymbol{X}^{C_{\mathrm{non}}}$ and $\boldsymbol{Y}^{C_{\mathrm{non}}}$ to the set $V_2 \setminus V_{\mathrm{set}}$ by sampling $(X_{V_2\setminus V_{\mathrm{set}}}^{C_{\mathrm{non}}}, Y_{V_2\setminus V_{\mathrm{set}}}^{C_{\mathrm{non}}})$ from the optimal coupling between $\nu_{V_2\setminus V_{\mathrm{set}},X}(\cdot \mid \boldsymbol{X}^{C_{\mathrm{non}}})$ and $\nu_{V_2\setminus V_{\mathrm{set}},Y}(\cdot \mid \boldsymbol{Y}^{C_{\mathrm{non}}})$;
15 **return** $(\boldsymbol{X}^{C_{\mathrm{non}}}, \boldsymbol{Y}^{C_{\mathrm{non}}})$*;*

Proof. After each execution of the while-loop, the size of V_{set} will increase by 1. The size of V_{set} is at most n. Thus, the coupling procedure will terminate eventually.

We prove the second property for $\boldsymbol{X}^{C_{\mathrm{non}}}$. The result for $\boldsymbol{Y}^{C_{\mathrm{non}}}$ can be proved in a similar way. In Line 1, the coupling samples the $X_{v_0}^{C_{\mathrm{non}}}$ independently from the distribution $\nu_{v_0,X}$. Given the current configuration $\boldsymbol{X}^{C_{\mathrm{non}}}$, the coupling picks an unassigned variable u, then draw $X_u^{C_{\mathrm{non}}}$ from the conditional marginal distribution $\nu_{u,X}(\cdot \mid \boldsymbol{X}^{C_{\mathrm{non}}})$ in Line 6. Finally, the coupling samples $X_{V\setminus V_2}^{C_{\mathrm{non}}}$ from the conditional distribution. Note that $V_1 \subseteq V_{\mathrm{set}}$. When the coupling terminates, all variables $v \in V$ gets a value $X_v^{C_{\mathrm{non}}} \in \Sigma'_v$. By the chain rule, the output $\boldsymbol{X}^{C_{\mathrm{non}}} \in \Sigma'$ follows the law ν_X.

To prove the last property, we show that after the while loop, it holds that

- $X_{V_2\cap V_{\mathrm{set}}}^{C_{\mathrm{non}}} = Y_{V_2\cap V_{\mathrm{set}}}^{C_{\mathrm{non}}}$;
- $\nu_{V_2\setminus V_{\mathrm{set}},X}(\cdot \mid \boldsymbol{X}^{C_{\mathrm{non}}})$ and $\nu_{V_2\setminus V_{\mathrm{set}},Y}(\cdot \mid \boldsymbol{Y}^{C_{\mathrm{non}}})$ are identical distributions, thus all variables in $V_2 \setminus V_{\mathrm{set}}$ can be coupled perfectly.

Combining these two properties proves the last property in the lemma. The first property is easy to verify, because if $X_u^{C_{\mathrm{non}}} \neq Y_u^{C_{\mathrm{non}}}$, then u must be added into V_1 in Line 9. To prove the second property, we claim that, after the while-loop, there is no hyperedge $e \in \mathcal{E}$ such that $e \cap V_1 \neq \varnothing$ and $e \cap V_2 \neq \varnothing$. Suppose such hyperedge e exists. There are two possibilities for such hyperedge.

- $(e \cap V_2) \setminus V_{\mathrm{set}} \neq \varnothing$: In this case, the while-loop cannot terminate. Contradiction.
- $(e \cap V_2) \setminus V_{\mathrm{set}} = \varnothing$: Note that it always holds that $V_1 \subseteq V_{\mathrm{set}}$. In this case, it holds that $e \subseteq V_{\mathrm{set}}$. Note that $e \cap V_1 \neq \varnothing$ and $e \cap V_2 \neq \varnothing$. Hence, after the Line 1, there is no such hyperedge e. If such hyperedge e exists, it must be produced by the while-loop. Since $e \subseteq V_{\mathrm{set}}$, such hyperedge e will either be removed in Line 11, or added into V_1 in Line 13 (after which $e \cap V_2 = \varnothing$). This implies that such hyperedge does not exist when the while-loop terminates. Contradiction.

Hence, after the while-loop, all variables are divided into two parts V_1 and V_2. Besides, all the constraints $c \in C$ such that $\mathsf{vbl}(c) \cap V_1 \neq \emptyset$ and $\mathsf{vbl}(c) \cap V_2 \neq \emptyset$ are satisfied by both $\boldsymbol{X}^{C_{\text{non}}}$ and $\boldsymbol{Y}^{C_{\text{non}}}$. This implies, conditional on $\boldsymbol{X}^{C_{\text{non}}}$, the variables in V_2 is independent with the variables in V_1, and the same result holds for $\boldsymbol{Y}^{C_{\text{non}}}$. Note that two instances Φ^X and Φ^Y differ only at variable v_0, two projection schemes $\boldsymbol{h}^X$ and $\boldsymbol{h}^Y$ also differ only at v_0, and $v_0 \in V_1$. Since $X^{C_{\text{non}}}_{V_2 \cap V_{\text{set}}} = Y^{C_{\text{non}}}_{V_2 \cap V_{\text{set}}}$, $\nu_{V_2 \setminus V_{\text{set}}, X}(\cdot \mid \boldsymbol{X}^{C_{\text{non}}}) = \nu_{V_2 \setminus V_{\text{set}}, X}(\cdot \mid X^{C_{\text{non}}}_{V_2 \cap V_{\text{set}}})$ and $\nu_{V_2 \setminus V_{\text{set}}, Y}(\cdot \mid \boldsymbol{Y}^{C_{\text{non}}}) = \nu_{V_2 \setminus V_{\text{set}}, Y}(\cdot \mid Y^{C_{\text{non}}}_{V_2 \cap V_{\text{set}}})$ are identical distributions. □

For each hyperedge $e \in \mathcal{E}$, we say e is failed in coupling C_{non} if the following condition holds.

Definition 8.13. A hyperedge $e \in \mathcal{E}$ fails in the coupling C_{non} if one of the following two events occur.

- **Type-I failure**: there is a variable $u \in e \setminus \{v_0\}$ such that the coupling picks e and u in Line 5, and $X^{C_{\text{non}}}_u \neq Y^{C_{\text{non}}}_u$ after the coupling.
- **Type-II failure**: consider the time when the while-loop terminates. It holds that $e \subseteq V_{\text{set}}$ and the constraint represented by e is not satisfied by both $\boldsymbol{X}^{C_{\text{non}}}$ and $\boldsymbol{Y}^{C_{\text{non}}}$.

Let $\text{Lin}(H)$ denote the line graph of H, where each vertex in $\text{Lin}(H)$ is a hyperedge in H, two hyperedges $e, e' \in \mathcal{E}$ are connected if $e \cap e' \neq \emptyset$. Let $\text{Lin}^k(H)$ denote the k-th power graph of $\text{Lin}(H)$, two hyperedges e and e' are adjacent in $\text{Lin}^k(H)$ if their distance in $\text{Lin}(H)$ is no more than k. For each variable, we use $N(v)$ to denote the set of hyperedges incident to v:

$$N(v) \triangleq \{e \in \mathcal{E} \mid v \in e\}.$$

For any $k \geq 1$, define

$$N^k(v) \triangleq \left\{e \in \mathcal{E} \mid \exists e' \in N(v) \text{ s.t. } \text{dist}_{\text{Lin}(H)}(e, e') \leq k - 1\right\}, \tag{58}$$

where $\text{dist}_{\text{Lin}(H)}(e, e')$ denotes the length of the shortest path between e and e' in graph $\text{Lin}(H)$. Remark that $N(v) = N^1(v)$ by definition.

When the coupling C_{non} terminates, each variable $v \in V_1$ satisfies the following property.

Lemma 8.14. *For any $v \in V_1 \setminus \{v_0\}$, there exists a path $e_1, e_2, \ldots, e_\ell$ in $\text{Lin}^2(H)$ such that*

- *$e_1 \in N^2(v_0)$ and $v \in e_\ell$;*
- *for all $1 \leq i \leq \ell$, the hyperedge e_i fails in the coupling.*

Proof. Let $V_1 = \{v_0, v_1, v_2, \ldots, v_m\}$ denote the variables in V_1, where v_i is the i-th variables added into V_1. Remark that if a set of variables are added into V_1 at the same time (Line 13), we break tie arbitrarily. We prove the lemma by induction on index i.

The base case is v_0, the lemma holds for v_0 trivially. Suppose the lemma holds for $v_0, v_1, \ldots, v_{k-1}$. We prove the lemma for variable v_k. The variable v_k is added into V_1 either in Line 9 or Line 13.

- Suppose v_k is added into V_1 in Line 9. Variable v_k must be picked in Line 5. Consider the hyperedge e picked in Line 5. The hyperedge e fails in type-I because $v_k \in e$ and $X^{C_{\text{non}}}_{v_k} \neq Y^{C_{\text{non}}}_{v_k}$. Besides, it holds that $v_k \in e$ and $v_j \in e$ for some $j < k$. If $j = 0$, the lemma holds trivially. If $0 < j < k$, by induction hypothesis, there is a path $e_1, e_2, \ldots, e_t$ for v_j. Since $v_j \in e_t$ and $v_j \in e$, the lemma holds for v_k with the path $e_1, e_2, \ldots, e_t, e$.
- Suppose v_k is added into V_1 in Line 13. Let e denote the hyperedge in Line 13. It holds that that $v_k \in e$. By Line 12, $e \subseteq V_{\text{set}}$. Since e is not deleted in Line 3 or Line 11, the constraint represented by e is not satisfied by both $\boldsymbol{X}^{C_{\text{non}}}$ and $\boldsymbol{Y}^{C_{\text{non}}}$. This property holds up to the end of the coupling. Thus e fails in type-II. Since $e \subseteq V_{\text{set}}$ and $v_k \neq v_0$, the while-loop must have picked a hyperedge e' and $v_k \in e'$ in Line 5. Thus, e' contains a variable v_j for $j < k$ (e' may not fail). If $j = 0$, then $e \in N^2(v_0)$, and the lemma holds for v_k with single hyperedge e. If $0 < j < k$, by induction hypothesis, there is a path $e_1, e_2, \ldots, e_t$ for v_j. Since $e_t \cap e' \neq \emptyset$ and $e' \cap e \neq \emptyset$, e and e_t are adjacent in $\text{Lin}^2(H)$. the lemma holds for v_k with the path $e_1, e_2, \ldots, e_t, e$.

Combining two cases proves the lemma. □

If the $X^{C_{\text{non}}}_{v_\star} \neq Y^{C_{\text{non}}}_{v_\star}$, we have the following result.

Lemma 8.15. *If $X_{v_\star}^{C_{\rm non}} \neq Y_{v_\star}^{C_{\rm non}}$, then there exists a path $e_1, e_2, \ldots, e_\ell$ in $\mathrm{Lin}^2(H)$ such that*

- *$e_1 \in N^2(v_0)$ and $v_\star \in e_\ell$;*
- *for all $1 \le i \le \ell - 1$, the hyperedge e_i fails in the coupling;*
- *the hyperedge e_ℓ is not satisfied by both $X_S^{C_{\rm non}}$ and $Y_S^{C_{\rm non}}$, where $S = e_\ell \setminus \{v_\star\}$.*

Proof. If $X_{v_\star}^{C_{\rm non}} \neq Y_{v_\star}^{C_{\rm non}}$, by Lemma 8.12, it must hold that $v_\star \in V_1$ and $v_\star$ is added into V_1 in Line 9, because $v_\star \neq v_0$, and if $v_\star$ is added into V_1 in Line 13, then $X_{v_\star}^{C_{\rm non}} = Y_{v_\star}^{C_{\rm non}}$. Consider the moment when $v_\star$ is added into V_1. Suppose the while-loop picks the hyperedge $e_\star$. It must hold that $v_\star \in e_\star$ and the while loop picks $v_\star$ to sample its values in $X^{C_{\rm non}}$ and $Y^{C_{\rm non}}$. In Line 5, the algorithm always picks the variable in $e_\star$ with lowest ID and the ID of $v_\star$ is the n. This implies all $(e_\star \cap V_2) \setminus V_{\rm set} = \{v_\star\}$. Note that $V_1 \subseteq V_{\rm set}$. Thus, all variables in $e_\star \setminus \{v_\star\}$ get the value and $e_\star$ is not satisfied in both $X_S^{C_{\rm non}}$ and $Y_S^{C_{\rm non}}$, where $S = e_\star \setminus \{v_\star\}$. Otherwise, $e_\star$ is removed in Line 3 or Line 11, the while-loop cannot pick $e_\star$.

Let $V_1 = \{v_0, v_1, v_2, \ldots, v_m\}$ denote the variables in V_1, where v_i is the i-th variables added into V_1. Remark that if a set of variables are added into V_1 at the same time (Line 13), we break tie arbitrarily. Suppose $v_\star = v_k$. Since $e_\star$ is picked in Line 5, it must hold that $v_j \in e_\star$ for some $j < k$. If $j = 0$, the lemma holds with single hyperedge $e_\star$. If $0 < j < k$, there exists a path $e_1, e_2, \ldots, e_{\ell-1}$ in $\mathrm{Lin}^2(H)$ satisfying the condition in Lemma 8.14 for v_j. Since $v_j \in e_{\ell-1}$ and $v_j \in e_\star$, the lemma holds with the path $e_1, e_2, \ldots, e_{\ell-1}, e_\star$. □

We modify the path in Lemma 8.15 to the following sequence of hyperedges, which will be used in the analysis.

Corollary 8.16. *If $X_{v_\star}^{C_{\rm non}} \neq Y_{v_\star}^{C_{\rm non}}$, then there exists a path $e_1, e_2, \ldots, e_\ell$ in $\mathrm{Lin}^3(H)$ such that*

- *$e_1 \in N^3(v_0)$, $v_\star \in e_\ell$, and $e_1, e_2, \ldots, e_\ell$ are mutually disjoint.*
- *for all $1 \le i \le \ell - 1$, the hyperedge e_i fails in the coupling;*
- *the hyperedge e_ℓ is not satisfied by both $X_S^{C_{\rm non}}$ and $Y_S^{C_{\rm non}}$, where $S = e_\ell \setminus \{v_\star\}$.*

Proof. Let $e'_1, e'_2, \ldots, e'_m$ denote the path in Lemma 8.15. We first show that how to construct the path $e_1, e_2, \ldots, e_\ell$ in $\mathrm{Lin}^3(H)$, then we show that such path satisfies the properties in the corollary.

Let $\mathcal{S}$ be an empty stack. Let P denote the sequence $(e'_1, e'_2, \ldots, e'_m)$. We pick the last hyperedge in the path P, denote this hyperedge as e'_i. We push e'_i into the stack $\mathcal{S}$. We search for the minimum index j such that $j < i$ and $e'_i \cap e'_j \neq \emptyset$. Here are two cases depending on whether such index j exists.

- If such index j does not exist, remove e'_i from the path P.
- If such index j exists, remove all e'_k for $j \le k \le i$ from the path P.

Repeat the above procedure until P becomes an empty sequence. Let $e_1, e_2, \ldots, e_\ell$ be the elements in stack $\mathcal{S}$ from top to bottom.

It is easy to verify $e_\ell = e'_m$. By Lemma 8.15, $v_\star \in e_\ell$ and e_ℓ satisfies the last property in the corollary. It is also easy to see all $e_1, e_2, \ldots, e_\ell$ are mutually disjoint. By Lemma 8.15, the hyperedge e_i fails in the coupling for all $1 \le i \le \ell - 1$. We only need to prove the following two properties

- $e_1 \in N^3(v_0)$;
- $e_1, e_2, \ldots, e_\ell$ forms a path in $\mathrm{Lin}^3(H)$.

We first prove $e_1 \in N^3(v_0)$. If $e_1 = e'_1$, then the property holds trivially. Suppose $e_1 = e'_k$ for some $k > 1$. When the procedure adds e'_k into the stack, the hyperedge e'_1 must be removed. This implies $e'_k \cap e'_1 \neq \emptyset$. By Lemma 8.15, $e'_1 \in N^2(v_0)$. It holds that $e_1 = e'_k \in N^3(v_0)$.

Next, we prove that $e_1, e_2, \ldots, e_\ell$ forms a path in $\mathrm{Lin}^3(H)$. Consider two adjacent hyperedges e_{i-1} and e_i. Suppose $e_i = e'_j$ and $e_{i-1} = e'_k$. If $j = k + 1$, since e'_j and e'_k are adjacent in $\mathrm{Lin}^2(H)$, e_i and e_{i-1} are adjacent in $\mathrm{Lin}^3(H)$. Suppose $j > k + 1$. In this case, e'_{k+1} is removed and e'_k is not removed, thus $e'_j \cap e'_{k+1} \neq \emptyset$. Since e'_k and e'_{k+1} are adjacent in $\mathrm{Lin}^2(H)$, e'_j and e'_k are adjacent in $\mathrm{Lin}^3(H)$. □

Fix a path $e_1, e_2, \ldots, e_\ell$ in $\mathrm{Lin}^3(H)$ such that it satisfies the first property except $v_\star \in e_\ell$ in Corollary 8.16, i.e. $e_1 \in N^3(v_0)$, and $e_1, e_2, \ldots, e_\ell$ are mutually disjoint. We call such path a *percolation path* (PP). We say a percolation path $e_1, e_2, \ldots, e_\ell$ is a percolation path for $v_\star$ if $v_\star \in e_\ell$.

Definition 8.17. Fix a percolation path $e_1, e_2, \ldots, e_\ell$. For each $1 \le i \le \ell$, a hyperedge e_i is *bad* if

- for $1 \le i \le \ell - 1$: the hyperedge e_i fails in the coupling C_{non} (Definition 8.13);
- for $i = \ell$: the hyperedge e_ℓ is not satisfied by both $X_S^{C_{\text{non}}}$ and $Y_S^{C_{\text{non}}}$, where $S = e_\ell \setminus \{v_\star\}$; and $v_\star$ is assigned different values in $\boldsymbol{X}^{C_{\text{non}}}$ and $\boldsymbol{Y}^{C_{\text{non}}}$, i.e. $X_{v_\star}^{C_{\text{non}}} \neq Y_{v_\star}^{C_{\text{non}}}$.

By Corollary 8.16, if $X_{v_\star}^{C_{\text{non}}} \neq Y_{v_\star}^{C_{\text{non}}}$ in coupling C_{non}, then there is a percolation path for $v_\star$: $e_1, e_2, \ldots, e_\ell$ such that e_i is bad for all $1 \le i \le \ell$. We give the following key lemma in this proof.

Lemma 8.18. *Suppose the original input CSP formula $\Phi = (V, \boldsymbol{Q}, C)$ of Algorithm 1 satisfies*

$$\log \frac{1}{p} \ge \frac{50}{\beta} \log \left(\frac{2000 D^4}{\beta} \right). \tag{59}$$

Fix a percolation path (PP) $e_1, e_2, \ldots, e_\ell$ for $v_\star$ in $\mathrm{Lin}^3(H)$. It holds that

$$\mathrm{Pr}_{C_{\text{non}}} [\forall 1 \le i \le \ell, e_i \text{ is bad}] \le \left(\frac{1}{4D^3} \right)^\ell \frac{\beta}{50} \left(\frac{1}{2} \right)^{\frac{\beta |e_\ell|}{50}},$$

which implies

$$\mathrm{Pr}_{C_{\text{non}}} \left[X_{v_\star}^{C_{\text{non}}} \neq Y_{v_\star}^{C_{\text{non}}} \right] \le \sum_{e_1, e_2, \ldots e_\ell \text{ is a PP for } v_\star} \left(\frac{1}{4D^3} \right)^\ell \frac{\beta}{50} \left(\frac{1}{2} \right)^{\frac{\beta |e_\ell|}{50}}.$$

The proof of Lemma 8.18 is deferred to Section 8.4.1. We now use Lemma 8.18 to prove Lemma 8.9.

Proof of Lemma 8.9. We will use Lemma 8.18 to show that

$$\sum_{v \in V \setminus \{v_0\}} d_{\mathrm{TV}} \left(\nu_v^{X_{V \setminus \{v\}}}, \nu_v^{Y_{V \setminus \{v\}}} \right) \le \frac{1}{2}.$$

By the assumption in Lemma 8.9, it holds that $\log \frac{1}{p} \ge \frac{50}{\beta} \log \left(\frac{2000 D^4}{\beta} \right)$. Note that the condition in Lemma 8.18 holds. Note that $\log \frac{1}{p} \ge \frac{50}{\beta} \log \left(\frac{2000 D^4}{\beta} \right) \ge \frac{55}{\beta} (\log D + 3)$. By Lemma 8.11, the projection schemes satisfying Condition 8.10 exists. By Lemma 8.12, the $\boldsymbol{X}^{C_{\text{non}}}$ in C_{non} follows the distribution ν_X and the $\boldsymbol{Y}^{C_{\text{non}}}$ in C_{non} follows the distribution ν_Y. By the definition of ν_X and ν_Y, it holds that $\nu_{v_\star, X} = \nu_{v_\star}^{X_{V \setminus \{v_\star\}}}$ and $\nu_{v_\star, Y} = \nu_{v_\star}^{Y_{V \setminus \{v_\star\}}}$. By the coupling lemma and Lemma 8.18, it holds that

$$d_{\mathrm{TV}} \left(\nu_{v_\star}^{X_{V \setminus \{v_\star\}}}, \nu_{v_\star}^{Y_{V \setminus \{v_\star\}}} \right) \le \mathrm{Pr}_{C_{\text{non}}} \left[X_{v_\star}^{C_{\text{non}}} \neq Y_{v_\star}^{C_{\text{non}}} \right] \le \sum_{e_1, e_2, \ldots e_\ell \text{ is a PP for } v_\star} \left(\frac{1}{4D^3} \right)^\ell \frac{\beta}{50} \left(\frac{1}{2} \right)^{\frac{\beta |e_\ell|}{50}}.$$

Note that the hypergraph H is same for any $v_\star \in V \setminus \{v_0\}$. We can use the above inequality with $v_\star = v$ for all $v \in V \setminus \{v_0\}$. Thus,

$$\begin{aligned} \sum_{v \in V \setminus \{v_0\}} d_{\mathrm{TV}} \left(\nu_v^{X_{V \setminus \{v\}}}, \nu_v^{Y_{V \setminus \{v\}}} \right) &\le \sum_{v \in V \setminus \{v_0\}} \sum_{e_1, e_2, \ldots e_\ell \text{ is a PP for } v} \left(\frac{1}{4D^3} \right)^\ell \frac{\beta}{50} \left(\frac{1}{2} \right)^{\frac{\beta |e_\ell|}{50}} \\ \text{(by double counting)} \quad &\le \sum_{e_1, e_2, \ldots e_\ell \text{ is a PP}} \left(\frac{1}{4D^3} \right)^\ell \frac{\beta |e_\ell|}{50} \left(\frac{1}{2} \right)^{\frac{\beta |e_\ell|}{50}}. \end{aligned}$$

Note that $x \left(\frac{1}{2} \right)^x \le 1$ for all $x \ge 0$. We have

$$\sum_{v \in V \setminus \{v_0\}} d_{\mathrm{TV}} \left(\nu_v^{X_{V \setminus \{v\}}}, \nu_v^{Y_{V \setminus \{v\}}} \right) \le \sum_{e_1, e_2, \ldots e_\ell \text{ is a PP}} \left(\frac{1}{4D^3} \right)^\ell.$$

If $e_1, e_2, \dots e_\ell$ is a percolation path, then $e_1, e_2, \dots e_\ell$ is a path in $\mathrm{Lin}^3(H)$ and $e_1 \in N^3(v_0)$. Note that $\left|N^3(v_0)\right| \le D + D(D-1) + D(D-1)^2 \le D^3$ (due to (58)) and the maximum degree of $\mathrm{Lin}^3(H)$ is at most D^3. The number of such paths is at most $D^{3\ell}$. We have

$$\sum_{v \in V \setminus \{v_0\}} d_{\mathrm{TV}}\left(\nu_v^{X_{V\setminus\{v\}}}, \nu_v^{Y_{V\setminus\{v\}}}\right) \le \sum_{e_1,e_2,\dots e_\ell \text{ is a PP}} \left(\frac{1}{4D^3}\right)^\ell \le \sum_{\ell=1}^{\infty} D^{3\ell}\left(\frac{1}{4D^3}\right)^\ell \le \frac{1}{2}. \qquad \square$$

8.4.1. *Proof of Lemma 8.18.* We first introduce some notations for proving Lemma 8.18. Let $\Phi = (V, \boldsymbol{Q}, C)$ to denote the original input CSP formula of Algorithm 1. Let D denote the maximum degree of the dependency graph of Φ. For each $v \in V$, let $q_v = |Q_v|$. Let

$$p \triangleq \max_{c \in C} \prod_{v \in \mathsf{vbl}(c)} \frac{1}{q_v}.$$

Let $\boldsymbol{h}$ denote the original projection scheme for Φ satisfying Condition 3.4 with parameters α and β. Recall that $\Phi^X = (V, \boldsymbol{Q}^X = (Q_u^X)_{u\in V}, C)$ and $\Phi^Y = (V, \boldsymbol{Q}^Y = (Q_v^Y)_{v\in V}, C)$ are defined in (54). Recall that $\boldsymbol{h}^X = (h_v^X)_{v\in V}$ and $\boldsymbol{h}^Y = (h_v^Y)_{v\in V}$ denote the projection schemes for Φ^X and Φ^Y, where $h_v^X : Q_v^X \to \Sigma_v'$ and $h_v^Y : Q_v^Y \to \Sigma_v'$. Recall that $\boldsymbol{h}^X$ and $\boldsymbol{h}^Y$ satisfy Condition 8.10. For each $v \in V$, $s_v^X = s_v^Y = s_v'$. The following lemma gives the key property for $\nu_{v,X}$ and $\nu_{v,Y}$ in Line 6.

Lemma 8.19. *Suppose the original input CSP formula Φ of Algorithm 1 satisfies*

$$\log\frac{1}{p} \ge \frac{50}{\beta}\log\left(\frac{2000D^4}{\beta}\right).$$

Let $\Lambda \subseteq V$ and $v \in V \setminus \Lambda$. Let $\sigma_X, \sigma_Y \in \Sigma_\Lambda' = \bigotimes_{u\in\Lambda}\Sigma_u'$ be two partial assignments on Λ. For any $c_X, c_Y \in \Sigma_v'$,

$$\frac{\left|(h_v^X)^{-1}(c_X)\right|}{q_v^X}\left(1 - \frac{\beta}{500D^3}\right) \le \nu_{v,X}(c_X \mid \sigma_X) \le \frac{\left|(h_v^X)^{-1}(c_X)\right|}{q_v^X}\left(1 + \frac{\beta}{500D^3}\right),$$
$$\frac{\left|(h_v^Y)^{-1}(c_Y)\right|}{q_v^Y}\left(1 - \frac{\beta}{500D^3}\right) \le \nu_{v,Y}(c_Y \mid \sigma_Y) \le \frac{\left|(h_v^Y)^{-1}(c_Y)\right|}{q_v^Y}\left(1 + \frac{\beta}{500D^3}\right).$$

Furthermore, if the variable v satisfies $\log\left\lfloor\frac{q_v^X}{s_v'}\right\rfloor \ge t + \frac{5}{4}\log\left(\frac{2000D^4}{\beta}\right)$ and $\log\left\lfloor\frac{q_v^Y}{s_v'}\right\rfloor \ge t + \frac{5}{4}\log\left(\frac{2000D^4}{\beta}\right)$ for some $t \ge 0$,then for any $c_X, c_Y \in \Sigma_v'$,

$$\frac{\left|(h_v^X)^{-1}(c_X)\right|}{q_v^X}\left(1 - \frac{\beta 2^{-t}}{500D^3}\right) \le \nu_{v,X}(c_X \mid \sigma_X) \le \frac{\left|(h_v^X)^{-1}(c_X)\right|}{q_v^X}\left(1 + \frac{\beta 2^{-t}}{500D^3}\right),$$
$$\frac{\left|(h_v^Y)^{-1}(c_Y)\right|}{q_v^Y}\left(1 - \frac{\beta 2^{-t}}{500D^3}\right) \le \nu_{v,Y}(c_Y \mid \sigma_Y) \le \frac{\left|(h_v^Y)^{-1}(c_Y)\right|}{q_v^Y}\left(1 + \frac{\beta 2^{-t}}{500D^3}\right).$$

Proof. We prove the lemma for $\nu_{v,X}(c_X \mid \sigma_X)$. The result for $\nu_{v,Y}(c_Y \mid \sigma_Y)$ can be proved in a similar way. To simplify the notation, denote $\sigma = \sigma_X$, $c^\star = c_X$. We define a new instance $\widetilde{\Phi} = (V, \widetilde{\boldsymbol{Q}} = (\widetilde{Q}_u)_{u\in V}, C)$:

$$\forall u \in V, \quad \widetilde{Q}_u = \begin{cases} (h_u^X)^{-1}(\sigma_u) & \text{if } u \in \Lambda; \\ Q_u^X & \text{if } u \notin \Lambda. \end{cases}$$

Let $\widetilde{\mu}$ denote the uniform distribution of all satisfying assignments to $\widetilde{\Phi}$. By the definition of the projected distribution, if $X \sim \widetilde{\mu}$, then $\Pr\left[X_v \in (h_v^X)^{-1}(c^\star)\right]$ equals to $\nu_{v,X}(c^\star \mid \sigma)$. By Condition 8.10, for any constraint $c \in C$, it holds that

$$\sum_{v\in\mathsf{vbl}(c)} \log\left\lfloor\frac{q_v^X}{s_v'}\right\rfloor \ge \frac{\beta}{10}\log\frac{1}{p} \ge 5\log\left(\frac{2000D^4}{\beta}\right). \tag{60}$$

Let $\mathcal{D}$ denote the product distribution such that each variable $u \in V$ takes a value from $\widetilde{Q}_u$ uniformly at random. For each constraint $c \in C$, let B_c denote the bad event that c is not satisfied. Let $\mathcal{B}$ denote the collection of bad events $(B_c)_{c\in C}$. Let $\Gamma(\cdot)$ be defined as in the Lovász local lemma (Theorem 2.1). We define a function $x : \mathcal{B} \to (0,1)$ such that

$$\forall c \in C \text{ s.t. } v \notin \mathsf{vbl}(c), \quad x(B_c) = \frac{\beta}{2000D^4};$$
$$\forall c \in C \text{ s.t. } v \in \mathsf{vbl}(c), \quad x(B_c) = \frac{\beta \lfloor q_v^X / s'_v \rfloor}{2000D^4 q_v^X}.$$

Since $\boldsymbol{h}^X$ is a balanced projection scheme, $\left|\widetilde{Q}_u\right| \geq \left\lfloor \frac{q_u^X}{s'_u} \right\rfloor$ for all $u \in V$. For any constraint $c \in C$ such that $v \notin \mathsf{vbl}(c)$, it holds that

$$\Pr_{\mathcal{D}}[B_c] = \prod_{u\in\mathsf{vbl}(c)} \frac{1}{|\widetilde{Q}_u|} \leq \prod_{u\in\mathsf{vbl}(c)} \frac{1}{\lfloor q_u^X/s'_u \rfloor} \leq \frac{\beta}{2000^5D^{20}} \leq \frac{\beta}{2000D^4}\left(1-\frac{\beta}{2000D^4}\right)^{2000D^4/\beta-1}$$
$$\leq \frac{\beta}{2000D^4}\left(1-\frac{\beta}{2000D^4}\right)^{D} \leq x(B_c)\prod_{B_{c'}\in\Gamma(B_c)}(1-x(B_{c'})), \tag{61}$$

where the last inequality holds because $x(B_c) \leq \frac{\beta}{2000D^4}$ for all $c \in C$. Note that $v \notin \Lambda$. For any $c \in C$ such that $v \in \mathsf{vbl}(c)$, by (60), it holds that

$$\Pr_{\mathcal{D}}[B_c] = \frac{1}{q_v^X}\prod_{u\in\mathsf{vbl}(c):u\neq v} \frac{1}{|\widetilde{Q}_u|} \leq \frac{\lfloor q_v^X/s'_v \rfloor}{q_v^X}\prod_{u\in\mathsf{vbl}(c)} \frac{1}{\lfloor q_u^X/s'_u \rfloor} \leq \frac{\lfloor q_v^X/s'_v \rfloor}{q_v^X}\cdot\frac{\beta}{2000^5D^{20}}$$
$$\leq \frac{\beta\lfloor q_v^X/s'_v \rfloor}{2000D^4q_v^X}\left(1-\frac{\beta}{2000D^4}\right)^{2000D^4/\beta-1} \leq \frac{\beta\lfloor q_v^X/s'_v \rfloor}{2000D^4q_v^X}\left(1-\frac{\beta}{2000D^4}\right)^{D}$$
$$\leq x(B_c)\prod_{B_{c'}\in\Gamma(B_c)}(1-x(B_{c'})).$$

Fix a value $c^\star \in \Sigma'_v$. Let A denote the event that v takes a value in $(h_v^X)^{-1}(c^\star)$. It holds that $|\Gamma(A)| \leq D$. For any $B_c \in \Gamma(A)$, it holds that $v \in \mathsf{vbl}(c)$ and $x(B_c) = \frac{\beta\lfloor q_v^X/s'_v \rfloor}{2000D^4q_v^X}$. Recall that $\widetilde{\mu}$ denotes the uniform distribution of all satisfying assignments to $\widetilde{\Phi}$. By Lovász local lemma (Theorem 2.1),

$$\Pr_{\widetilde{\mu}}[A] = \nu_{v,X}(c^\star \mid \sigma) \leq \frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\left(1-\frac{\beta\lfloor q_v^X/s'_v \rfloor}{2000D^4q_v^X}\right)^{-D} \leq \frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\exp\left(\frac{\beta\lfloor q_v^X/s'_v \rfloor}{1000D^3q_v^X}\right)$$
$$\leq \frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\left(1+\frac{\beta\lfloor q_v^X/s'_v \rfloor}{500D^3q_v^X}\right) \leq \frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\left(1+\frac{\beta}{500D^3}\right).$$

This proves the upper bound. Let A' denote the event that v does not take any value in $(h_v^X)^{-1}(c^\star)$, then $|\Gamma(A')| \leq D$. For any $B_c \in \Gamma(A')$, it holds that $v \in \mathsf{vbl}(c)$ and $x(B_c) = \frac{\beta\lfloor q_v^X/s'_v \rfloor}{2000D^4q_v^X}$. By Theorem 2.1,

$$\Pr_{\widetilde{\mu}}[A'] = 1-\nu_{v,X}(c^\star \mid \sigma) \leq \left(1-\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\right)\left(1-\frac{\beta\lfloor q_v^X/s'_v \rfloor}{2000D^4q_v^X}\right)^{-D}$$
$$\leq \left(1-\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\right)\exp\left(\frac{\beta\lfloor q_v^X/s'_v \rfloor}{1000D^3q_v^X}\right) \leq \left(1-\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\right)\left(1+\frac{\beta\lfloor q_v^X/s'_v \rfloor}{500D^3q_v^X}\right).$$

Let $a = \left|(h_v^X)^{-1}(c^\star)\right|/q_v^X$ and $b = \lfloor q_v^X/s'_v \rfloor/q_v^X$. Since $\boldsymbol{h}^X$ is a balanced projection scheme (Condition 8.10), it holds that $\left|(h_v^X)^{-1}(c^\star)\right| \geq \lfloor q_v^X/s'_v \rfloor$ and $a \geq b$. Thus

$$\nu_{v,X}(c^\star \mid \sigma) \geq 1-(1-a)\left(1+\frac{\beta b}{500D^3}\right) = a\left(1+\frac{\beta b}{500D^3}-\frac{\beta b}{500aD^3}\right) \geq a\left(1-\frac{\beta b}{500aD^3}\right)$$
$$(\text{by } a \geq b) \quad \geq a\left(1-\frac{\beta}{500D^3}\right) = \frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\left(1-\frac{\beta}{500D^3}\right). \tag{62}$$

This proves the lower bound.

Next, we assume

$$\log\left\lfloor\frac{q_v^X}{s'_v}\right\rfloor \geq t+\frac{5}{4}\log\left(\frac{2000D^4}{\beta}\right). \tag{63}$$

For each bad event B_c, we define a function $x:\mathcal{B}\to(0,1)$ such that

$$\forall c\in C \text{ s.t. } v\notin \mathsf{vbl}(c),\quad x(B_c)=\frac{\beta}{2000D^4};$$
$$\forall c\in C \text{ s.t. } v\in \mathsf{vbl}(c),\quad x(B_c)=\frac{\beta 2^{-t}\left\lfloor q_v^X/s'_v\right\rfloor}{2000D^4q_v^X}.$$

Note that for any $c\in C$, it holds that $x(B_c)\leq\frac{\beta}{2000D^4}$. By the same proof, for any constraint $c\in C$ such that $v\notin\mathsf{vbl}(c)$, (61) still holds. For any constraint $c\in C$ such that $v\in\mathsf{vbl}(c)$, we have

$$\begin{aligned}
\mathrm{Pr}_{\mathcal{D}}[B_c]&=\frac{1}{q_v^X}\prod_{u\in\mathsf{vbl}(c):u\neq v}\frac{1}{|\widetilde{Q}_u|}\leq\frac{\left\lfloor q_v^X/s'_v\right\rfloor}{q_v^X}\prod_{u\in\mathsf{vbl}(c)}\frac{1}{\left\lfloor q_u^X/s'_u\right\rfloor}\leq\frac{\left\lfloor q_v^X/s'_v\right\rfloor}{q_v^X}\frac{1}{\left\lfloor q_v^X/s'_v\right\rfloor}\\
\text{(by (63) and } \beta\leq 1)\quad&\leq\frac{\left\lfloor q_v^X/s'_v\right\rfloor}{q_v^X}\cdot\frac{\beta 2^{-t}}{2000^{5/4}D^5}\leq\frac{\beta 2^{-t}\left\lfloor q_v^X/s'_v\right\rfloor}{2000D^4q_v^X}\left(1-\frac{\beta}{2000D^4}\right)^{2000D^4/\beta-1}\\
&\leq\frac{\beta 2^{-t}\left\lfloor q_v^X/s'_v\right\rfloor}{2000D^4q_v^X}\left(1-\frac{\beta}{2000D^4}\right)^{D}\leq x(B_c)\prod_{B_{c'}\in\Gamma(B_c)}(1-x(B_{c'})).
\end{aligned}$$

Thus, the function $x:\mathcal{B}\to(0,1)$ satisfies the Lovász local lemma condition. By Theorem 2.1,

$$\begin{aligned}
\mathrm{Pr}_{\widetilde{\mu}}[A]=\nu_{v,X}(c^\star\mid\sigma)&\leq\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\left(1-\frac{\beta 2^{-t}\left\lfloor q_v^X/s'_v\right\rfloor}{2000D^4q_v^X}\right)^{-D}\leq\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\exp\left(\frac{\beta 2^{-t}\left\lfloor q_v^X/s'_v\right\rfloor}{1000D^3q_v^X}\right)\\
&\leq\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\left(1+\frac{\beta 2^{-t}\left\lfloor q_v^X/s'_v\right\rfloor}{500D^3q_v^X}\right)\leq\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\left(1+\frac{\beta 2^{-t}}{500D^3}\right).
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\mathrm{Pr}_{\widetilde{\mu}}[A']=1-\nu_{v,X}(c^\star\mid\sigma)&\leq\left(1-\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\right)\left(1-\frac{\beta 2^{-t}\left\lfloor q_v^X/s'_v\right\rfloor}{2000D^4q_v^X}\right)^{-D}\\
&\leq\left(1-\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\right)\exp\left(\frac{\beta 2^{-t}\left\lfloor q_v^X/s'_v\right\rfloor}{1000D^3q_v^X}\right)\leq\left(1-\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\right)\left(1+\frac{\beta 2^{-t}\left\lfloor q_v^X/s'_v\right\rfloor}{500D^3q_v^X}\right).
\end{aligned}$$

By the same proof in (62), we have

$$\nu_{v,X}(c^\star\mid\sigma)\geq\frac{\left|(h_v^X)^{-1}(c^\star)\right|}{q_v^X}\left(1-\frac{\beta 2^{-t}}{500D^3}\right).$$

□

Now, we are ready to prove Lemma 8.18. Fix a percolation path (PP) $e_1,e_2,\ldots,e_\ell$ in $\mathrm{Lin}^3(H)$. We bound the probability that all e_i are bad for $1\leq i\leq\ell$. Recall $s'_v=s_v^X=s_v^Y$ for all $v\in V$. For each hyperedge e_i, define

$$V(e_i)\triangleq\{v\in e_i\mid s'_v\neq 1\text{ and }v\neq v_0\}.$$

Note that for variables $v\in e_i\setminus(V(e_i)\cup\{v_0\})$, it must hold that $s'_v=\left|\Sigma'_v\right|=1$. It must hold that $X_v^{C_{\mathrm{non}}}=Y_v^{C_{\mathrm{non}}}$, which implies the coupling on v cannot be failed. Hence, if there is a variable $u\in e_i\backslash\{v_0\}$ such that $X_u^{C_{\mathrm{non}}}\neq Y_u^{C_{\mathrm{non}}}$, it must hold that $u\in V(e_i)$. In the while-loop, the coupling C_{non} assigns values to variables one-by-one, using the optimal coupling between marginal distributions. Let

$$k(e_i)\triangleq|V(e_i)|.$$

Fix an index $1\leq i\leq\ell-1$. Let $c(e_i)$ denote the constraint represented by e_i. We can define $k(e_i)+1$ bad events $B_i^{(j)}$ for $1\leq j\leq k(e_i)+1$:

- if $1 \le j \le k(e_i)$: the constraint $c(e_i)$ is not satisfied by both $\boldsymbol{X}^{C_{\text{non}}}$ and $\boldsymbol{Y}^{C_{\text{non}}}$ after $j-1$ variables in $V(e_i)$ are assigned values by C_{non}, and the coupling on j-th variable fails, i.e. $X_{v_j}^{C_{\text{non}}} \neq Y_{v_j}^{C_{\text{non}}}$, where $v_j \in V(e_i)$ is the j-th variable in $V(e_i)$ whose value is assigned by the coupling C_{non};
- if $j = k(e_i)+1$: the constraint $c(e_i)$ is not satisfied by both $\boldsymbol{X}^{C_{\text{non}}}$ and $\boldsymbol{Y}^{C_{\text{non}}}$ after all variables in e_i are assigned values by the coupling C_{non}.

Let B_i denote the event $\bigvee_{j=1}^{k(e_i)+1} B_i^{(j)}$. By Definition 8.17, we have the following relation

$$e_i \text{ is bad} \quad \Longleftrightarrow \quad e_i \text{ fails} \quad \Longrightarrow \quad B_i = \bigvee_{j=1}^{k(e_i)+1} B_i^{(j)}.$$

By Definition 8.13, if e_i fails in type-I, then there must exist $1 \le j \le k(e_i)$ such that the coupling of j-th variable in $V(e_i)$ fails and e_i is not satisfied by both $\boldsymbol{X}^{C_{\text{non}}}$ and $\boldsymbol{Y}^{C_{\text{non}}}$ after $j-1$ variables in $V(e_i)$ are assigned values (otherwise, e_i will be removed in Line 3 or Line 11). Hence, if e_i fails in type-I, $\bigvee_{j=1}^{k(e_i)} B_i^{(j)}$ must occur. If e_i fails in type-II, then $B_i^{(k(e_i)+1)}$ must occur. This proves the above relation.

For hyperedge e_ℓ, let $c(e_\ell)$ denote the constraint represented by e_ℓ, we define the bad event B_ℓ as

- B_ℓ: the constraint $c(e_\ell)$ is not satisfied by both $\boldsymbol{X}^{C_{\text{non}}}$ and $\boldsymbol{Y}^{C_{\text{non}}}$ after all variables in $e_\ell \setminus \{v_\star\}$ are assigned values by the coupling C_{non}, and the coupling on $v_\star$ fails, i.e. $X_{v_\star}^{C_{\text{non}}} \neq Y_{v_\star}^{C_{\text{non}}}$.

By Definition 8.17, we have the following relation

$$e_\ell \text{ is bad} \quad \Longrightarrow \quad B_\ell.$$

Let $\Omega_B = \bigotimes_{i=1}^{\ell-1} [k(e_i)+1]$, where $[k(e_i)+1] = \{1,2,\ldots,k(e_i)+1\}$. We have the following relation

$$\Pr_{C_{\text{non}}}[\forall 1 \le i \le \ell : e_i \text{ is bad}] \le \Pr_{C_{\text{non}}}[\forall 1 \le i \le \ell : B_i] \le \sum_{z \in \Omega_B} \Pr_{C_{\text{non}}}\left[B_\ell \wedge \forall 1 \le i \le \ell-1 : B_i^{(z_i)}\right],$$

where $z \in \Omega_B$ is a $(\ell-1)$-dimensional vector and $z_i \in [k(e_i)+1]$. Fix a vector $z \in \Omega_B$. Let

$$\mathcal{E}_1 = \{e_i \mid 1 \le i \le \ell-1 \wedge z_i \le k(e_i)\}$$
$$\mathcal{E}_2 = \{e_i \mid 1 \le i \le \ell-1 \wedge z_i = k(e_i)+1\}.$$

We will prove that

$$\begin{aligned} &\Pr_{C_{\text{non}}}\left[B_\ell \wedge \forall 1 \le i \le \ell-1 : B_i^{(z_i)}\right] \\ \le &\prod_{e_i \in \mathcal{E}_1}\left(\left(\frac{3}{4}\right)^{z_i-1}\frac{1}{200D^3}\right) \times \prod_{e_j \in \mathcal{E}_2}\left(\frac{1}{200D^3}\right) \times \left(\frac{\beta}{200D^3}\left(\frac{1}{2}\right)^{\frac{\beta|e_\ell|}{50}}\right). \end{aligned} \tag{64}$$

By (64), we have

$$\begin{aligned} \Pr_{C_{\text{non}}}[\forall 1 \le i \le \ell : e_i \text{ is bad}] &\le \sum_{z \in \Omega_B} \Pr_{C_{\text{non}}}\left[B_\ell \wedge \forall 1 \le i \le \ell-1 : B_i^{(z_i)}\right] \\ \text{(by (64))} \quad &\le \left(\frac{1}{200D^3} + \frac{1}{200D^3}\sum_{j=1}^{k(e_i)}\left(\frac{3}{4}\right)^{j-1}\right)^{\ell-1} \times \left(\frac{\beta}{200D^3}\left(\frac{1}{2}\right)^{\frac{\beta|e_\ell|}{50}}\right) \\ &\le \left(\frac{1}{40D^3}\right)^{\ell-1}\frac{\beta}{200D^3}\left(\frac{1}{2}\right)^{\frac{\beta|e_\ell|}{50}} \le \left(\frac{1}{4D^3}\right)^{\ell}\frac{\beta}{50}\left(\frac{1}{2}\right)^{\frac{\beta|e_\ell|}{50}}. \end{aligned}$$

This proves Lemma 8.18. The rest of this section is dedicated to the proof of (64).

Note that the RHS of (64) is a product. Although all hyperedges in a percolation path are mutually disjoint, we cannot show that all bad events $B_i^{(z_i)}$ and B_ℓ are mutually independent. Because all the bad events are defined by C_{non}, they may have some correlations with each other. To prove (64), we will use an independent random process to dominate the event that all $B_i^{(z_i)}$ and B_ℓ occur.

To prove (64), we first divide the bad event B_ℓ into two parts $B_\ell^{(1)}$ and $B_\ell^{(2)}$, where $B_\ell^{(1)}$ denotes the event that the constraint $c(e_\ell)$ is not satisfied by both $X_S^{C_{\text{non}}}$ and $Y_S^{C_{\text{non}}}$, where $S = e_\ell \setminus \{v_\star\}$, and $B_\ell^{(2)}$

denotes the event that the coupling on $v_\star$ fails, i.e. $X^{C_{\text{non}}}_{v_\star} \neq Y^{C_{\text{non}}}_{v_\star}$. It is easy to see $B_\ell = B^{(1)}_\ell \wedge B^{(2)}_\ell$. Note that $v_\star \in e_\ell$ and $q^X_{v_\star} = q^Y_{v_\star}$. By (57) in Condition 8.10, one of the following two conditions must be satisfied:

$$\min\left(\sum_{v\in\mathsf{vbl}(c)\setminus\{v_\star\}} \log \frac{q^X_v}{\lceil q^X_v/s'_v\rceil}, \sum_{v\in\mathsf{vbl}(c)\setminus\{v_\star\}} \log \frac{q^Y_v}{\lceil q^Y_v/s'_v\rceil}\right) \geq \frac{\beta}{20}\left(\sum_{v\in\mathsf{vbl}(c)} \log q_v\right), \tag{65}$$

$$\log\left\lfloor \frac{q^X_{v_\star}}{s'_{v_\star}}\right\rfloor = \log\left\lfloor \frac{q^Y_{v_\star}}{s'_{v_\star}}\right\rfloor \geq \frac{\beta}{20}\left(\sum_{v\in\mathsf{vbl}(c)} \log q_v\right). \tag{66}$$

If (65) holds, we can prove (64) by bounding the RHS of the following inequality

$$\Pr_{C_{\text{non}}}\left[B_\ell \wedge \forall 1\leq i\leq \ell-1 : B^{(z_i)}_i\right] \leq \Pr_{C_{\text{non}}}\left[B^{(1)}_\ell \wedge \forall 1\leq i\leq \ell-1 : B^{(z_i)}_i\right]. \tag{67}$$

If (66) holds, we can prove (64) by bounding the RHS of the following inequality

$$\Pr_{C_{\text{non}}}\left[B_\ell \wedge \forall 1\leq i\leq \ell-1 : B^{(z_i)}_i\right] \leq \Pr_{C_{\text{non}}}\left[B^{(2)}_\ell \wedge \forall 1\leq i\leq \ell-1 : B^{(z_i)}_i\right]. \tag{68}$$

In the rest of the proof, we mainly focus on the case when (65) holds. If (66) holds, we can modify our proof to bound the RHS of (68), this part will be discussed later.

Assume (65) holds. We start to bound the RHS of (67). To do this, we will give a particular implementation of the coupling C_{non} such that if $B^{(1)}_\ell$ and all $B^{(z_i)}_i$ occur, then some independent events must occur in our implementation and their probabilities are easy to bound. We first sample a set $\mathcal{R}$ of real numbers from $[0,1]$ uniformly and independently.

- For each $e_i \in \mathcal{E}_1$, sample $k(e_i)$ random real numbers $r_{e_i}(j) \in [0,1]$ for $1 \leq j \leq k(e_i)$ uniformly and independently.
- For each $e_i \in \mathcal{E}_2 \cup \{e_\ell\}$, for each variable $v \in e_i$, sample a random real number $r_v \in [0,1]$ uniformly and independently.

We then run the coupling C_{non} in Algorithm 4, but in some particular steps, we will use the random numbers in $\mathcal{R}$ to implement the sampling step in C_{non}.

We start from the special variable v_0. Note that if v_0 appears in the percolation path, then $v_0 \in e_1$. The coupling C_{non} will sample the values of v_0 in Line 1. We use the real number r_{v_0} to implement this sampling step if and only if $v_0 \in e_1$ and $e_1 \in \mathcal{E}_2$. Let $c(e_1)$ denote the constraint represented by e_1. Suppose $c(e_1)$ forbids the configuration $\sigma \in Q_{e_1}$, i.e. $(c(e_1))(\sigma) = \texttt{False}$. By definition, in Φ^X, $Q^X_{v_0} = h^{-1}_{v_0}(X_{v_0})$ and in Φ^Y, $Q^Y_{v_0} = h^{-1}_{v_0}(Y_{v_0})$. Note that $Q^X_{v_0} \neq Q^Y_{v_0}$. Thus, e_1 must be satisfied in Φ^X or Φ^Y, because it must hold that $\sigma_{v_0} \notin Q^X_{v_0}$ or $\sigma_{v_0} \notin Q^Y_{v_0}$. If e_1 is satisfied in both Φ^X and Φ^Y, then the hyperedge e_1 cannot be bad. We may assume e_1 is not satisfied in Φ^X (i.e. $\sigma_{v_0} \in Q^X_{v_0}$) and e_1 is satisfied in Φ^Y (i.e. $\sigma_{v_0} \notin Q^Y_{v_0}$). Otherwise, we can swap the roles of X and Y in the whole analysis. We use r_{v_0} to sample $X^{C_{\text{non}}}_{v_0}$ in Line 1 of C_{non}. Note that there is only one $j \in \Sigma'_{v_0}$ such that $\sigma_{v_0} \in (h^X_{v_0})^{-1}(j)$. We can set $X^{C_{\text{non}}}_{v_0} = j$ if $r_{v_0} \leq \nu_{v_0,X}(j)$. By Lemma 8.19, $\nu_{v_0,X}(j) \leq (1+\frac{1}{500D^3})\lceil q^X_{v_0}/s'_{v_0}\rceil/q^X_{v_0}$. Note that if $s'_{v_0} = 1$, then $\nu_{v_0,X}(j) = 1$, which implies $\nu_{v_0,X}(j) = 1 = (\frac{\lceil q^X_{v_0}/s'_{v_0}\rceil}{q^X_{v_0}})^{0.95}$. If $s'_{v_0} \geq 2$, then $\lceil q^X_{v_0}/s'_{v_0}\rceil/q^X_{v_0} \leq \lceil q^X_{v_0}/2\rceil/q^X_{v_0} \leq \frac{2}{3}$ (because $q^X_{v_0} \geq s'_{v_0} \geq 2$), which implies

$$\nu_{v_0,X}(j) \leq (1+\frac{1}{500D^3})\frac{\lceil q^X_{v_0}/s'_{v_0}\rceil}{q^X_{v_0}} \leq \frac{501}{500}\frac{\lceil q^X_{v_0}/s'_{v_0}\rceil}{q^X_{v_0}} \leq \left(\frac{\lceil q^X_{v_0}/s'_{v_0}\rceil}{q^X_{v_0}}\right)^{0.95}.$$

After Line 1, if e_1 is not satisfied by both $X^{C_{\text{non}}}_{v_0}$ and $Y^{C_{\text{non}}}_{v_0}$, then the following event must occur

$$r_{v_0} \leq \left(\frac{\lceil q^X_{v_0}/s'_{v_0}\rceil}{q^X_{v_0}}\right)^{0.95}. \tag{69}$$

During the while-loop of C_{non}, we maintain an index j_i for each hyperedge $e_i \in \mathcal{E}_1$. Initially, all $j_i = 0$. Suppose the coupling C_{non} picks a variable u in Line 5. Suppose $u \in e_i$ for some $1 \leq i \leq \ell$. Note that such hyperedge e_i is unique because all hyperedges in a percolation path are mutually disjoint. Let $c(e_i)$ denote the constraint represented by e_i. Suppose $c(e_i)$ forbids the configuration $\tau \in \mathbf{Q}_{e_i}$, i.e. $(c(e_i))(\tau) = \texttt{False}$. Since $u \neq v_0$, by Condition 8.10, it holds that $h_u^X = h_u^Y$. Let $c^\star \in \Sigma_u'$ denote the value such that $\tau_u \in (h_u^X)^{-1}(c^\star) = (h_u^Y)^{-1}(c^\star)$. We need to sample $c_x \in \Sigma_u'$ and $c_y \in \Sigma_u'$ from the optimal coupling between $\nu_{u,X}(\cdot \mid \mathbf{X}^{C_{\text{non}}})$ and $\nu_{u,Y}(\cdot \mid \mathbf{Y}^{C_{\text{non}}})$ in Line 6. By (47) and (48), the optimal coupling satisfies the following properties,

$$\begin{aligned}
\Pr\left[c_x = c_y\right] = \sum_{j \in \Sigma_u'} \Pr\left[c_x = c_y = j\right] &= \sum_{j \in \Sigma_u'} \min\left(\nu_{u,X}(j \mid \mathbf{X}^{C_{\text{non}}}), \nu_{u,Y}(j \mid \mathbf{Y}^{C_{\text{non}}})\right) \\
&= 1 - d_{\mathrm{TV}}\left(\nu_{u,X}(\cdot \mid \mathbf{X}^{C_{\text{non}}}), \nu_{u,Y}(\cdot \mid \mathbf{Y}^{C_{\text{non}}})\right), \\
\Pr\left[c_x = c^\star \vee c_y = c^\star\right] &= \max\left(\nu_{u,X}(c^\star \mid \mathbf{X}^{C_{\text{non}}}), \nu_{u,Y}(c^\star \mid \mathbf{Y}^{C_{\text{non}}})\right).
\end{aligned}$$

Let $t_{\max} \triangleq \max\left(\nu_{u,X}(c^\star \mid \mathbf{X}^{C_{\text{non}}}), \nu_{u,Y}(c^\star \mid \mathbf{Y}^{C_{\text{non}}})\right)$ and $d_{\mathsf{TV}} \triangleq d_{\mathrm{TV}}\left(\nu_{u,X}(\cdot \mid \mathbf{X}^{C_{\text{non}}}), \nu_{u,Y}(\cdot \mid \mathbf{Y}^{C_{\text{non}}})\right)$. Note that either $e_i \in \mathcal{E}_1$ or $e_i \in \mathcal{E}_2 \cup \{e_\ell\}$. We will use the following procedure to implement the sampling step in Line 6.

- Case $e_i \in \mathcal{E}_1$ and $u \in V(e_i)$. Set $j_i \leftarrow j_i + 1$ and let $r = r_{e_i}(j_i)$. If $j_i < z_i$, we sample c_x and c_y such that $c_x = c^\star \vee c_y = c^\star$ if and only if $r \leq t_{\max}$. If $j_i = z_i$, we sample c_x and c_y such that $c_x \neq c_y$ if and only if $r \leq d_{\mathsf{TV}}$. If $j_i > z_i$, we arbitrarily sample c_x and c_y from their optimal coupling.
- Case $e_i \in \mathcal{E}_2 \cup \{e_\ell\}$. Let $r = r_u$. Sample c_x and c_y such that $c_x = c^\star \vee c_y = c^\star$ if and only if $r \leq t_{\max}$.
- Otherwise, we do not use random numbers in $\mathcal{R}$ to implement the coupling.

We will use the following properties to analysis our implementation. Note that after we assigned the values to variable u, if $c(e_i)$ is not satisfied by both $X_u^{C_{\text{non}}}$ and $Y_u^{C_{\text{non}}}$, then it must hold that $c_x = c^\star$ or $c_y = c^\star$. Since $u \neq v_0$, by Condition 8.10, $Q_u^X = Q_u^Y$ and $h_u^X = h_u^Y$. By Lemma 8.19, we can prove the following properties. For any u with $s_u' > 1$, we have $q_u^X = q_u^Y \geq s_u' > 1$, thus

$$t_{\max} \leq \frac{\lceil q_u^X / s_u' \rceil}{q_u^X}\left(1 + \frac{1}{500D^3}\right) \leq \frac{\lceil q_u^X / 2 \rceil}{q_u^X}\left(1 + \frac{1}{500}\right) \leq \frac{2}{3}\left(1 + \frac{1}{500}\right) \leq \frac{3}{4}. \tag{70}$$

For any $u \in V \setminus \{v_0\}$, since $Q_u^X = Q_u^Y$ and $h_u^X = h_u^Y$, by Lemma 8.19, it holds that

$$t_{\max} \leq \min\left(1, \frac{\lceil q_u^X / s_u' \rceil}{q_u^X}\left(1 + \frac{1}{500D^3}\right)\right) \leq \min\left(1, \frac{501 \lceil q_u^X / s_u' \rceil}{500 q_u^X}\right) \leq \left(\frac{\lceil q_u^X / s_u' \rceil}{q_u^X}\right)^{0.95}; \tag{71}$$

$$d_{\mathsf{TV}} \leq \frac{1}{2} \sum_{j \in \Sigma_u'} \frac{\left|(h_u^X)^{-1}(j)\right|}{q_u^X}\left(\frac{2}{500D^3}\right) = \frac{1}{500D^3} \leq \frac{1}{200D^3}. \tag{72}$$

Inequality (71) can be proved by considering two cases. If $s_u' = 1$, then $\left(\frac{\lceil q_u^X / s_u' \rceil}{q_u^X}\right)^{0.95} = 1$, the inequality holds trivially. If $s_u' > 1$, then $\frac{\lceil q_u^X / s_u' \rceil}{q_u^X} \leq \frac{2}{3}$, this implies (71). To prove (72), note that $Q_u^X = Q_u^Y$ (thus, $q_u^X = q_u^Y$); and $\boldsymbol{h}^X$ and $\boldsymbol{h}^Y$ use the same way to map $Q_u^X = Q_u^Y$ to Σ_u' (i.e. $h_u^X = h_u^Y$). Hence, we can use the upper and lower bound in Lemma 8.19 to bound the total variation distance d_{TV}.

Consider a hyperedge $e_i \in \mathcal{E}_1$. If the event $B_i^{(z_i)}$ occurs, then by definition, $c(e_i)$ is not satisfied after $z_i - 1$ variables in $V(e_i)$ get the values and the coupling on z_i-th variable in $V(e_i)$ fails. Note that for all $v \in V(e_i)$, $s_v' > 1$. By (70) and (72), the bad event $B_i^{(z_i)}$ implies the following event:

- $\mathcal{A}_i$: for all $1 \leq j \leq z_i - 1$, $r_{e_i}(j) \leq \frac{3}{4}$ and $r_{e_i}(z_i) \leq \frac{1}{200D^3}$.

This bad event $\mathcal{A}_i$ occurs with probability

$$\Pr\left[\mathcal{A}_i\right] = \left(\frac{3}{4}\right)^{z_i - 1} \frac{1}{200D^3}. \tag{73}$$

Consider a hyperedge $e_i \in \mathcal{E}_2$. If the event $B_i^{(z_i)} = B_i^{(k(e_i)+1)}$ occurs, then by definition, $c(e_i)$ is not satisfied after all variables in e_i get the value. In our implementation, for any $v \in e_i$, we use r_v to sample values for $X_v^{C_{\text{non}}}$ and $Y_v^{C_{\text{non}}}$. By (69) and (71), the bad event $B_i^{(z_i)}$ implies

- $\mathcal{A}_i$: for all $v \in e_i$, $r_v \le \left(\frac{\lceil q_v^X/s'_v \rceil}{q_v^X}\right)^{0.95}$.

Since $e_i \in \mathcal{E}_2$, it holds that $v_\star \notin e_i$. By Condition 8.10 and (59), it holds that

$$\sum_{v \in e_i} \log \frac{q_v^X}{\lceil q_v^X/s'_v \rceil} \ge \frac{\beta}{10} \sum_{v \in e_i} \log q_v \ge 5 \log\left(\frac{2000D^4}{\beta}\right),$$

This bad event $\mathcal{A}_i$ occurs with probability

$$\Pr\left[\mathcal{A}_i\right] = \prod_{v \in e_i} \left(\frac{\lceil q_u^X/s'_u \rceil}{q_u^X}\right)^{0.95} \le \left(\frac{1}{2000D^{20}}\right)^{0.95} \le \frac{1}{200D^3}. \tag{74}$$

Consider the hyperedge e_ℓ. If the event $B_\ell^{(1)}$ occurs, then by definition, $c(e_\ell)$ is not satisfied after all variables in $e_i \setminus \{v_\star\}$ get the value. In our implementation, for any $v \in e_\ell$, we use r_v to sample values for $X_v^{C_{\text{non}}}$ and $Y_v^{C_{\text{non}}}$. By (69) and (71), the bad event $B_\ell^{(1)}$ implies

- $\mathcal{A}_\ell$: for all $v \in e_\ell \setminus \{v_\star\}$, $r_v \le \left(\frac{\lceil q_v^X/s'_v \rceil}{q_v^X}\right)^{0.95}$.

By (65), we have

$$\sum_{v \in e_\ell \setminus \{v_\star\}} \log \frac{q_v^X}{\lceil q_v^X/s'_v \rceil} \ge \frac{\beta}{20} \sum_{v \in e_\ell} \log q_v,$$

Note that in the original input CSP formula $\Phi = (V, \mathbf{Q}, C)$ of Algorithm 1, the domain size of each variable is at least 2 (otherwise,the value of such variable is fixed and we can remove such variable), it holds that $q_v \ge 2$ for all $v \in V$. This implies $\sum_{v \in e_\ell} \log q_v \ge |e_\ell|$. By (59), it holds that $\sum_{v \in e_\ell} \log q_v \ge \log \frac{1}{p} \ge \frac{50}{\beta} \log\left(\frac{2000D^4}{\beta}\right)$. We have

$$\sum_{v \in e_\ell \setminus \{v_\star\}} \log \frac{q_v^X}{\lceil q_v^X/s'_v \rceil} \ge \frac{\beta}{40} |e_\ell| + \frac{\beta}{40} \cdot \frac{50}{\beta} \log\left(\frac{2000D^4}{\beta}\right) = \frac{\beta}{40} |e_\ell| + \frac{5}{4} \log\left(\frac{2000D^4}{\beta}\right).$$

Hence, this bad event $\mathcal{A}_\ell$ occurs with probability

$$\Pr\left[\mathcal{A}_\ell\right] = \prod_{v \in e_\ell \setminus \{v_\star\}} \left(\frac{\lceil q_v^X/s'_v \rceil}{q_v^X}\right)^{0.95} \le \left(\frac{1}{2}\right)^{\frac{0.95\beta}{40}|e_\ell|} \cdot \left(\frac{\beta^{5/4}}{2000^{5/4}D^5}\right)^{0.95} \le \left(\frac{1}{2}\right)^{\frac{\beta}{50}|e_\ell|} \frac{\beta}{200D^3}, \tag{75}$$

where the last inequality holds because $\beta \le 1$.

Finally, if $B_\ell^{(1)}$ and all $B_i^{(z_i)}$ for $1 \le i \le \ell - 1$ occur, then $\mathcal{A}_i$ occurs for all $1 \le i \le \ell$. By definition, the event $\mathcal{A}_i$ is determined by a subset of random variables $S_i \subseteq \mathcal{R}$. For any $i \ne j$, the subset S_i and S_j are disjoint, thus all events $\mathcal{A}_i$ are mutually independent. Combining (67), (73), (74) and (75),

$$\begin{aligned}
\Pr_{C_{\text{non}}}\left[B_\ell \wedge \forall 1 \le i \le \ell - 1 : B_i^{(z_i)}\right] &\le \Pr_{C_{\text{non}}}\left[B_\ell^{(1)} \wedge \forall 1 \le i \le \ell - 1 : B_i^{(z_i)}\right] \\
&\le \Pr\left[\forall 1 \le i \le \ell, \mathcal{A}_i\right] = \prod_{i=1}^{\ell} \Pr\left[\mathcal{A}_i\right] \\
&\le \prod_{e_i \in \mathcal{E}_1} \left(\left(\frac{3}{4}\right)^{z_i - 1} \frac{1}{200D^3}\right) \times \prod_{e_j \in \mathcal{E}_2} \left(\frac{1}{200D^3}\right) \times \left(\frac{\beta}{200D^3}\left(\frac{1}{2}\right)^{\frac{\beta|e_\ell|}{50}}\right).
\end{aligned}$$

This proves (64) in case of (65).

Suppose the condition in (66) holds. In this case, we need to bound the RHS of (68). Compared with the above proof, the only difference is that we need to bound the probability of $B_\ell^{(2)}$, where $B_\ell^{(2)}$ denotes the coupling on $v_\star$ fails, i.e. $X_{v_\star}^{C_{\text{non}}} \ne Y_{v_\star}^{C_{\text{non}}}$. In this case, we have $\log\lfloor \frac{q_{v_\star}^X}{s'_{v_\star}} \rfloor = \log\lfloor \frac{q_{v_\star}^Y}{s'_{v_\star}} \rfloor \ge$

$\frac{\beta}{20}\left(\sum_{v\in e_\ell}\log q_v\right)$. Note that in the original input CSP formula of Algorithm 1, it holds that $q_v \geq 2$ for all $v \in V$. This implies $\sum_{v\in e_\ell}\log q_v \geq |e_\ell|$. By (59), it holds that $\sum_{v\in e_\ell}\log q_v \geq \log\frac{1}{p} \geq \frac{50}{\beta}\log\left(\frac{2000D^4}{\beta}\right)$. Thus, we have

$$\log\left\lfloor\frac{q^X_{v_\star}}{s'_{v_\star}}\right\rfloor = \log\left\lfloor\frac{q^Y_{v_\star}}{s'_{v_\star}}\right\rfloor \geq \frac{\beta}{20}\left(\sum_{v\in e_\ell}\log q_v\right) \geq \frac{\beta}{40}|e_\ell| + \frac{5}{4}\log\left(\frac{2000D^4}{\beta}\right).$$

Note that $Q^X_{v_\star} = Q^Y_{v_\star}$ and $h^X_{v_\star} = h^Y_{v_\star}$. In Lemma 8.19, we can set the parameter $t = \frac{\beta}{40}|e_\ell|$. This implies that when C_{non} couples $X^{C_{\mathrm{non}}}_{v_\star}$ and $Y^{C_{\mathrm{non}}}_{v_\star}$, the probability that the coupling fails is at most

$$\frac{1}{2}\sum_{j\in\Sigma'_u}\frac{\left|(h^X_{v_\star})^{-1}(j)\right|}{q^X_{v_\star}}\left(\frac{2\beta}{500D^3}\right)\left(\frac{1}{2}\right)^{\frac{\beta}{40}|e_\ell|} \leq \left(\frac{1}{2}\right)^{\frac{\beta}{50}|e_\ell|}\frac{\beta}{200D^3}.$$

The proof of this case is almost the same as the above proof. The only difference is that when coupling $v_\star$, we sample a random real number $r_{v_\star} \in [0,1]$ uniformly and independently. We use $r_{v_\star}$ to implement the coupling such that $X^{C_{\mathrm{non}}}_{v_\star} \neq Y^{C_{\mathrm{non}}}_{v_\star}$ only if $r_{v_\star} \leq \left(\frac{1}{2}\right)^{\frac{\beta}{50}|e_\ell|}\frac{\beta}{200D^3}$. We define the bad event $\mathcal{A}_\ell$ as $r_{v_\star} \leq \left(\frac{1}{2}\right)^{\frac{\beta}{50}|e_\ell|}\frac{\beta}{200D^3}$. By the same proof, we have

$$\begin{aligned}\Pr_{C_{\mathrm{non}}}\left[B_\ell \wedge \forall 1\leq i\leq \ell-1 : B_i^{(z_i)}\right] &\leq \Pr_{C_{\mathrm{non}}}\left[B_\ell^{(2)} \wedge \forall 1\leq i\leq \ell-1 : B_i^{(z_i)}\right]\\ &\leq \prod_{e_i\in\mathcal{E}_1}\left(\left(\frac{3}{4}\right)^{z_i-1}\frac{1}{200D^3}\right)\times\prod_{e_j\in\mathcal{E}_2}\left(\frac{1}{200D^3}\right)\times\left(\frac{\beta}{200D^3}\left(\frac{1}{2}\right)^{\frac{\beta|e_\ell|}{50}}\right).\end{aligned}$$

This proves (64) in case of (66).

8.4.2. *Proof of Lemma 8.11.* Without loss of generality, we assume $\left|Q^X_{v_0}\right| \leq \left|Q^Y_{v_0}\right|$. Otherwise, we can swap the roles of X and Y in this proof. Since the original projection scheme $\boldsymbol{h}$ is uniform,

$$0 \leq \left|Q^Y_{v_0}\right| - \left|Q^X_{v_0}\right| \leq 1. \tag{76}$$

We first construct the projection scheme $\boldsymbol{h}^X$ for Φ^X. To do this, we introduce a CSP formula $\widetilde{\Phi}^X = (V, \widetilde{\boldsymbol{Q}}^X = (\widetilde{Q}^X_v)_{v\in V}, C)$. We first construct a projection scheme $\widetilde{\boldsymbol{h}}^X$ for $\widetilde{\Phi}^X$, then transform $\widetilde{\boldsymbol{h}}^X$ to the projection scheme $\boldsymbol{h}^X$. Recall the original projection scheme is $\boldsymbol{h} = (h_v)_{v\in V}$, where $h_v : Q_v \to \Sigma_v$. Recall $q_v = |Q_v|$. The CSP formula $\widetilde{\Phi}^X$ is define as follows:

$$\widetilde{Q}^X_u = \begin{cases} h_u^{-1}(X_u) & \text{if } u \neq v_\star;\\ h_u^{-1}(j) & \text{if } u = v_\star,\end{cases}$$

where $j \in \Sigma_{v_\star}$ is an arbitrary value satisfying $\left|h^{-1}_{v_\star}(j)\right| = \lfloor q_{v_\star}/s_{v_\star}\rfloor$. For each $v \in V$, let $\widetilde{q}^X_v = \left|\widetilde{Q}^X_v\right|$. Let $\widetilde{p}$ denote $\max_{c\in C}\prod_{v\in\mathsf{vbl}(c)}\frac{1}{\widetilde{q}^X_v}$. By Condition 3.4, we have for any constraint $c \in C$,

$$\sum_{v\in\mathsf{vbl}(c)}\log\widetilde{q}^X_v \geq \beta\sum_{v\in\mathsf{vbl}(c)}\log q_v.$$

By the condition assumed in Lemma 8.11, it holds that

$$\log\frac{1}{\widetilde{p}} \geq \beta\log\frac{1}{p} \geq 55(\log D + 3). \tag{77}$$

Recall that the maximum degree of the dependency graph of $\widetilde{\Phi}^X$ is also D. We can use Theorem 3.8 on instance $\widetilde{\Phi}^X$ such that the parameter α and β in Theorem 3.8 are set as $\alpha = 8/9$ and $\beta = 1/9$. Remark that in the proof of Theorem 3.8, we use Lovász loca lemma to prove that the projection scheme described in theorem must exist. When $\alpha = 8/9$ and $\beta = 1/9$, the condition in Theorem 3.8 becomes

$$\log\frac{1}{\widetilde{p}} \geq \frac{25\cdot 9^3}{7^3}(\log D + 3).$$

This implies that under the condition in (77), there exists a balanced projection scheme $\widetilde{\boldsymbol{h}}^X = (\widetilde{h}_v^X)_{v\in V}$, where $\widetilde{h}_v^X : \widetilde{Q}_v^X \to \widetilde{\Sigma}_v^X$ and $\widetilde{s}_v^X = \left|\widetilde{\Sigma}_v^X\right|$ such that for any $c \in C$,

$$
\begin{aligned}
\sum_{v\in\mathsf{vbl}(c)} \log \frac{\widetilde{q}_v^X}{\lceil \widetilde{q}_v^X / \widetilde{s}_v^X \rceil} &\geq \left(1 - \frac{8}{9}\right) \sum_{v\in\mathsf{vbl}(c)} \log \widetilde{q}_v^X \geq \frac{\beta}{9} \sum_{v\in\mathsf{vbl}(c)} \log q_v;\\
\sum_{v\in\mathsf{vbl}(c)} \log \left\lfloor \frac{\widetilde{q}_v^X}{\widetilde{s}_v^X} \right\rfloor &\geq \frac{1}{9} \sum_{v\in\mathsf{vbl}(c)} \log \widetilde{q}_v^X \geq \frac{\beta}{9} \sum_{v\in\mathsf{vbl}(c)} \log q_v.
\end{aligned} \tag{78}
$$

Note that $\widetilde{\Phi}^X$ and Φ^X differ only at variable $v_\star$. Given the projection scheme $\widetilde{\boldsymbol{h}}^X$ and the original projection scheme $\boldsymbol{h}$, the projection scheme $\boldsymbol{h}^X$ can be constructed as follows

$$
h_u^X = \begin{cases} \widetilde{h}_u^X & \text{if } u \neq v_\star; \\ h_u & \text{if } u = v_\star. \end{cases}
$$

By definition, $\boldsymbol{h}^X$ is a balanced projection scheme and $h_{v_\star}^X = h_{v_\star}$. Since $\widetilde{\boldsymbol{h}}^X$ and $\boldsymbol{h}^X$ differ only at variable $v_\star$, for any constraint $c \in C$ such that $v_\star \notin \mathsf{vbl}(c)$, by (78),

$$
\begin{aligned}
\sum_{v\in\mathsf{vbl}(c)} \log \frac{q_v^X}{\lceil q_v^X / s_v^X \rceil} &= \sum_{v\in\mathsf{vbl}(c)} \log \frac{\widetilde{q}_v^X}{\lceil \widetilde{q}_v^X / \widetilde{s}_v^X \rceil} \geq \frac{\beta}{10} \sum_{v\in\mathsf{vbl}(c)} \log q_v;\\
\sum_{v\in\mathsf{vbl}(c)} \log \left\lfloor \frac{q_v^X}{s_v^X} \right\rfloor &= \sum_{v\in\mathsf{vbl}(c)} \log \left\lfloor \frac{\widetilde{q}_v^X}{\widetilde{s}_v^X} \right\rfloor \geq \frac{\beta}{10} \sum_{v\in\mathsf{vbl}(c)} \log q_v.
\end{aligned}
$$

For variable $v_\star$, it holds that $\lfloor q_{v_\star}^X / s_{v_\star}^X \rfloor = \lfloor q_{v_\star} / s_{v_\star} \rfloor = \widetilde{q}_{v_\star}^X$, because $\boldsymbol{h}^X$ uses the same way to partition $Q_{v_\star}$ as in the original projection scheme $\boldsymbol{h}$. Hence, for any constraint $c \in C$ such that $v_\star \in \mathsf{vbl}(c)$,

$$
\begin{aligned}
\sum_{v\in\mathsf{vbl}(c)} \log \left\lfloor \frac{q_v^X}{s_v^X} \right\rfloor &\geq \sum_{v\in\mathsf{vbl}(c)} \log \left\lfloor \frac{\widetilde{q}_v^X}{\widetilde{s}_v^X} \right\rfloor \geq \frac{\beta}{10} \sum_{v\in\mathsf{vbl}(c)} \log q_v;\\
\log \left\lfloor \frac{q_{v_\star}^X}{s_{v_\star}^X} \right\rfloor + \sum_{v\in\mathsf{vbl}(c)\setminus\{v_\star\}} \log \frac{q_v^X}{\lceil q_v^X / s_v^X \rceil} &= \log \left\lfloor \frac{q_{v_\star}^X}{s_{v_\star}^X} \right\rfloor + \sum_{v\in\mathsf{vbl}(c)\setminus\{v_\star\}} \log \frac{\widetilde{q}_v^X}{\lceil \widetilde{q}_v^X / \widetilde{s}_v^X \rceil}\\
\left(\text{by } \lfloor q_{v_\star}^X / s_{v_\star}^X \rfloor = \widetilde{q}_{v_\star}^X\right) \quad &\geq \sum_{v\in\mathsf{vbl}(c)} \log \frac{\widetilde{q}_v^X}{\lceil \widetilde{q}_v^X / \widetilde{s}_v^X \rceil}\\
&\geq \frac{\beta}{9} \sum_{v\in\mathsf{vbl}(c)} \log q_v \geq \frac{\beta}{10} \sum_{v\in\mathsf{vbl}(c)} \log q_v.
\end{aligned} \tag{79}
$$

This implies that $\boldsymbol{h}^X$ satisfies all the conditions in Condition 8.10.

Given the projection scheme $\boldsymbol{h}^X$, the projection scheme $\boldsymbol{h}^Y$ for Φ^Y can be defined as follows. For each variable $v \in V \setminus \{v_0\}$, $h_v^Y = h_v^X$. For variable v_0, we construct $\Sigma_{v_0}^Y = \Sigma_{v_0}^X$ and $s_{v_0}^Y = \left|\Sigma_{v_0}^Y\right|$, then arbitrarily map $Q_{v_0}^Y$ to $\Sigma_{v_0}^Y$ such that for any $j \in \Sigma_{v_0}^Y$, $\lfloor q_{v_0}^Y / s_{v_0}^Y \rfloor \leq \left|(h_{v_0}^Y)^{-1}(j)\right| \leq \lceil q_{v_0}^Y / s_{v_0}^Y \rceil$. It is easy to see $\boldsymbol{h}^Y$ is also a balanced projection scheme and $h_{v_\star}^Y = h_{v_\star}$. It is also easy to see $\Sigma_{v_0}^X = \Sigma_{v_0}^Y$, and $h_u^X = h_u^Y$ for all $u \in V \setminus \{v_0\}$. We now only need to verify that for any $c \in C$,

$$
\sum_{v\in\mathsf{vbl}(c)} \log \left\lfloor \frac{q_v^Y}{s_v^Y} \right\rfloor \geq \frac{\beta}{10} \sum_{v\in\mathsf{vbl}(c)} \log q_v; \tag{80}
$$

for any $c \in C$ satisfying $v_\star \notin \mathsf{vbl}(c)$,

$$
\sum_{v\in\mathsf{vbl}(c)} \log \frac{q_v^Y}{\lceil q_v^Y / s_v^Y \rceil} \geq \frac{\beta}{10} \left(\sum_{v\in\mathsf{vbl}(c)} \log q_v \right); \tag{81}
$$

and for any $c \in C$ satisfying $v_\star \in \mathsf{vbl}(c)$,

$$\log \left\lfloor \frac{q^Y_{v_\star}}{s^Y_{v_\star}} \right\rfloor + \sum_{v \in \mathsf{vbl}(c) \setminus \{v_\star\}} \log \frac{q^Y_v}{\lceil q^Y_v / s^Y_v \rceil} \geq \frac{\beta}{10} \left(\sum_{v \in \mathsf{vbl}(c)} \log q_v \right). \tag{82}$$

Note that for all $u \in V \setminus \{v_0\}$, it holds that $s^X_u = s^Y_u$ and $q^X_u = q^Y_u$. Also note that $s^X_{v_0} = s^Y_{v_0}$. If $q^X_{v_0} = q^Y_{v_0}$, (80), (81) and (82) hold trivially. By (76), we assume $q^Y_{v_0} = q^X_{v_0} + 1$. Since $q^Y_u \geq q^X_u$ and $s^X_u = s^Y_u$ for all $u \in V$, for any $c \in C$,

$$\sum_{v \in \mathsf{vbl}(c)} \log \left\lfloor \frac{q^Y_v}{s^Y_v} \right\rfloor \geq \sum_{v \in \mathsf{vbl}(c)} \log \left\lfloor \frac{q^X_v}{s^X_v} \right\rfloor \geq \frac{\beta}{10} \sum_{v \in \mathsf{vbl}(c)} \log q_v.$$

This proves (80). Note that for all $u \neq v_0$, $q^X_u = q^Y_u$ and $s^X_u = s^Y_u$. Also note that $v_\star \neq v_0$. It holds that

$$\left\lfloor \frac{q^Y_{v_\star}}{s^Y_{v_\star}} \right\rfloor = \left\lfloor \frac{q^X_{v_\star}}{s^X_{v_\star}} \right\rfloor \quad \text{and} \quad \forall v \in V \setminus \{v_0\}, \quad \frac{q^Y_v}{\lceil q^Y_v / s^Y_v \rceil} = \frac{q^X_v}{\lceil q^X_v / s^X_v \rceil}. \tag{83}$$

To prove (81) and (82), we only need to compare $\frac{q^X_{v_0}}{\lceil q^X_{v_0} / s^X_{v_0} \rceil}$ with $\frac{q^Y_{v_0}}{\lceil q^Y_{v_0} / s^Y_{v_0} \rceil}$. We claim

$$\frac{q^Y_{v_0}}{\lceil q^Y_{v_0} / s^Y_{v_0} \rceil} = \frac{q^X_{v_0} + 1}{\lceil (q^X_{v_0} + 1) / s^X_{v_0} \rceil} \geq \frac{1}{2} \frac{q^X_{v_0}}{\lceil q^X_{v_0} / s^X_{v_0} \rceil}. \tag{84}$$

By (78), (83) and (84), for any $c \in C$ such that $v_\star \notin \mathsf{vbl}(c)$, we have

$$\sum_{v \in \mathsf{vbl}(c)} \log \frac{q^Y_v}{\lceil q^Y_v / s^Y_v \rceil} \geq \left(\sum_{v \in \mathsf{vbl}(c)} \log \frac{q^X_v}{\lceil q^X_v / s^X_v \rceil} \right) - 1 \geq \frac{\beta}{9} \left(\sum_{v \in \mathsf{vbl}(c)} \log q_v \right) - 1 \geq \frac{\beta}{10} \left(\sum_{v \in \mathsf{vbl}(c)} \log q_v \right),$$

where the last inequality holds because $\beta \sum_{v \in \mathsf{vbl}(c)} \log q_v \geq \beta \log \frac{1}{p} \geq 55(\log D + 3) \geq 165$. This proves (81). Similarly, for any $c \in C$ such that $v_\star \in \mathsf{vbl}(c)$, we have

$$\begin{aligned} \log \left\lfloor \frac{q^Y_{v_\star}}{s^Y_{v_\star}} \right\rfloor + \sum_{v \in \mathsf{vbl}(c) \setminus \{v_\star\}} \log \frac{q^Y_v}{\lceil q^Y_v / s^Y_v \rceil} &\geq \log \left\lfloor \frac{q^X_{v_\star}}{s^X_{v_\star}} \right\rfloor + \left(\sum_{v \in \mathsf{vbl}(c) \setminus \{v_\star\}} \log \frac{q^X_v}{\lceil q^X_v / s^X_v \rceil} \right) - 1 \\ \text{(by (79))} \quad &\geq \frac{\beta}{9} \left(\sum_{v \in \mathsf{vbl}(c)} \log q_v \right) - 1 \geq \frac{\beta}{10} \left(\sum_{v \in \mathsf{vbl}(c)} \log q_v \right). \end{aligned}$$

To prove (84), we consider two case. Recall $s^X_{v_0} = s^Y_{v_0}$. If $q^X_{v_0}$ cannot be divided by $s^X_{v_0}$, then $\lceil (q^X_{v_0} + 1)/s^X_v \rceil = \lceil q^X_{v_0} / s^X_{v_0} \rceil$ and (84) holds trivially. If $q^X_{v_0}$ can be divided by $s^X_{v_0}$, then we need to show

$$\frac{q^X_{v_0} + 1}{1 + q^X_{v_0} / s^X_{v_0}} \geq \frac{1}{2} s^X_{v_0},$$

which is equivalent to $q^X_{v_0} \geq s^X_{v_0} - 2$, then (84) holds because $q^X_{v_0} \geq s^X_{v_0}$.

8.5. **Proofs of Lemma 5.2 and Lemma 5.6.** Lemma 5.2 is proved by combining Lemma 2.3, Proposition 8.1 and Lemma 8.9. Note that the condition in Lemma 5.2 is $\log \frac{1}{p} \geq \frac{50}{\beta} \log \left(\frac{2000 D^4}{\beta} \right)$, which suffices to imply the conditions in Proposition 8.1 and Lemma 8.9. This implies the Glauber dynamics has the unique stationary distribution ν and the mixing rate is $T_{\mathsf{mix}}(\varepsilon) \leq \lceil 2n \log \frac{n}{\varepsilon} \rceil$.

Lemma 5.6 is proved by combining Lemma 2.3, Proposition 8.1 and Lemma 8.2. Given a (k, d)-CSP formula, the maximum degree D of the dependency graph is at most dk, thus the condition in Proposition 8.1 becomes $k \log q \geq \frac{1}{\beta} \log(2edk)$. The condition in Lemma 5.6 is $k \log q \geq \frac{1}{\beta} \log \left(3000 q^2 d^6 k^6 \right)$, which suffices to imply the conditions in Proposition 8.1 and Lemma 8.2. This implies the Glauber dynamics has the unique stationary distribution ν and the mixing rate is $T_{\mathsf{mix}}(\varepsilon) \leq \lceil 2n \log \frac{n}{\varepsilon} \rceil$.

References

[AI16] Dimitris Achlioptas and Fotis Iliopoulos. Random walks that find perfect objects and the Lovász local lemma. *J. ACM*, 63(3):22, 2016.

[AIS19] Dimitris Achlioptas, Fotis Iliopoulos, and Alistair Sinclair. Beyond the lovász local lemma: Point to set correlations and their algorithmic applications. In *FOCS*, pages 725–744. IEEE, 2019.

[Alo91] Noga Alon. A parallel algorithmic version of the local lemma. *Random Struct. Algorithms*, 2(4):367–378, 1991. (Conference version in *FOCS*'91).

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

[BDK06] Magnus Bordewich, Martin E. Dyer, and Marek Karpinski. Stopping times, metrics and approximate counting. In *ICALP*, volume 4051 of *Lecture Notes in Computer Science*, pages 108–119. Springer, 2006.

[BDK08] Magnus Bordewich, Martin E. Dyer, and Marek Karpinski. Path coupling using stopping times and counting independent sets and colorings in hypergraphs. *Random Struct. Algorithms*, 32(3):375–399, 2008.

[Bec91] József Beck. An algorithmic approach to the Lovász local lemma. *Random Struct. Algorithms*, 2(4):343–365, 1991.

[BGG+19] Ivona Bezáková, Andreas Galanis, Leslie A. 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.

[CS00] Artur Czumaj and Christian Scheideler. Coloring nonuniform hypergraphs: a new algorithmic approach to the general Lovász local lemma. *Random Struct. Algorithms*, 17(3-4):213–237, 2000.

[EL75] Paul Erdős and László Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), Vol. II*, pages 609–627. Colloq. Math. Soc. János Bolyai, Vol. 10. 1975.

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

[FGYZ19] Weiming Feng, Heng Guo, Yitong Yin, and Chihao Zhang. Fast sampling and counting k-SAT solutions in the local lemma regime. *arXiv preprint arXiv:1911.01319*, 2019. (Full version).

[FGYZ20] Weiming Feng, Heng Guo, Yitong Yin, and Chihao Zhang. Fast sampling and counting k-SAT solutions in the local lemma regime. In *STOC*, pages 854–867. ACM, 2020.

[FM11] Alan M. Frieze and Páll Melsted. Randomly coloring simple hypergraphs. *Inf. Process. Lett.*, 111(17):848–853, 2011.

[FP01] Alan M. Frisch and Timothy J. Peugniez. Solving non-Boolean satisfiability problems with stochastic local search. In *IJCAI*, pages 282–290, 2001.

[GGGY20] Andreas Galanis, Leslie Ann Goldberg, Heng Guo, and Kuan Yang. Counting solutions to random CNF formulas. In *ICALP*, volume 168 of *LIPIcs*, pages 53:1–53:14, 2020.

[GJL19] Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the Lovász local lemma. *J. ACM*, 66(3):18:1–18:31, 2019. (Conference version in *STOC*'17).

[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. (Conference version in *STOC*'18).

[Har20] David G. Harris. New bounds for the Moser-Tardos distribution. *Random Struct. Algorithms*, 57(1):97–131, 2020.

[Har21] David G. Harris. Oblivious resampling oracles and parallel algorithms for the Lopsided Lovász Local Lemma. volume 17, pages Art. 1, 32, 2021. (Conference version in *SODA*'19).

[HH17] Bernhard Haeupler and David G. Harris. Parallel algorithms and concentration bounds for the lovász local lemma via witness dags. *ACM Trans. Algorithms*, 13(4):53:1–53:25, 2017. (Conference version in *SODA*'17).

[HS17a] David G. Harris and Aravind Srinivasan. Algorithmic and enumerative aspects of the Moser-Tardos distribution. *ACM Trans. Algorithms*, 13(3):Art. 33, 40, 2017. (Conference version in *SODA*'16).

[HS17b] David G. Harris and Aravind Srinivasan. A constructive Lovász local lemma for permutations. *Theory Comput.*, 13:Paper No. 17, 41, 2017. (Conference version in *SODA*'14).

[HS19] David G. Harris and Aravind Srinivasan. The Moser-Tardos framework with partial resampling. *J. ACM*, 66(5):Art. 36, 45, 2019. (Conference version in *FOCS*'13).

[HSS11] Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. New constructive aspects of the Lovász local lemma. *J. ACM*, 58(6):28, 2011. (Conference version in *FOCS*'10).

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

[HV20] Nicholas J.A. Harvey and Jan Vondrák. An algorithmic proof of the Lovász local lemma via resampling oracles. *SIAM J. Comput.*, 49(2):394–428, 2020. (Conference version in *FOCS*'15).

[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.

[KM11] Kolipaka Kashyap, Babu Rao and Szegedy Mario. Moser and Tardos meet Lovász. In *STOC*, pages 235–244, 2011.

[Kol18a] Vladimir Kolmogorov. Commutativity in the algorithmic Lovász local lemma. *SIAM J. Comput.*, 47(6):2029–2056, 2018. (Conference version in *FOCS*'16).

[Kol18b] Vladimir Kolmogorov. A faster approximation algorithm for the Gibbs partition function. In *COLT*, pages 228–249. PMLR, 2018.

[LKM03] Cong Liu, Andreas Kuehlmann, and Matthew W. Moskewicz. CAMA: A multi-valued satisfiability solver. In *ICCAD*, pages 326–333, 2003.

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

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

[Moi19] Ankur Moitra. Approximate counting, the Lovász local lemma, and inference in graphical models. *J. ACM*, 66(2):10:1–10:25, 2019. (Conference version in *STOC*'17).

[Mos09] Robin A. Moser. A constructive proof of the Lovász local lemma. In *STOC*, pages 343–350, 2009.

[MR98] Michael Molloy and Bruce Reed. Further algorithmic aspects of the local lemma. In *STOC*, pages 524–529, 1998.

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

[She85] James B. Shearer. On a problem of Spencer. *Combinatorica*, 5(3):241–245, 1985.

[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.

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