A Perfect Sampler for Hypergraph Independent Sets

Guoliang Qiu
Shanghai Jiao Tong University, China

Yanheng Wang[1]
ETH Zürich, Switzerland

Chihao Zhang
Shanghai Jiao Tong University, China

Abstract

The problem of uniformly sampling hypergraph independent sets is revisited. We design an efficient *perfect* sampler for the problem under a similar condition of the asymmetric Lovász local lemma. When specialized to d-regular k-uniform hypergraphs on n vertices, our sampler terminates in expected $O(n \log n)$ time provided $d \leq c \cdot 2^{\frac{k}{2}}$ where $c > 0$ is a constant, matching the rapid mixing condition for Glauber dynamics in Hermon, Sly and Zhang [10]. The analysis of our algorithm is simple and clean.

2012 ACM Subject Classification Theory of computation → Random walks and Markov chains

Keywords and phrases Coupling from the past, Markov chains, Hypergraph independent sets

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.103

Category Track A: Algorithms, Complexity and Games

Funding *Chihao Zhang*: The author is supported by National Natural Science Foundation of China Grant 61902241.

1 Introduction

The problem of uniformly sampling hypergraph independent sets, or equivalently the solutions of monotone CNF formulas, has been well-studied in recent years. Consider a hypergraph $\Phi = (V, \mathcal{C})$ on $|V| = n$ vertices. A set $S \subseteq V$ is an independent set if $C \cap S \neq C$ for all $C \in \mathcal{C}$. Assuming the hypergraph is d-regular and k-uniform, [10] showed that the natural Glauber dynamics mixes in $O(n \log n)$ time when $d \leq c \cdot 2^{\frac{k}{2}}$ for some constant $c > 0$. The sampler implies a fully polynomial-time randomized approximation scheme (FPRAS) for counting hypergraph independent sets [13]. On the other hand, it was shown in [1] that there is no FPRAS for the problem when $d \geq 5 \cdot 2^{\frac{k}{2}}$, unless $\mathbf{NP} = \mathbf{RP}$. Therefore, the result of [10] is tight up to a multiplicative constant.

The proof in [10] analyzes the continuous-time Glauber dynamics under the framework of *information percolation* developed in [14] for studying the cutoff phenomenon of the Ising model. In this framework, one can view the coupling history of a Markov chain as time-space slabs, and the failure of the coupling at time t as a discrepancy path percolating from t back to the beginning. The analysis of this structure can result in (almost) optimal bounds in many interesting settings.

The percolation analysis in [10] is technically complicated due to the *continuous* nature of the chain which leads to involved dependencies in both time and space. Recently, discrete analogs of the time-space slabs have been introduced in sampling solutions of general CNF

[1] Part of the work was done while the author was an undergraduate student at Shanghai Jiao Tong University.

49th International Colloquium on Automata, Languages, and Programming (ICALP 2022).
Editors: Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff;
Article No. 103; pp. 103:1–103:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

formulas in the Lovász local lemma regime [11, 8]. Notably in [8], an elegant discrete time-space structure, tailored to *systematic scan*, was introduced to support the analysis of *coupling from the past* (CFTP) paradigm [17]. In contrast to simulating Glauber dynamics, the method of CFTP allows to produce *perfect samples* (i.e. without approximation error) from stationary distribution.

In this article, we refine the discrete time-space structure in [8], which we call the *witness graph*, and apply it to the problem of sampling hypergraph independent sets. This leads to an efficient CFTP sampler that (1) outputs perfect samples; (2) matches the bound in [10] and operates also in the asymmetric case under Lovász local lemma-like condition; and (3) has significantly simpler analysis. More specifically, we study a natural grand coupling of the systematic scan for sampling hypergraph independent sets. To apply the method of CFTP, one needs to detect the coalescence of the grand coupling efficiently at each stage of the algorithm. The monotone property of hypergraph independent sets allows us to reduce the detection to arguing about percolation on the witness graph. This observation provides sufficient flexibility for us to carefully bound related probabilities. We first show that, under a condition similar to the *asymmetric* Lovász local lemma, a perfect sampler exists.

▶ **Theorem 1.** *Let $\mathcal{G}$ collects all hypergraphs $\Phi = (V, \mathcal{C})$ such that*

$$\forall C \in \mathcal{C} : 2|C| \cdot 2^{-|C|} \leq (1-\varepsilon) \cdot x(C) \cdot \prod_{C' \in \Gamma_{\Phi^2}(C)} (1 - x(C')),$$

for some constant $\varepsilon \in (0,1)$ and function $x : \mathcal{C} \to (0,1)$. Here $\Gamma_{\Phi^2}(C)$ denotes the set of hyperedges within distance 2 to C in Φ (excluding C itself).

There exists an algorithm that inputs a hypergraph $\Phi \in \mathcal{G}$ and outputs an independent set of Φ uniformly at random, in expected time

$$O\left(-\frac{1}{\log(1-\varepsilon)} \cdot \log\left(\sum_{C \in \mathcal{C}} \frac{x(C)}{1 - x(C)}\right) \cdot \sum_{C \in \mathcal{C}} \sum_{v \in C} d_v |C|\right),$$

where d_v is number of hyperedges in Φ that contains v.

In the proof of Theorem 1, we view a discrepancy path of the coupling percolating from time t back to the beginning as an object similar to the *witness tree* in [16] for certifying the non-termination of a randomized algorithm. This is an interesting analogy between an algorithm that *samples* the solutions of CNF formulas and an algorithm that *finds* them. We map the discrepancy paths in the witness graph to connected trees generated by *multi-type Galton-Watson process* in the hypergraph. As a result, a certain spatial mixing property implies the rapid mixing of the chain.

For k-uniform d-regular hypergraphs, the result in Theorem 1 translates to the condition $d \leq \frac{c \cdot 2^{k/2}}{k^{1.5}}$ for some constant $c > 0$ by choosing $x(C) = \frac{1}{d^2 k^2}$. We give a refined analysis for this symmetric case to remove the denominator:

▶ **Theorem 2.** *Let $\mathcal{G}$ collects all k-uniform d-regular hypergraphs $\Phi = (V, \mathcal{C})$ such that*

$$d \leq \left(\frac{1}{4}\sqrt{\frac{9-\varepsilon}{2}} - \frac{1}{2}\right) \cdot 2^{k/2}.$$

for some constant $\varepsilon \in (0,1)$. There exists an algorithm that inputs a hypergraph $\Phi \in \mathcal{G}$ and outputs a hypergraph independent set of Φ uniformly at random, in expected time $O\left(-\frac{1}{\log(1-\varepsilon)} \cdot k^2 d^2 n \cdot (\log n + \log d)\right)$ where $n := |V|$.

Compared to the result in [10], our sampler has the advantage of being perfect. Our analysis inductively enumerates discrepancy paths, taking into account the structure of the overlapping between hyperedges. Thanks to the clean structure of the witness graph, our proof is much simpler than the one in [10].

Relation to sampling solutions of general CNF formulas

The problem of sampling hypergraph independent sets is a special case of sampling the solutions of general CNF formulas or CSP instances which draw a lot of recent attention [15, 3, 11, 8, 12, 7, 6, 5, 4, 2, 9]. For general k-CNF formulas where each variable is of degree d (corresponding to the k-uniform d-regular hypergraphs here), the best condition needed for an efficient sampler is $d < \frac{1}{\mathrm{poly}(k)} \cdot 2^{\frac{k}{40/7}}$ [11, 8] which is much worse than the condition $d < c \cdot 2^{\frac{k}{2}}$ for hypergraph independent sets. A main reason is that for general CNF formulas, the state space of the local Markov chain is no longer connected, and therefore one needs to project the chain onto a sub-instance induced by some special set of variables. The loss of the projection step, however, is not well-understood. On the other hand, this is not an issue for hypergraph independent sets and therefore (almost) optimal bounds can be obtained.

We remark that the technique developed here for non-uniform graphs can also be applied to general CNF formulas to analyze the projection chain.

2 Preliminaries

2.1 Hypergraph independent sets

Recall in the introduction we mentioned that a set $S \subseteq V$ is an independent set of a hypergraph $\Phi = (V, \mathcal{C})$ if $S \cap C \neq C$ for every $C \in \mathcal{C}$. Sometimes we represent it as a (binary) coloring $\sigma \in \{0,1\}^V$, which is the indicator for S. That is, we say σ is an hypergraph independent set if $\forall C \in \mathcal{C}, \exists v \in C : \sigma(v) = 0$. Denoting by Ω_Φ the collection of all independent sets of Φ, our goal is to efficiently produce a sample from the uniform measure μ on Ω_Φ.

Let us fix some notations used throughout our discussion:

- We assume $V := [n]$ and $m := |\mathcal{C}|$. We denote the degree of a vertex $v \in V$ as $d_v := |\{C \in \mathcal{C} : v \in C\}|$ and the maximum degree as $d := \max_{v\in V} d_v$. A hypergraph is d-regular if every vertex is of degree d and is k-uniform if every hyperedge is of size k.
- For any $C \in \mathcal{C}$, we use $\Gamma^+_{\Phi^2}(C)$ to denote the set of hyperedges C' such that either $C \cap C' \neq \varnothing$ or there exists C^* such that $C \cap C^* \neq \varnothing$ and $C^* \cap C' \neq \varnothing$. Furthermore, let $\Gamma_{\Phi^2}(C) := \Gamma^+_{\Phi^2}(C) \setminus \{C\}$.
- For any coloring $\sigma \in \{0,1\}^V$, we use $\sigma^{v\leftarrow r}$ to denote the coloring obtained after recoloring $v \in V$ by $r \in \{0,1\}$ in σ.

2.2 Systematic scan and coupling from the past

Fix a hypergraph $\Phi = (V, \mathcal{C})$ and the uniform distribution μ over Ω_Φ. We use the so-called *systematic scan* Markov chain to sample independent sets from μ.

Let us define a transition map $f : \Omega_\Phi \times V \times \{0,1\} \to \Omega_\Phi$ by

$$f(\sigma; v, r) := \begin{cases} \sigma^{v\leftarrow r} & \text{if } \sigma^{v\leftarrow r} \in \Omega_\Phi; \\ \sigma & \text{otherwise.} \end{cases}$$

It takes the given coloring $\sigma \in \Omega_\Phi$ and tries to recolor vertex v with the proposed r. Next, we fix a deterministic *scan sequence* $v_1, v_2, \ldots$ where $v_i := (i \bmod n) + 1$ and write

$$F(\sigma; r_1, \ldots, r_t) := f(f(\cdots f(\sigma; v_1, r_1)\cdots); v_t, r_t).$$

Basically, it begins with the given coloring $\sigma \in \Omega_\Phi$ and runs f for t steps to update vertex colors. The vertices are updated periodically as specified by the scan sequence; the proposed colors for every steps are provided by the arguments $r_1, \ldots, r_t$. For a collection of initial states $S \subseteq \Omega_\Phi$, we use $F(S; r_1, \ldots, r_t)$ to denote the set $\{F(\sigma; r_1, \ldots, r_t) : \sigma \in S\}$.

▶ **Definition 3.** *The* systematic scan *is a Markov chain* (X_t) *defined by*

$$X_t := F(\sigma; R_1, \ldots, R_t) \quad \forall t$$

for some $\sigma \in \Omega_\Phi$ *and some independent* $\texttt{Bernoulli}(1/2)$ *variables* $R_1, R_2, \ldots$.

The systematic scan is not a time-homogeneous Markov chain. However, we can (and will) bundle every n steps into an atomic round so that the bundled version – denoting its transition matrix P_Φ – is homogeneous. It is easy to check that P_Φ is irreducible and aperiodic with stationary distribution μ.

Given a Markov chain with stationary μ, in the usual Markov chain Monte Carlo method, one obtains a sampler for μ as follows: Starting from some initial X_0, simulate the chain for t steps with sufficiently large t and output X_t. The fundamental theorem of Markov chains says that if the chain is finite, irreducible and aperiodic, then the distribution of X_t converges to μ when t approaches infinity. However, since we always terminate the simulation after some fixed finite steps, the sampler obtained is always an "approximate" sampler instead of a "perfect" one.

The work of [17] proposed an ingenious method called *coupling from the past* (CFTP) to simulate the given chain in a reverse way with a random stopping time. A perfect sampler for μ can be obtained in this way. Roughly speaking, it essentially simulates an infinite long chain using only finitely many steps. To achieve this, it relies upon a routine to detect whether the (finite) simulation coalesces with the virtual infinite chain. Once the coalescence happens, one can output the result of the simulation – which is also the result of the infinite chain and thus follows the stationary distribution perfectly.

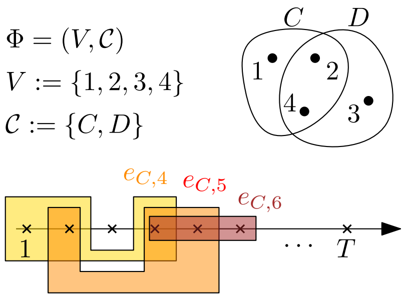
We use CFTP to simulate our systematic scan and obtain a desired perfect sampler for hypergraph independent sets. The key ingredient to apply CFTP is how to detect the coalescence. We describe and analyze our algorithm in Section 3.

3 Perfect Sampler via Information Percolation

In this section, we describe our perfect sampler for hypergraph independent sets. With the help of a data structure introduced in [8], we apply the argument of information percolation to analyze the algorithm. We utilize the monotonicity of the hypergraph independent sets and establish a sufficient condition for the algorithm to terminate.

3.1 The witness graph

In this section, we introduce the notion of witness graph $H_T = (V_T, E_T)$ for the systematic scan up to some time $T \in \mathbb{N}$. A similar structure was used in [8] for sampling general CNF formulas.



$e_{C,1} = \{1\}$	
$e_{C,2} = \{1,2\}$	$e_{D,2} = \{2\}$
	$e_{D,3} = \{2,3\}$
$e_{C,4} = \{1,2,4\}$	$e_{D,4} = \{2,3,4\}$
$e_{C,5} = \{5,2,4\}$	
$e_{C,6} = \{5,6,4\}$	$e_{D,6} = \{6,3,4\}$
	$e_{D,7} = \{6,7,4\}$
$e_{C,8} = \{5,6,8\}$	$e_{D,8} = \{6,7,8\}$
...	...

Figure 1 An illustration of the witness graph. In this example, the hypergraph Φ has four vertices and two hyperedges. We list the vertices of the witness graph in the table on the right. The lower left picture visualizes three vertices in H_T; all of them have label C.

Given a vertex $v \in V$, we denote its last update time up to moment t as

$$\texttt{UpdTime}(v,t) = \max\{t^* \le t : v_{t^*} = v\}.$$

Clearly $\texttt{UpdTime}(v,t) \in (t-n, t]$ is a deterministic number.

For $C \in \mathcal{C}$ and $t \in [T]$, let $e_{C,t} := \{\texttt{UpdTime}(v,t) : v \in C\}$ be timestamps when the elements in C got their latest updates up to time t. Conversely, we say $e_{C,t}$ has *label* C and denote it by $C(e_{C,t}) := C$.

The vertex set of witness graph is given by

$$V_T := \{e_{C,t} : t \in [T], v_t \in C \in \mathcal{C}\}.$$

We put a directed edge $e_{C,t} \to e_{C',t'}$ into E_T if and only if $t' \in (e_{C,t} \cap e_{C',t'}) \setminus \{t\}$. Note that H_T is acyclic, since $\max e > \max e'$ for any directed edge $e \to e'$. We remark that the witness graph is a deterministic object which by itself does not incorporate any randomness.

The following lemma measures the number of vertices labelled by a certain $C \in \mathcal{C}$ that are 2-distant from a given vertex in the witness graph. It is useful throughout the enumeration in later discussions.

▶ **Lemma 4.** *For any $e_{C,t} \in V_T$ and $C' \in \mathcal{C}$, we have $|e_{C',t'} \in V_T : \texttt{dist}(e_{C,t}, e_{C',t'}) = 2| \le 2|C'|$.*

Proof. If $C' \notin \Gamma^+_{\Phi^2}(C)$ then $|e_{C',t'} \in V_T : \texttt{dist}(e_{C,t}, e_{C',t'}) = 2| = 0$. Otherwise, $t' \in [t-2n+2, t)$, and there are at most $2|C'|$ timestamps $e_{C',t'}$ satisfying $t' \in [t-2n+2, t)$. ◀

3.2 Coalescence and percolation

For any $L \in \mathbb{N}$ we fix $T = T(L) := n(L+1)$. On top of the witness graph H_T we define a probability space as follows. We tie an independent $\texttt{Bernoulli}(1/2)$ variable B_t to each time point $t \in [T]$. We say a vertex $e \in V_T$ is *open* if $B_t = 1$ for all $t \in e$, and call a set of vertices $P \subseteq V$ *open* if all $e \in P$ are open. The event $\mathcal{B}_L$ is defined as

> "H_T contains an induced open path $P = (e_1, \ldots, e_L) \subseteq V_T$ of length L where $e_1 \cap (T-n, T] \ne \varnothing$".

For the coming few pages, the notation $\mathcal{B}_L(R_1, \ldots, R_T)$ indicates that we are using (external) random variables $R_1, \ldots, R_T$ as *concrete realizations* of our abstract variables $B_1, \ldots, B_T$. Needless to say, $R_1, \ldots, R_T$ themselves should be independent $\texttt{Bernoulli}(1/2)$ for such notation to make sense.

Starting from Section 4, however, we will switch back to the abstract setting and sweep the concrete realization under the rug.

Recall F is the transition map for our systematic scan.

▶ **Lemma 5.** *Assume $L \in \mathbb{N}$, $T := n(L+1)$ and $R_1, \dots, R_T$ are independent* $\texttt{Bernoulli}(1/2)$ *variables. If $|F(\Omega_\Phi; R_1, \dots, R_T)| > 1$ then $\mathcal{B}_L = \mathcal{B}_L(R_1, \dots, R_T)$ happens.*

To prove Lemma 5 we specify a *grand coupling*, namely a family of Markov chains that share the same random sequence $R_1, \dots, R_T$ provided by the lemma.

For every $\sigma \in \Omega_\Phi$, define a copy of systematic scan $(X_{\sigma,t})_{0 \le t \le T}$ by

$$X_{\sigma,t} := F(\sigma, R_1, \dots, R_t) \quad \forall 0 \le t \le T.$$

In addition, we define an auxiliary Markov chain $(Y_t)_{0 \le t \le T}$ by

$$Y_0 := \{1\}^V, \qquad Y_t := Y_{t-1}^{v_t \leftarrow R_t} \quad \forall 1 \le t \le T.$$

The chain $(Y_t)_{0 \le t \le T}$ dominates the execution of $(X_{\sigma,t})_{0 \le t \le T}$ by monotonicity:

▶ **Proposition 6.** *For all $\sigma \in \Omega_\Phi$ and $0 \le t \le T$, we have $X_{\sigma,t} \le Y_t$.*

Proof. Initially, $X_{\sigma,0} = \sigma \le \{1\}^V = Y_0$ for all $\sigma \in \Omega_\Phi$. At any time $t \ge 1$, all the chains update the same vertex v_t and (i) if $R_t = 0$ then $X_{\sigma,t}(v_t) = 0$; (ii) if $R_t = 1$ then $Y_t(v_t) = 1$. So the ordering $X_{\sigma,t} \le Y_t$ is preserved throughout. ◀

▶ **Proposition 7.** *Let $0 \le t \le T$ be a time point. If there exist $\sigma, \tau : X_{\sigma,t}(v_t) \ne X_{\tau,t}(v_t)$, then there is a hyperedge $C \in \mathcal{C}$ containing v_t such that $C \setminus \{v_t\}$ was fully colored by "1" in exactly one of $X_{\sigma,t}$ and $X_{\tau,t}$. Furthermore, $Y_t(C) = \{1\}^C$.*

Proof. Since $X_{\sigma,t}(v_t) \ne X_{\tau,t}(v_t)$, we know that $R_t = 1$ and thus $Y_t(v_t) = 1$. According to the definition of transition map f, one of the two chains failed to recolor v_t by "1" because there exists a hyperedge C such that $C \setminus \{v_t\}$ was fully colored "1" in that chain just before time t. Hence $Y_t(C \setminus \{v_t\}) = \{1\}^{C \setminus \{v_t\}}$ by Proposition 6. Putting together we have $Y_t(C) = \{1\}^C$. ◀

We are now ready to prove Lemma 5.

Proof of Lemma 5. Assume $|F(\Omega_\Phi; R_1, \dots, R_T)| > 1$; that is, there exist $\sigma, \tau \in \Omega_\Phi$ and $v \in V$ such that $X_{\sigma,T}(v) \ne X_{\tau,T}(v)$. We inductively construct a path in H_T as follows:

1. Set $i \leftarrow 1$. Let $t_1 := \texttt{UpdTime}(v, T) \in (T-n, T]$.
2. While $t_i \ge n$ do the following. Regarding Proposition 7 and the fact $X_{\sigma,t_i}(v_{t_i}) \ne X_{\tau,t_i}(v_{t_i})$, there is a hyperedge $C_i \in \mathcal{C}$ containing v_{t_i} such that $C_i \setminus \{v_{t_i}\}$ was fully colored by "1" in exactly one of X_{σ,t_i} and X_{τ,t_i}. So we may find an earliest time $t_{i+1} \in e_i := e_{C_i,t_i}$ such that $X_{\sigma,t_{i+1}}(v_{t_{i+1}}) \ne X_{\tau,t_{i+1}}(v_{t_{i+1}})$ and $t_{i+1} < t_i$. Moreover, $Y_{t_i}(C_i) = \{1\}^{C_i}$ implies that e_i is open. Let $i \leftarrow i+1$ and repeat.

Note that $t_{i+1} \in e_i$ by definition. On the other hand $t_{i+1} \in e_{i+1} = e_{C_{i+1},t_{i+1}}$ since $v_{t_{i+1}} \in C_{i+1}$. Combining with the condition $t_{i+1} < t_i$, we see $t_{i+1} \in (e_i \cap e_{i+1}) \setminus \{t_i\}$ and consequently $e_i \to e_{i+1}$ is indeed an edge in H_T. Therefore the above procedure returns an open (not necessarily induced) path $P^\circ = (e_1, \dots, e_r) \subseteq V_T$ where r is the number of rounds. It starts at vertex e_1 that intersects $(T-n, T]$, and ends at vertex e_r that intersects $[1, n)$.

Let $P \subseteq P^\circ$ be a *shortest path* from e_1 to e_r in $H_T[P^\circ]$. It must be an induced path in H_T since H_T is acyclic. It is open since $P \subseteq P^\circ$. Finally, every vertex $e \in P$ "spans" a time interval of at most n, so the length of P is at least $T/n - 1 = L$. ◀

3.3 The perfect sampler

We are ready to introduce our perfect sampler for hypergraph independent sets. Let $L \in \mathbb{N}$ be a parameter to be fixed later (roughly in the order of $O(\log m)$) and $T = T(L) := n(L+1)$ as usual. The algorithm follows the standard framework of CFTP.

Algorithm 1 The CFTP sampler of hypergraph independent sets.

Input: A hypergraph $\Phi = (V, \mathcal{C})$.
Output: An independent set of Φ sampled uniformly from μ.

$j \leftarrow 0$;
repeat
  $j \leftarrow j+1$;
  generate independent $\texttt{Bernoulli}(1/2)$ variables $R_{-jT+1}, \ldots, R_{-(j-1)T}$;
until *not* $\texttt{Detect}(\Phi, R_{-jT+1}, \ldots, R_{-(j-1)T})$;
return $F\left(\{0\}^V; R_{-jT+1}, \ldots, R_0\right)$.

Algorithm 2 The Detect subroutine.

Input: A hypergraph $\Phi = (V, \mathcal{C})$ and a sequence $R_1, \ldots, R_T$.
Output: Whether $\mathcal{B}_L(R_1, \ldots, R_T)$ happens.

foreach $C \in \mathcal{C}$ **do**
  foreach $t \in [T] : v_t \in C$ **do**
    foreach $v \in C \setminus \{v_t\}$ **do**
      foreach $C' \in \mathcal{C} : v \in C' \neq C$ **do**
        $t' \leftarrow \texttt{UpdTime}(v, t)$;
        connect $e_{C,t} \to e_{C',t'}$ if both of them are open with regard to $R_1, \ldots, R_T$;
breadth-first search from all $e_{C,t} : C \in \mathcal{C}, t \in (T-n, T]$ until the current vertex intersects $[1, n)$;
return *if the path has length* $\geq L$.

▶ **Theorem 8.** *Suppose there exist constant $\varepsilon \in (0,1)$ and β such that $\mathbf{Pr}[\mathcal{B}_L] \leq \beta \cdot (1-\varepsilon)^L$ for all L. With the concrete choice $L := \left\lceil \frac{\log(2\beta)}{-\log(1-\varepsilon)} \right\rceil$, Algorithm 1 terminates with probability 1 and has expected running time $O\left(\frac{\log(2\beta)}{-\log(1-\varepsilon)} \cdot \sum_{C \in \mathcal{C}} \sum_{v \in C} d_v |C|\right)$. Moreover, its output distribution ν is exactly μ upon termination.*

Proof. The $\texttt{Detect}$ subroutine (Algorithm 2) essentially constructs the witness graph H_T and decides if $\mathcal{B}_L(R_1, \ldots, R_T)$ happens. Since the inputs fed by Algorithm 1 are always independent $\texttt{Bernoulli}(1/2)$ variables as required, we have

$$p := \mathbf{Pr}\left[\texttt{Detect}(\Phi, R_{-jT+1}, \ldots, R_{(-j-1)T})\right] = \mathbf{Pr}[\mathcal{B}_L] \leq \beta \cdot (1-\varepsilon)^L = \frac{1}{2}.$$

where the inequality follows from the assumption of the theorem.

Note that Algorithm 1 feeds disjoint (hence independent) sequences into Algorithm 2 in different rounds, so the corresponding return values are independent. Therefore, the total number of rounds J follows geometric distribution $\texttt{Geom}(1-p)$. In particular $\mathbf{Pr}[J < \infty] = 1$ and $\mathbf{E}[J] = \frac{1}{1-p} \leq 2$. So the algorithm terminates with probability 1.

Next we analyze the running time of a single call to Algorithm 2. Each iteration of the nested loop can be implemented in constant time: We index each vertex $e_{C,t}$ by the pair (C,t) to allow random access; the test of openness can be computed and stored beforehand to facilitate fast lookup. So the nested loop takes

$$O\left(\sum_{C\in\mathcal{C}}|C|\cdot L\cdot\sum_{v\in C}d_v\right)$$

time to finish. The standard breadth-first search consumes time proportional to the number of edges in the witness graph, which is bounded by the same quantity.

Finally, recall that Algorithm 1 calls Algorithm 2 J times. But $\mathbf{E}[J]\leq 2$, so the expected running time of our sampler is $O\left(L\cdot\sum_{C\in\mathcal{C}}\sum_{v\in C}d_v|C|\right)$.

We could now turn to show $\nu=\mu$ as advertised. Upon termination, the percolation event $\mathcal{B}_L(R_{-JT+1},\ldots,R_{-(J-1)T})$ does not happen and hence $\left|F(\Omega_\Phi;R_{-JT+1},\ldots,R_{-(J-1)T})\right|=1$ by Lemma 5. But this in particular means

$$|F(\Omega_\Phi;R_{-JT+1},\ldots,R_0)|=1. \tag{1}$$

For each integer $s\geq 0$, we define a (virtual) copy $(Z_t^s)_{-sn\leq t\leq 0}$ of systematic scan which starts from $\{0\}^V$ at time $-sn$:

$$Z_t^s:=F\left(\{0\}^V;R_{-sn+1},\ldots,R_t\right).$$

Let ν^s denote the distribution of Z_0^s. Then $\lim_{s\to\infty}\nu^s=\mu$ by convergence theorem. On the other hand,

$$\begin{aligned}\|\nu^s-\nu\| &\leq \mathbf{Pr}\left[F\left(\{0\}^V;R_{-sn+1},\ldots,R_0\right)\neq F\left(\{0\}^V;R_{-JT+1},\ldots,R_0\right)\right] && \text{(coupling)}\\ &\leq \mathbf{Pr}\left[sn<JT\right] && \text{(by (1))}\\ &\to 0. && (\text{as } s\to\infty)\end{aligned}$$

Hence $\nu=\lim_{s\to\infty}\nu^s=\mu$. ◀

3.4 Proofs of Theorem 1 and Theorem 2

Armed with Theorem 8, we only need to bound $\mathbf{Pr}[\mathcal{B}_L]$ under various conditions. We bound the quantity in Section 4. As a result, Theorem 1 follows from Lemma 9 and Theorem 2 follows from Lemma 13.

4 Percolation on Witness Graphs

Let $\Phi=(V,\mathcal{C})$ be a hypergraph with $|V|=n$ and let H_T be its witness graph for any $T=(L+1)n$, $L\in\mathbb{N}$. We will analyze the probability $\mathbf{Pr}[\mathcal{B}_L]$ in the abstract probability space, where each time point $t\in[T]$ is associated with an independent $\texttt{Bernoulli}(1/2)$ variable.

4.1 General hypergraphs

▶ **Lemma 9.** *If there exists a constant $\varepsilon\in(0,1)$ and a function $x:\mathcal{C}\to(0,1)$ satisfying*

$$\forall C\in\mathcal{C}: 2|C|\cdot 2^{-|C|}\leq(1-\varepsilon)\cdot x(C)\cdot\prod_{C'\in\Gamma_{\Phi^2}(C)}(1-x(C')),$$

then $\mathbf{Pr}[\mathcal{B}_L]\leq\sum_{C\in\mathcal{C}}\frac{x(C)}{1-x(C)}\cdot(1-\varepsilon)^{\lfloor\frac{L+1}{2}\rfloor}$ for all $L\in\mathbb{N}$.

For every $C \in \mathcal{C}$, we use $\mathcal{P}_{C,L}$ to denote the collection of induced paths $(e_1, e_2, \ldots, e_L)$ in H_T where $e_1 \cap (T-n, T] \neq \varnothing$ and the label of e_1 is C (namely $C(e_1) = C$). Let

$$\mathcal{P}^2_{C,\lfloor\frac{L+1}{2}\rfloor} := \left\{(e_1, e_3, e_5 \ldots, e_{2\lfloor\frac{L-1}{2}\rfloor+1}) : (e_1, e_2, e_3 \ldots, e_L) \in \mathcal{P}_C\right\}.$$

Then by the union bound, we have

$$\begin{aligned}\mathbf{Pr}\left[\mathcal{B}_L\right] &\leq \sum_{C\in\mathcal{C}}\sum_{P\in\mathcal{P}_{C,L}} \mathbf{Pr}\left[P \text{ is open}\right] \leq \sum_{C\in\mathcal{C}}\sum_{P\in\mathcal{P}^2_{C,\lfloor\frac{L+1}{2}\rfloor}} \mathbf{Pr}\left[P \text{ is open}\right] \\ &= \sum_{C\in\mathcal{C}}\sum_{(e_1,\ldots,e_{\lfloor\frac{L+1}{2}\rfloor})\in\mathcal{P}^2_{C,\lfloor\frac{L+1}{2}\rfloor}} \prod_{i=1}^{\lfloor\frac{L+1}{2}\rfloor} 2^{-|C(e_i)|},\end{aligned} \tag{2}$$

where the last equality follows from the fact that $e_i \cap e_{i+1} = \varnothing$ holds for every pair of consecutive vertices of a path in $\mathcal{P}^2_{C,\lfloor\frac{L+1}{2}\rfloor}$.

▶ **Lemma 10.** *If there exists a function $x : \mathcal{C} \to (0,1)$ satisfying*

$$\forall C \in \mathcal{C} : 2|C| \cdot 2^{-|C|} \leq (1-\varepsilon)\cdot x(C) \cdot \prod_{C'\in\Gamma_{\Phi^2}(C)} (1 - x(C')),$$

then

$$\sum_{C\in\mathcal{C}}\sum_{(e_1,\ldots,e_{\lfloor\frac{L+1}{2}\rfloor})\in\mathcal{P}^2_{C,\lfloor\frac{L+1}{2}\rfloor}} \prod_{i=1}^{\lfloor\frac{L+1}{2}\rfloor} 2^{-|C(e_i)|} \leq \sum_{C\in\mathcal{C}} \frac{x(C)}{1-x(C)} \cdot (1-\varepsilon)^{\lfloor\frac{L+1}{2}\rfloor}.$$

Lemma 9 is clearly a consequence of Lemma 10. We prove the latter by analyzing a *multi-type Galton-Watson branching process.*

4.1.1 Multi-type Galton-Watson branching process

Recall $\Phi = (V, \mathcal{C})$ is a fixed hypergraph and we assume that there exists a function $x : \mathcal{C} \to (0,1)$ assigning each $C \in \mathcal{C}$ a number in $(0,1)$. A $\mathcal{C}$-labelled tree is a tuple $\tau = (V_\tau, E_\tau, \mathcal{L})$ where $\mathcal{L} : V_\tau \to \mathcal{C}$ labels each vertex in V_τ with a hyperedge in $\mathcal{C}$.

Let $C \in \mathcal{C}$ be a hyperedge. Consider the following process which generates a random $\mathcal{C}$-labelled tree with the root labelled with C:

- First produce a root vertex u with label C. Initialize the active set A as $\{u\}$.
- Repeat the following until A is empty:
  - Pick some $u \in A$.
  - For every $C' \in \Gamma^+_{\Phi^2}(\mathcal{L}(u))$: create a new child for u labelled with C' *with probability* $x(C')$ independently; Add the new child to A.
  - Remove u from A.

We let $\mathcal{T}_C$ be the set of all labelled trees that can be generated by the above process and let $\mu_{\mathcal{T}_C}$ be the distribution over $\mathcal{T}_C$ induced by the process.

▶ **Lemma 11.** *For every labelled tree $\tau = (V_\tau, E_\tau, \mathcal{L}) \in \mathcal{T}_C$, it holds that*

$$\mu_{\mathcal{T}_C}(\tau) = \frac{1-x(C)}{x(C)} \cdot \prod_{v\in V_\tau} x(\mathcal{L}(v)) \cdot \prod_{C'\in\Gamma_{\Phi^2}(\mathcal{L}(v))} (1 - x(C')).$$

Proof. For a vertex $v \in V_\tau$ we use $W_v \subseteq \Gamma^+_{\Phi^2}(\mathcal{L}(v))$ to denote the set of labels that do not occur as a label of some child nodes of v. Then clearly, the probability that the Galton-Watson branching process produces exactly the tree τ is given by

$$\mu_{\mathcal{T}_C}(\tau) = \frac{1}{x(C)} \prod_{v \in V_\tau} x(\mathcal{L}(v)) \prod_{C' \in W_v} (1 - x(C')).$$

In order to get rid of the W_v, we can rewrite the expression as

$$\begin{aligned}\mu_{\mathcal{T}_C}(\tau) &= \frac{1 - x(C)}{x(C)} \prod_{v \in V_\tau} \frac{x(\mathcal{L}(v))}{1 - x(\mathcal{L}(v))} \prod_{C' \in \Gamma^+_{\Phi^2}(\mathcal{L}(v))} (1 - x(C')) \\ &= \frac{1 - x(C)}{x(C)} \prod_{v \in V_\tau} x(\mathcal{L}(v)) \prod_{C' \in \Gamma_{\Phi^2}(\mathcal{L}(v))} (1 - x(C')).\end{aligned}$$ ◀

For every $\ell \geq 1$, we also use $\mathcal{T}_{C,\ell}$ to denote the set of labelled trees in $\mathcal{T}_C$ containing exactly ℓ vertices. In the following, we slightly abuse notation and say that a labelled tree $\tau = (V_\tau, E_\tau, \mathcal{L}) \in \mathcal{T}_C$ (or $\mathcal{T}_{C,\ell}$) if there exists some $\tau' \in \mathcal{T}_C$ (or $\mathcal{T}_{C,\ell}$) such that τ and τ' are isomorphic.

4.1.2 Proof of Lemma 10

Let $\ell = \lfloor \frac{L+1}{2} \rfloor$. It follows from the assumption that for every $C \in \mathcal{C}$,

$$\begin{aligned}&\sum_{P=(e_1,e_2,\ldots,e_\ell) \in \mathcal{P}^2_{C,\ell}} \prod_{i=1}^{\ell} 2^{-|C(e_i)|} \\ &\quad \leq (1-\varepsilon)^\ell \cdot \sum_{P=(e_1,e_2,\ldots,e_\ell) \in \mathcal{P}^2_{C,\ell}} \prod_{i=1}^{\ell} \frac{1}{2|C(e_i)|} \cdot x(C(e_i)) \prod_{C' \in \Gamma_{\Phi^2}(C(e_i))} (1 - x(C')).\end{aligned} \tag{3}$$

We now define a mapping $\Psi : \mathcal{P}^2_{C,\ell} \to \mathcal{T}_{C,\ell}$ that maps each $P = (e_1, e_2, \ldots, e_\ell)$ to a labelled directed path $\tau = (V_\tau, E_\tau, \mathcal{L})$ where

- $V_\tau = \{u_1, u_2, \ldots, u_\ell\}$;
- $E_\tau = \{(u_i, u_{i+1}) : i \in [\ell - 1]\}$; and
- for every $i \in [\ell]$, $\mathcal{L}(u_i) = C(e_i)$.

The mapping Ψ is not necessarily an injection, but we can bound its multiplicity.

▶ **Lemma 12.** *For every labelled tree $\tau = (V_\tau, E_\tau, \mathcal{L}) \in \mathcal{T}_{C,\ell}$ with $V_\tau = \{u_1, \ldots, u_\ell\}$ and $E_\tau = \{(u_i, u_{i+1}) : i \in [\ell - 1]\}$, it holds that*

$$\left|\Psi^{-1}(\tau)\right| \leq 2^{\ell-1} \cdot \prod_{i=1}^{\ell} |\mathcal{L}(u_i)|.$$

Proof. We prove it by induction:

1. When $\ell = 1$, the lemma holds because there are at most $|C|$ timestamps e with label C such that $e \cap (T - n + 1, T] \neq \varnothing$;
2. We assume the statement holds for $\ell \geq 1$. Let $V_\tau = \{u_1, u_2, \ldots, u_\ell, u_{\ell+1}\}$ and $V_{\tau'} = \{u_1, u_2, \ldots, u_\ell\}$. According to the definition of Ψ, we know that for all $P \in \Psi^{-1}(\tau)$, it was extended from some $P' \in \Psi^{-1}(\tau')$. Therefore, it suffices to analyze the possible extension from P' to $P \in \Psi^{-1}(\tau)$. Assuming $P' = (e_1, e_2, \ldots, e_\ell)$ and $P = \{e_1, e_2, \ldots, e_\ell, e_{\ell+1}\}$, if P

was extended from P', then $\mathtt{dist}(e_\ell, e_{\ell+1}) = 2$ and $C(e_{\ell+1}) = \mathcal{L}(u_{\ell+1})$. Therefore, there are at most $2|\mathcal{L}(u_{\ell+1})|$ timestamps $e_{\ell+1}$ satisfying $P \in \Psi^{-1}(\tau)$ according to Lemma 4. Combining with the induction hypothesis, the statement holds for $\ell + 1$. ◀

Clearly for those $\tau \in \mathcal{T}_{C,\ell}$ which are not directed paths, its pre-image $\Psi^{-1}(\tau) = \varnothing$. Therefore, it follows from Lemma 11 and Lemma 12 that

$$\begin{aligned}
&\sum_{P=(e_1,e_2,\dots,e_\ell)\in\mathcal{P}^2_{C,\ell}} \prod_{i=1}^{\ell} \frac{1}{2|C(e_i)|} \cdot x(C(e_i)) \prod_{C'\in\Gamma_{\Phi^2}(C(e_i))} (1 - x(C')) \\
&\le \sum_{\substack{\tau=(V_\tau,E_\tau,\mathcal{L})\in\mathcal{T}_{C,\ell}:\\ V_\tau=\{u_1,\dots,u_\ell\}}} \prod_{i=1}^{\ell} x(\mathcal{L}(u_i)) \prod_{C'\in\Gamma_{\Phi^2}(\mathcal{L}(u_i))} (1 - x(C')) \\
&= \frac{x(C)}{1 - x(C)} \sum_{\tau\in\mathcal{T}_{C,\ell}} \mu_{\mathcal{T}_C}(\tau) \\
&< \frac{x(C)}{1 - x(C)}.
\end{aligned}$$

4.2 Refined analysis for uniform hypergraphs

If the instance $\Phi = (V, \mathcal{C})$ is a d-regular k-uniform hypergraph, we can choose $x(C) = \frac{1}{d^2k^2}$ in Lemma 9 and the condition becomes $d \le \frac{c\cdot 2^{k/2}}{k^{1.5}}$ for some constant $c > 0$. In this section, we present a refined analysis for this case which removes the denominator $k^{1.5}$.

▶ **Lemma 13.** *For all $\varepsilon \in (0, 1)$ and $k \ge 2$, if*

$$d \le \left(\frac{1}{4}\sqrt{\frac{9-\varepsilon}{2}} - \frac{1}{2}\right) \cdot 2^{k/2}$$

then $\mathbf{Pr}[\mathcal{B}_L] \le \frac{k}{2^k} \cdot m \cdot (1 - \varepsilon)^{\lfloor \frac{L}{2} - 1 \rfloor}$ for all $L \in \mathbb{N}$.

For every induced path $(u_1, \dots, u_\ell)$ in H_T, and $L > 0$, we use $\mathcal{P}_{(u_1,\dots,u_\ell),L}$ to denote the collection of induced paths in H_T of length L with prefix $(u_1, \dots, u_\ell)$[2].

▶ **Lemma 14.** *For all $\varepsilon \in (0, 1)$ and $k \ge 2$, if*

$$d \le \left(\frac{1}{4}\sqrt{\frac{9-\varepsilon}{2}} - \frac{1}{2}\right) \cdot 2^{k/2}$$

then for every $u \in V_{H_T}$ and every $L \in \mathbb{N}$,

$$\mathbf{Pr}\left[\exists \text{ open } P \in \mathcal{P}_{(u),L} \;\middle|\; u \text{ is open}\right] \le (1 - \varepsilon)^{\lfloor \frac{L}{2} - 1 \rfloor}.$$

We first show that Lemma 13 follows from Lemma 14.

Proof of Lemma 13. By the union bound, we have

$$\begin{aligned}
\mathbf{Pr}[\mathcal{B}_L] &\le \sum_{C\in\mathcal{C}} \sum_{e_{C,t}:t\in(T-n,T]} \mathbf{Pr}\left[\exists \text{ open } P \in \mathcal{P}_{(e_{C,t}),L}\right] \\
&= \sum_{C\in\mathcal{C}} \sum_{e_{C,t}:t\in(T-n,T]} \mathbf{Pr}\left[\exists \text{ open } P \in \mathcal{P}_{(e_{C,t}),L} \;\middle|\; e_{C,t} \text{ is open}\right] \cdot \mathbf{Pr}\left[e_{C,t} \text{ is open}\right] \\
&\le km(1 - \varepsilon)^{\lfloor \frac{L}{2} - 1 \rfloor} \cdot 2^{-k}.
\end{aligned}$$ ◀

[2] Unlike the notation $\mathcal{P}_{C,L}$ defined in Section 4.1, we do not require $u_1 \cap \{T - n + 1, \dots, T\} \ne \varnothing$ here.

The remaining part of the section is devoted to a proof of Lemma 14. We apply induction on L. The base case is that $L = 1$ and $L = 2$, in which the lemma trivially holds. For larger L and every path $P = (u_1, u_2, u_3 \dots, u_L) \in \mathcal{P}_{(u_1),L}$, we discuss how the vertices u_1, u_2 and u_3 overlap.

Recall for every $u = e_{C,t} \in V_T$, $C(u) = C$ is its label. Similar to [10], we classify the tuple (u_1, u_2, u_3) into three categories.

- We say u_2 is *good* if $|C(u_2) \cap C(u_1)| \leq \alpha \cdot k$ where $\alpha \in [0, 1]$ is a parameter to be set;
- If u_2 is not good, we say (u_2, u_3) is of *type I* if $C(u_3) \cap C(u_1) \neq \varnothing$;
- If u_2 is not good, we say (u_2, u_3) is of *type II* if $C(u_3) \cap C(u_1) = \varnothing$.

Then we can write

$$\begin{aligned}
&\mathbf{Pr}\left[\exists \text{ open } P \in \mathcal{P}_{(u_1),L} \;\middle|\; u_1 \text{ is open}\right] \\
\leq\, &\mathbf{Pr}\left[\exists \text{ open } P = (u_1, u_2, u_3, \dots) \in \mathcal{P}_{(u_1),L} \text{ where } u_2 \text{ is good} \;\middle|\; u_1 \text{ is open}\right] \\
&+ \mathbf{Pr}\left[\exists \text{ open } P = (u_1, u_2, u_3, \dots) \in \mathcal{P}_{(u_1),L} \text{ where } (u_2, u_3) \text{ is of type I} \;\middle|\; u_1 \text{ is open}\right] \\
&+ \mathbf{Pr}\left[\exists \text{ open } P = (u_1, u_2, u_3, \dots) \in \mathcal{P}_{(u_1),L} \text{ where } (u_2, u_3) \text{ is of type II} \;\middle|\; u_1 \text{ is open}\right]
\end{aligned} \tag{4}$$

In the following, we bound the probabilities in the three cases respectively.

u_2 is good

In this case, we have $|C(u_2) \cap C(u_1)| \leq \alpha \cdot k$. Therefore, conditioned on u_1 being open, at least $(1 - \alpha) \cdot k$ variables in u_2 are free (i.e. independent of those variables related to u_1). The number of choices of $C(u_2)$ is at most $k \cdot d$ and the number of choices of u_2 with fixed $C = C(u_2)$ is at most k. Combining this with the induction hypothesis, we have

$$\begin{aligned}
&\mathbf{Pr}\left[\exists \text{ open } P = (u_1, u_2, u_3, \dots) \in \mathcal{P}_{(u_1),L} \text{ where } u_2 \text{ is good} \;\middle|\; u_1 \text{ is open}\right] \\
=\, &\mathbf{Pr}\left[\bigcup_{\text{good } u_2^*} \left[\exists \text{ open } P \in \mathcal{P}_{(u_1,u_2^*),L}\right] \;\middle|\; u_1 \text{ is open}\right] \\
\leq &\sum_{\text{good } u_2^*} \mathbf{Pr}\left[\exists \text{ open } P \in \mathcal{P}_{(u_1,u_2^*),L} \;\middle|\; u_1 \text{ is open}\right] \\
= &\sum_{\text{good } u_2^*} \mathbf{Pr}\left[u_2^* \text{ is open} \mid u_1 \text{ is open}\right] \cdot \mathbf{Pr}\left[\exists \text{ open } P \in \mathcal{P}_{(u_1,u_2^*),L} \;\middle|\; (u_1, u_2^*) \text{ is open}\right] \\
= &\sum_{\text{good } u_2^*} \mathbf{Pr}\left[u_2^* \text{ is open} \mid u_1 \text{ is open}\right] \cdot \mathbf{Pr}\left[\exists \text{ open } P' \in \mathcal{P}_{(u_2^*),L-1} \;\middle|\; u_2^* \text{ is open}\right] \\
\leq\, &k^2 d \cdot 2^{-(1-\alpha)k} \cdot (1 - \varepsilon)^{\lfloor \frac{L-3}{2} \rfloor}.
\end{aligned} \tag{5}$$

(u_2, u_3) is of type I

In this case, we know $C(u_3)$ is adjacent to $C(u_1)$ in Φ and therefore the number of the choices of $C(u_3)$ is at most kd. The choice of u_3 with fixed $C = C(u_3)$ is at most $2k$ by Lemma 4. On the other hand, since P is an induced path, $u_1 \cap u_3 = \varnothing$ and $\mathbf{Pr}\left[u_3 \text{ is open} \mid u_1 \text{ is open}\right] = 2^{-k}$. Combining these facts with the induction hypothesis, we have

$$
\begin{aligned}
&\mathbf{Pr}\left[\exists \text{ open } P=(u_1,u_2,u_3,\dots)\in\mathcal{P}_{(u_1),L} \text{ where } (u_2,u_3) \text{ is of type I} \;\middle|\; u_1 \text{ is open}\right]\\
\le\;&\mathbf{Pr}\left[\bigcup_{u_3^*\,:\,\exists u_2,\ (u_2,u_3^*) \text{ type I}} \left[\exists \text{ open } P'\in\mathcal{P}_{(u_3^*),L-2}\right] \;\middle|\; u_1 \text{ is open}\right]\\
\le\;&\sum_{u_3^*\,:\,\exists u_2,\ (u_2,u_3^*) \text{ type I}} \mathbf{Pr}\left[\exists \text{ open } P'\in\mathcal{P}_{(u_3^*),L-2} \;\middle|\; u_1 \text{ is open}\right]\\
=\;&\sum_{u_3^*\,:\,\exists u_2,\ (u_2,u_3^*) \text{ type I}} \mathbf{Pr}\left[u_3^* \text{ is open} \;\middle|\; u_1 \text{ is open}\right]\cdot\mathbf{Pr}\left[\exists \text{ open } P'\in\mathcal{P}_{(u_3^*),L-2} \;\middle|\; (u_1,u_3^*) \text{ is open}\right]\\
=\;&\sum_{u_3^*\,:\,\exists u_2,\ (u_2,u_3^*) \text{ type I}} \mathbf{Pr}\left[u_3^* \text{ is open} \;\middle|\; u_1 \text{ is open}\right]\cdot\mathbf{Pr}\left[\exists \text{ open } P'\in\mathcal{P}_{(u_3^*),L-2} \;\middle|\; u_3^* \text{ is open}\right]\\
\le\;&2k^2d\cdot 2^{-k}\cdot(1-\varepsilon)^{\lfloor\frac{L-4}{2}\rfloor}.
\end{aligned}
\tag{6}
$$

(u_2, u_3) is of type II

This case is more complicated. We will enumerate all type II pairs (u_2, u_3) and bound the sum of probabilities that some path in $\mathcal{P}_{(u_2,u_3),L-1}$ is open conditioned on u_1 being open. To this end, we first enumerate all pairs $(C_2, C_3) \in \mathcal{C}^2$ and then consider all pairs (u_2, u_3) with $C(u_2) = C_2$ and $C(u_3) = C_3$.

For a fixed pair (C_2, C_3), we aim to bound the probability

$$
\sum_{\substack{(u_2,u_3) \text{ is of type II,}\\ C(u_2)=C_2, C(u_3)=C_3}} \mathbf{Pr}\left[\exists \text{ open } P\in\mathcal{P}_{(u_1,u_2,u_3),L} \;\middle|\; u_1 \text{ is open}\right]. \tag{7}
$$

To ease the notation, we let $C_1 := C(u_1)$. In order for the above sum to be nonzero, we must have $|C_2 \cap C_1| > \alpha \cdot k$ and $C_3 \cap C_1 = \varnothing$. Let $a := |C_2 \cap C_1|$ and $b := |C_3 \cap C_2|$. Recall the notations in Section 3.1. For those (u_2, u_3) with $C(u_2) = C_2$ and $C(u_3) = C_3$, we can write them as $u_2 = e_{C_2,t_2}$ and $u_3 = e_{C_3,t_3}$ respectively. In the following, all pairs (u_2, u_3) discussed are of type II.

Note that $e_{C_2,t_2} =: u_2 \neq u_2' := e_{C_2,t_2'}$ if and only if $t_2 \neq t_2'$. Similarly $e_{C_2,t_2} =: u_3' \neq u_3' := e_{C_2,t_2'}$ if and only if $t_3 \neq t_3'$. Moreover, if $u_2 \neq u_2'$, then $|u_2 \cap u_1| \neq |u_2' \cap u_1|$ and similarly if $u_3 \neq u_3'$, then $|u_3 \cap u_2| \neq |u_3' \cap u_2|$. As a result, we can enumerate type II pairs (u_2, u_3) by enumerating the sizes of $u_2 \cap u_1$ and $u_3 \cap u_2$ respectively. On the other hand, if we know that $|u_2 \cap u_1| = i$ and $|u_3 \cap u_2| = j$, then,

$$
\begin{aligned}
&\mathbf{Pr}\left[\exists \text{ open } P\in\mathcal{P}_{(u_1,u_2,u_3),L} \;\middle|\; u_1 \text{ is open}\right]\\
=\;&\mathbf{Pr}\left[u_2 \text{ is open} \mid u_1 \text{ is open}\right]\cdot\mathbf{Pr}\left[u_3 \text{ is open} \mid (u_1,u_2) \text{ is open}\right]\\
&\cdot\mathbf{Pr}\left[\exists \text{ open } P\in\mathcal{P}_{(u_1,u_2,u_3),L} \;\middle|\; (u_1,u_2,u_3) \text{ is open}\right]\\
\le\;&\mathbf{Pr}\left[u_2 \text{ is open} \mid u_1 \text{ is open}\right]\cdot\mathbf{Pr}\left[u_3 \text{ is open} \mid u_2 \text{ is open}\right]\\
&\cdot\mathbf{Pr}\left[\exists \text{ open } P'\in\mathcal{P}_{(u_3),L-2} \;\middle|\; u_3 \text{ is open}\right]\\
=\;&2^{-(k-i)}\cdot 2^{-(k-j)}\cdot\mathbf{Pr}\left[\exists \text{ open } P'\in\mathcal{P}_{(u_3),L-2} \;\middle|\; u_3 \text{ is open}\right].
\end{aligned}
$$

These facts together with the induction hypothesis give

$$\sum_{\substack{(u_2,u_3) \text{ is of type II:}\\ C(u_2)=C_2, C(u_3)=C_3}} \mathbf{Pr}\left[\exists \text{ open } P \in \mathcal{P}_{(u_1,u_2,u_3),L} \;\middle|\; u_1 \text{ is open}\right]$$
$$\le \sum_{i=1}^{a}\sum_{j=1}^{b} 2^{-(k-i)} \cdot 2^{-(k-j)} \cdot (1-\varepsilon)^{\lfloor \frac{L-4}{2} \rfloor}$$
$$\le 2^{a+b-2k} \cdot 4(1-\varepsilon)^{\lfloor \frac{L-4}{2} \rfloor}.$$

It remains to enumerate (C_2, C_3) pairs. Since we know that $|C_2 \cap C_1| > \alpha \cdot k$, the number of choices of C_2 is at most $\frac{d\cdot k}{\alpha \cdot k} = \frac{d}{\alpha}$[3]. For every fixed C_2, we let $\mathcal{C} = \left\{C_3^{(1)}, C_3^{(2)}, \ldots, C_3^{(m)}\right\}$ be the collection of all possible C_3. Denote $a := |C_2 \cap C_1|$ as before and for every $i \in [m]$ let $b_i := \left|C_3^{(i)} \cap C_2\right|$. The probability of interest can therefore be written as

$$\mathbf{Pr}\left[\exists \text{ open } (u_1,u_2,u_3,\ldots) \in \mathcal{P}_{(u_1),L} \text{ where } (u_2,u_3) \text{ is of type II and } C(u_2) = C_2 \;\middle|\; u_1 \text{ is open}\right]$$
$$\le \sum_{\substack{(u_1,u_2,u_3,\ldots)\in\mathcal{P}_{(u_1),L}:\\ (u_2,u_3) \text{ is of type II,}\\ C(u_2)=C_2}} \mathbf{Pr}\left[\exists \text{ open } P \in \mathcal{P}_{(u_1,u_2,u_3),L} \;\middle|\; u_1 \text{ is open}\right]$$
$$\le \left(\sum_{i=1}^{m} 2^{a+b_i-2k}\right) \cdot 4(1-\varepsilon)^{\lfloor \frac{L-4}{2} \rfloor}.$$

To bound the term $\sum_{i=1}^m 2^{a+b_i-2k}$, let us list some properties of the numbers involved:

- $a \le k$ and $1 \le m \le (k-a)d$;
- $1 \le b_i \le k-a$ for all $i \in [m]$;
- $\sum_{i=1}^m b_i \le (k-a)d$. (This can be seen by an argument similar to the last footnote and the fact that $C_3 \cap C_1 = \varnothing$.)

Roughly the numbers are acting against each other: If a and m are large then the b_i's are small. So it is possible to control $\sum_{i=1}^m 2^{a+b_i-2k}$ reasonably:

▶ **Lemma 15.** *Suppose $a \le k$ and $1 \le m \le (k-a)d$. Then for any integers $b_1, \ldots, b_m \in [k-a]$ such that $\sum_{i=1}^m b_i \le (k-a)d$, we have the inequality $\sum_{i=1}^m 2^{a+b_i-2k} \le d \cdot 2^{-k}$.*

We arrange the proof in Appendix A. Returning to previous discussion, we derive

$$\mathbf{Pr}\left[\exists \text{ open } P = (u_1,u_2,u_3,\ldots) \in \mathcal{P}_{(u_1),L} \text{ where } (u_2,u_3) \text{ is of type II} \;\middle|\; u_1 \text{ is open}\right]$$
$$\le \frac{4d^2}{\alpha} \cdot 2^{-k} \cdot (1-\varepsilon)^{\lfloor \frac{L-4}{2} \rfloor}. \tag{8}$$

Putting all together

Plugging Equations (5), (6) and (8) into Equation (4), we have

$$\mathbf{Pr}\left[\exists \text{ open } P \in \mathcal{P}_{(u_1),L} \;\middle|\; u_1 \text{ is open}\right]$$
$$\le \left(k^2 d \cdot 2^{-(1-\alpha)k} + 2k^2 d \cdot 2^{-k} + \frac{4d^2}{\alpha} \cdot 2^{-k}\right) \cdot (1-\varepsilon)^{\lfloor \frac{L-4}{2} \rfloor}$$
$$< \left(2^{\alpha k+2} k^2 d + \frac{4}{\alpha} d^2\right) 2^{-k} \cdot (1-\varepsilon)^{\lfloor \frac{L-4}{2} \rfloor}$$

[3] To see this, consider the following way to enumerate all C_2 incident to C_1 in Φ: First pick a vertex in C_1 and then pick one of its incident hyperedges. This way we enumerated (with repetitions) $k \cdot d$ hyperedges in total, and every hyperedge $C_2 : |C_2 \cap C_1| > \alpha \cdot k$ is enumerated at least $\alpha \cdot k$ times.

where we used a very crude inequality $2^{\alpha k}+2 < 2^{\alpha k+2}$. Let us take $\alpha = \alpha(k) := \frac{1}{2}+\frac{3-2\log k}{k} \in [0,1]$ for any $k \geq 2$. Note that $2^{\alpha k+2}k^2 = 2^{5+k/2}$ by definition. On the other hand, from basic calculus one may find that $\alpha(k) : k \in \mathbb{N}$ attains minimum value $1/8$ at $k = 8$, thus $4/\alpha \leq 32 = 2^5$. Hence the above bound continues as:

$$\begin{aligned}\cdots &\leq \left(2^{k/2}d + d^2\right) 2^{5-k} \cdot (1-\varepsilon)^{\lfloor \frac{L-4}{2} \rfloor} \\ &\leq (1-\varepsilon) \cdot (1-\varepsilon)^{\lfloor \frac{L-4}{2} \rfloor} \\ &\leq (1-\varepsilon)^{\lfloor \frac{L-2}{2} \rfloor}\end{aligned}$$

where the second line is due to our assumption $d \leq \left(\frac{1}{4}\sqrt{\frac{9-\varepsilon}{2}} - \frac{1}{2}\right) \cdot 2^{k/2} =: d^*$. To see this, observe that the function $g(d) := 2^{k/2}d + d^2$ is monotonically increasing when $d \geq 0$, and that $g(d^*) = (1-\varepsilon) \cdot 2^{k-5}$.

▶ Remark 16. When $\varepsilon := 0.1$, say, the condition becomes (approximately) $d \leq 0.027 \cdot 2^{k/2}$. Taking smaller ε will allow larger applicable range of d, but it comes at the price of slowing down the sampler (by a multiplicative log-factor; see Theorem 8).

If k is sufficiently large we can further improve our bound to $d \lesssim \frac{\sqrt{33-\varepsilon}-1}{16} \cdot 2^{k/2}$.

References

1 Ivona Bezáková, Andreas Galanis, Leslie Ann Goldberg, Heng Guo, and Daniel Stefankovic. Approximation via correlation decay when strong spatial mixing fails. *SIAM Journal on Computing*, 48(2):279–349, 2019.

2 Weiming Feng, Heng Guo, and Jiaheng Wang. Improved bounds for randomly colouring simple hypergraphs. *arXiv preprint*, 2022. `arXiv:2202.05554`.

3 Weiming Feng, Heng Guo, Yitong Yin, and Chihao Zhang. Fast sampling and counting k-sat solutions in the local lemma regime. *Journal of the ACM*, 68(6):40:1–40:42, 2021.

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

5 Andreas Galanis, Leslie Ann Goldberg, Heng Guo, and Kuan Yang. Counting solutions to random cnf formulas. *SIAM Journal on Computing*, 50(6):1701–1738, 2021.

6 Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the Lovász local lemma. *Journal of the ACM*, 66(3):1–31, 2019.

7 Heng Guo, Chao Liao, Pinyan Lu, and Chihao Zhang. Counting hypergraph colorings in the local lemma regime. *SIAM Journal on Computing*, 48(4):1397–1424, 2019.

8 Kun He, Xiaoming Sun, and Kewen Wu. Perfect sampling for (atomic) Lovász local lemma. *arXiv preprint*, 2021. `arXiv:2107.03932`.

9 Kun He, Chunyang Wang, and Yitong Yin. Sampling Lovász local lemma for general constraint satisfaction solutions in near-linear time. *arXiv preprint*, 2022. `arXiv:2204.01520`.

10 Jonathan Hermon, Allan Sly, and Yumeng Zhang. Rapid mixing of hypergraph independent sets. *Random Structures & Algorithms*, 54(4):730–767, 2019.

11 Vishesh Jain, Huy Tuan Pham, and Thuy-Duong Vuong. On the sampling Lovász local lemma for atomic constraint satisfaction problems. *arXiv preprint*, 2021. `arXiv:2102.08342`.

12 Vishesh Jain, Huy Tuan Pham, and Thuy Duong Vuong. Towards the sampling Lovász local lemma. In *Proceedings of the 62nd IEEE Annual Symposium on Foundations of Computer Science*, 2021.

13 Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

14 Eyal Lubetzky and Allan Sly. Information percolation and cutoff for the stochastic ising model. *Journal of the American Mathematical Society*, 29(3):729–774, 2016.

15 Ankur Moitra. Approximate counting, the Lovász local lemma, and inference in graphical models. *Journal of the ACM*, 66(2):10:1–10:25, 2019.
16 Robin A Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. *Journal of the ACM*, 57(2):1–15, 2010.
17 James Gary Propp and David Bruce Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures & Algorithms*, 9(1-2):223–252, 1996.

A Proof of Lemma 15

For each $j \in [k-a]$ we introduce a counter $x_j := |\{i : b_i = j\}|$ that counts the number of b_i's taking a specific value j. Then the constraint of the lemma translates to

$$\sum_{j=1}^{k-a} j \cdot x_j \leq (k-a)d,$$

and we want to show

$$\sum_{j=1}^{k-a} 2^{a+j-2k} \cdot x_j \leq d \cdot 2^{-k}.$$

To this end, we consider a (relaxed) linear program with variables $x_1, \ldots, x_{k-a}$:

$$\begin{aligned} \max \quad & \sum_{j=1}^{k-a} 2^{a+j-2k} \cdot x_j \\ \text{s.t} \quad & \sum_{j=1}^{k-a} j \cdot x_j \leq (k-a)d \\ & x_j \geq 0 \quad (\forall j \in [k-a]). \end{aligned}$$

By the *strong duality theorem of linear programming*, its maximum value equals the minimum value of the dual program

$$\begin{aligned} \min \quad & (k-a)d \cdot y \\ \text{s.t} \quad & j \cdot y \geq 2^{a+j-2k} \quad (\forall j \in [k-a]) \\ & y \geq 0. \end{aligned}$$

Clearly the minimum value is obtained when y is as small as possible. It is clear that the sequence $h_j := \frac{2^j}{j}$ is monotonically increasing for integers $j \geq 1$, so the minimum possible y is given by the $(k-a)$-th constraint, namely

$$y^* := \frac{2^{a+(k-a)-2k}}{k-a} = \frac{2^{-k}}{k-a}.$$

Therefore, the minimum value of the dual – also the maximum value of the primal – is exactly $d \cdot 2^{-k}$, as desired.