Improved Bounds for Randomly Colouring Simple Hypergraphs

Weiming Feng
School of Informatics, University of Edinburgh, UK

Heng Guo
School of Informatics, University of Edinburgh, UK

Jiaheng Wang
School of Informatics, University of Edinburgh, UK

Abstract

We study the problem of sampling almost uniform proper q-colourings in k-uniform simple hypergraphs with maximum degree Δ. For any $\delta > 0$, if $k \geq \frac{20(1+\delta)}{\delta}$ and $q \geq 100\Delta^{\frac{2+\delta}{k-4/\delta-4}}$, the running time of our algorithm is $\tilde{O}(\mathrm{poly}(\Delta k) \cdot n^{1.01})$, where n is the number of vertices. Our result requires fewer colours than previous results for general hypergraphs (Jain, Pham, and Vuong, 2021; He, Sun, and Wu, 2021), and does not require $\Omega(\log n)$ colours unlike the work of Frieze and Anastos (2017).

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

Keywords and phrases Approximate counting, Markov chain, Mixing time, Hypergraph colouring

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2022.25

Category RANDOM

Related Version *Full Version*: `https://arxiv.org/abs/2202.05554` [7]

Funding This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 947778).

1 Introduction

The past few years have witnessed a bloom in techniques targeted at approximate counting and sampling problems, among which constraint satisfaction problems (CSPs) are probably the most studied. In fact, many problems can be cast as CSPs, e.g., Boolean satisfiability problems (SATs), proper colourings of graphs and hypergraphs, and independent sets, to name a few. In general, even deciding if a CSP instance can be satisfied or not is **NP**-hard. However, efficient algorithms become possible when the number of appearances of each variable (usually referred to as the degree) is not too high. For these instances, the Lovász Local Lemma [6] provides a fundamental criterion to guarantee the existence of a solution. Although the original local lemma does not provide an efficient algorithm, after two decades of effort [2, 1, 28, 5, 32, 29], the celebrated work of Moser and Tardos [30] provides an efficient algorithm matching the same conditions as the local lemma.

Unfortunately, the output distribution of the Moser–Tardos algorithm does not suit the need of approximate counting and sampling. This deficiency is fundamental, as it can be **NP**-hard to (uniformly or near-uniformly) sample satisfying assignments even when the criterion of the local lemma is satisfied and the corresponding searching problem lies in **P** [3, 14].[1] In other words, sampling problems are fundamentally more difficult than searching problems in the local lemma regime. Part of the difficulty comes from the possibility that

[1] As far as we are aware, all hardness results for sampling (including the ones mentioned in this paper) allow errors in total variation distances. It is not clear if stronger hardness results exist for perfect sampling (i.e. with no error).

the state space can be disconnected from local moves, but traditional algorithmic tools like Markov chain Monte Carlo rely on the connectivity. This barrier has been bypassed recently by some exciting developments [27, 16, 17, 23], and in particular the projected Markov chain approach [8, 9, 24, 19]. For searching problems, the local lemma is known to give a sharp computational transition threshold from **P** to **NP**-hard [30, 15] as the degree increases. Recent efforts aim to find and establish a similar threshold for sampling problems as well.

One very promising problem to establish such a threshold is (proper) q-colourings of hypergraphs, which is the original setting where the local lemma was developed [6], and has received considerable recent attention. A hypergraph $H = (V, \mathcal{E})$ consists of a set of vertices V and a set of hyperedges $\mathcal{E} \subseteq 2^V$. We say $H = (V, \mathcal{E})$ is k-uniform, if every hyperedge $e \in \mathcal{E}$ satisfies $|e| = k$. A colouring of a hypergraph is *proper* if no hyperedge is monochromatic. An efficient (perfect) sampler exists when $q \gtrsim \Delta^{3/(k-4)}$ (where $\gtrsim$ or $\lesssim$ hides some constant independent from q, k, and Δ) for k-uniform hypergraphs with maximum degree Δ [24, 19], while the sampling problem is **NP**-hard whenever $q \lesssim \Delta^{2/k}$ for even q [14]. For comparison, the local lemma shows that a proper q-colouring exists if $q \gtrsim \Delta^{1/(k-1)}$ (see also [35] for a recent alternative approach leading to a slightly better constant).

On the other hand, before the recent wave of local lemma inspired sampling algorithms, randomly sampling q-colourings in *simple* k-uniform hypergraphs[2] has already been studied [12, 11]. In particular, Frieze and Anastos [11] gave an efficient sampling algorithm when the number of colours satisfies $q \geq \max\{C_k \log n, 500k^3\Delta^{\frac{1}{k-1}}\}$, where n is the number of vertices and C_k depends only on k. Their algorithm is the standard Glauber dynamics with a random initial (not necessarily proper) colouring. The logarithmic lower bound on the number of colours is crucial to their analysis, as it guarantees that there is a giant connected component in the state space so that connectivity is not an issue.

In this paper, we study the projected Markov chain for sampling q-colourings in simple hypergraphs. Our result improves the bound of [24, 19] for general hypergraphs, and does not require unbounded number of colours, unlike in [12, 11]. Let μ denote the uniform distribution over all proper colourings. Our main result is stated as follows.

▶ **Theorem 1.** *For any $\delta > 0$, there is a sampling algorithm such that given any $\epsilon \in (0, 1)$, a k-uniform simple hypergraph $H = (V, E)$ with maximum degree Δ, where $k \geq \frac{20(1+\delta)}{\delta}$, and an integer $q \geq 100\Delta^{\frac{2+\delta}{k-4/\delta-4}}$, it returns a random q-colouring that is ϵ-close to μ in total variation distance in time $\tilde{O}(k^5\Delta^2 n \left(\frac{n\Delta}{\epsilon}\right)^{0.01})$, where $n = |V|$ and $\tilde{O}$ hides a $\mathrm{polylog}(n, \Delta, q, 1/\epsilon)$ factor.*

A few quick remarks are in order. First of all, the exponent of n in the running time can be made even closer to 1 if more colours are given. See Theorem 10 for the full technical statement. Secondly, our algorithm can be modified into a perfect sampler by applying the bounding chain method [21] based on coupling from the past (CFTP) [31], following the same lines of [19]. Moreover, using known reductions from approximate counting to sampling [25, 34, 22, 26] (see [8] for simpler arguments specialized to local lemma settings), one can efficiently and approximately count the number of proper colourings in simple hypergraphs under the same conditions in Theorem 1.

Our algorithm follows the recent projected Markov chain approach [8] with state compression [9]. Roughly speaking, instead of assigning colours to vertices, we split $[q]$ into $\sqrt{q}$ buckets of size $\sqrt{q}$ each and assign buckets to vertices. We run a (systematic scan) Markov chain on these bucket assignments to generate a sample, and then conditional on this sample to draw a nearly uniform q-colouring. The benefit of this bucketing is that, under the conditions of

[2] A hypergraph is *simple* if any two hyperedges intersect in at most one vertex. Simple hypergraphs are also known as linear hypergraphs.

Theorem 1, conditional on the assignments of all but one vertices, the assignment of the remaining vertex is close to uniformly at random. This implies that any atomic event[3] is exponentially unlikely in the number of distinct vertices it depends on. In order to show that this approach works, we need to show two things: 1) the projected Markov chain is rapidly mixing; 2) each step of the Markov chain can be efficiently implemented. For general hypergraphs, the previous $q \gtrsim \Delta^{3/(k-4)}$ bound comes from balancing the conditions so that the two claims are true simultaneously. However, there is no room left for relaxation on either claim. This means that, for our improvements in simple hypergraphs, new ingredients are required for both claims.

For rapid mixing, we take the information percolation approach [20, 24, 19], where the main effort is to trace discrepancies through a one-step greedy coupling, and to show that they are unlikely after a sufficient amount of time. In simple hypergraphs, an individual discrepancy path through time has more distinct updates of vertices than in the general case, and are thus more unlikely. This allows us to relax the condition. Our mixing time analysis is largely inspired by the work of Hermon, Sly, and Zhang [20], although we do need to handle some new complicacies, such as hyperedges whose vertices are consecutively updated in the discrepancy path.

For efficient implementation, we use rejection sampling. Here we want to sample the colour/bucket of a vertex conditional on the buckets of all other vertices. We can safely prune hyperedges containing vertices of different buckets. The remaining connected component containing the update vertex needs to have logarithmic size to guarantee efficiency of our rejection sampling. The standard approach to bound its size is to do a union bound over certain combinatorial structures with sufficiently many distinct vertices. Most previous analysis is based on enumerating so-called "2-trees", a notion first introduced by Alon [1]. Unfortunately, under the conditions of Theorem 1, there are too many "2-trees" to our need. Instead, we introduce a new structure called "2-block-trees" (see Definition 15). Here each "block" is a collection of θ connected hyperedges, and these blocks satisfy connectivity properties similar to a 2-tree. Since the hypergraph is simple, a block has at least $\theta k - \binom{\theta}{2}$ distinct vertices. As long as $\theta \ll k$, we have a good lower bound on the number of distinct vertices, which in turn implies a good upper bound on the probability of these structures showing up. To finish off with the union bound, we give a new counting argument for the number of 2-block-trees, which is based on finding a good encoding of these structures.

The exponent (roughly $2/k$) of Δ in Theorem 1 is unlikely to be tight, although it appears to be the limit of current techniques. In fact, we conjecture that the computational transition for sampling q-colourings in simple hypergraphs happens around the same threshold of the local lemma (namely, the exponent should be roughly $1/k$). This conjecture is supported by the hardness result of Galanis, Guo, and Wang [14] for general q, and by the algorithm of Frieze and Anastos [11] for $q = \Omega(\log n)$. Note that for a simple k-uniform hypergraph with maximum degree Δ, Frieze and Mubayi [10] showed that the chromatic number $\chi(H) \leq C_k \left(\frac{\Delta}{\log \Delta}\right)^{\frac{1}{k-1}}$ where C_k depends only on k. Their bound is asymptotically better than the bound given by the local lemma. Thus there may still be a gap between the searching threshold and the sampling threshold.

A final remark is that our method would still work as long as the overlap of hyperedges is much smaller than k. The condition on the parameters will deteriorate slightly but would still be better than those for general hypergraphs. In light of this, our method can potentially

[3] An event is *atomic* if each variable it depends on must take one particular value. In discrete spaces, any event can be decomposed into atomic ones.

be applied to improve the constant of [13] on sampling solutions to random CNF formulas, where the overlaps are at most 2 with high probability. On the other end of the spectrum, if any two intersecting hyperedges intersect at at least $k/2$ vertices, the algorithm by Guo, Jerrum, and Liu [16] almost matches the hardness result [14]. It is an intriguing question how the size of overlaps affects the complexity of these sampling problems, or whether it is possible to improve sampling algorithms via a better use of the overlap information.

Due to space limitation, we omit many proofs and only give sketches for a few important results. Complete proofs can be found in the full version.

2 Preliminaries

In this section we gather some preliminary definitions and results for later use. We generally use the bold font to denote vectors, matrices, and/or random variables. The notation $[q]$ stands for $\{1, 2, \cdots, q\}$.

2.1 Graph theory

Throughout this paper, we use the following notations for a graph $G = (V, E)$:

- $G[A]$: the induced subgraph of G on the vertex subset $A \subseteq V$.
- $\mathrm{dist}_G(A, B)$: the distance between two vertex sets $A \subseteq V$ and $B \subseteq V$ on G, which is defined by $\mathrm{dist}_G(A, B) := \min_{u\in A, v\in B} \mathrm{dist}_G(u, v)$ and $\mathrm{dist}_G(u, v)$ is the length of the shortest path between u and v in G.
- $\Gamma^i_G(A)$: the set of vertices u such that $\mathrm{dist}_G(A, u) = i$. Specifically, when $i = 1$, this notation represents the neighbourhood of the given set $A \subseteq V$, and is also denoted by $\Gamma_G(A)$.

We sometimes do not distinguish u and the singleton set $\{u\}$ in sub- or sup-scripts. For the sake of convenience, we may drop the subscript G when the underlying graph is clear from the context.

We need some more definitions for later use.

▶ **Definition 2** (Graph power). *Let G be an undirected graph. The i-th power of G, denoted by G^i, is another graph that has the same vertex set as G, and $\{u, v\}$ is an edge in G^i iff $1 \le \mathrm{dist}_G(u, v) \le i$.*

▶ **Definition 3** (Line graph). *Let $H = (V, \mathcal{E})$ be a hypergraph. Its line graph $\mathrm{Lin}(H) = (V_L, E_L)$ is given by $V_L = \mathcal{E}$, and $\{e, e'\} \in E_L$ iff $e \cap e' \neq \emptyset$.*

2.2 Coupling and Markov chains

Consider a discrete state space Ω and two distributions μ and ν over it. The *total variation distance* between μ and ν is defined by

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

A *coupling* between μ and ν is a joint distribution $(X, Y) \in \Omega^2$ such that its marginal distribution over X (resp. Y) is μ (resp. ν). The next lemma, usually referred to as the *coupling lemma*, bounds the total variation distance between μ and ν by any of their couplings.

▶ **Lemma 4** (Coupling lemma). *For any coupling (X, Y) between between μ and ν, $d_{\mathrm{TV}}(\mu, \nu) \le \Pr[X \neq Y]$. Moreover, there exists an optimal coupling reaching the equality.*

Given a finite state space Ω, a discrete-time *Markov chain* is a sequence $\{X_t\}_{t\geq 0}$ where the probability of each possible state of X_{t+1} only depends on the state of X_t. The transition of the chain is represented by the *transition matrix* $\boldsymbol{P} : \Omega^2 \to \mathbb{R}_{[0,1]}$, where $\boldsymbol{P}(i,j) = \Pr[X_{t+1} = j \mid X_t = i]$. When the state space Ω is clear from context, we simply denote the chain by its transition matrix. A Markov chain $\boldsymbol{P}$ is:

- *irreducible*, if for any $X, Y \in \Omega$, there exists $t > 0$ such that $\boldsymbol{P}^t(X,Y) > 0$;
- *aperiodic*, if for all $X \in \Omega$, it holds that $\gcd\{t \mid \boldsymbol{P}^t(X,X) > 0\} = 1$; and
- *reversible* with respect to a distribution π, if $\forall X, Y \in \Omega$, $\pi(X)\boldsymbol{P}(X,Y) = \pi(Y)\boldsymbol{P}(Y,X)$. This equation is usually known as the *detailed balance condition*.

A distribution π is *stationary* for $\boldsymbol{P}$, if $\pi\boldsymbol{P} = \pi$ (regarding π as a row vector). The detailed balance condition actually implies that the corresponding distribution is stationary. Furthermore, if a Markov chain is both irreducible and aperiodic, then it converges to a unique stationary distribution π. The speed of convergence towards π is characterised by its *mixing time*, defined by

$$t_{\mathrm{mix}}(\boldsymbol{P}, \epsilon) := \min\left\{t \mid \max_{X\in\Omega} d_{\mathrm{TV}}(\boldsymbol{P}^t(X,\cdot), \pi) < \epsilon\right\}.$$

The joint process $(X_t, Y_t)_{t\geq 0}$ is a *coupling of Markov chain* $\boldsymbol{P}$ if $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$ individually follow the transition rule of $\boldsymbol{P}$, and if $X_i = Y_i$ then $X_j = Y_j$ for all $j \geq i$. By the coupling lemma, for any coupling $(X_t, Y_t)_{t\geq 0}$ of $\boldsymbol{P}$, it holds that $d_{\mathrm{TV}}(P^t(X_0,\cdot), P^t(Y_0,\cdot)) \leq \Pr[X_t \neq Y_t]$. Hence, the mixing time of $\boldsymbol{P}$ can be bounded by

$$t_{\mathrm{mix}}(\boldsymbol{P}, \epsilon) \leq \max_{X_0, Y_0 \in \Omega} \min\left\{t \mid \Pr[X_t \neq Y_t] \leq \epsilon\right\}. \tag{1}$$

2.3 Lovász Local Lemma

Let $\mathcal{R} = \{R_1, \cdots, R_n\}$ be a set of mutually independent random variables. Given an event A, denote the set of variables that determines A by $\mathsf{vbl}(A) \subseteq \mathcal{R}$. Let $\mathcal{B} = \{B_1, \cdots, B_n\}$ be a collection of "bad" events. For any event A (not necessarily in $\mathcal{B}$), let $\Gamma(A) := \{B \in \mathcal{B} \mid B \neq A,\ \mathsf{vbl}(B) \cap \mathsf{vbl}(A) \neq \emptyset\}$. We will use the following version of *Lovász Local Lemma* from [18].

▶ **Theorem 5** ([6, 18]). *If there exists a function $x : \mathcal{B} \to (0,1)$ such that for any bad event $B \in \mathcal{B}$,*

$$\Pr[B] \leq x(B) \prod_{B' \in \Gamma(B)} (1 - x(B')), \tag{2}$$

then it holds that

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

Moreover, for any event A,

$$\Pr\left[A \mid \bigwedge_{B\in\mathcal{B}} \bar{B}\right] \leq \Pr[A] \prod_{B\in\Gamma(A)} (1 - x(B))^{-1}. \tag{3}$$

2.4 List hypergraph colouring and local uniformity

In our algorithm and analysis, we consider the general *list hypergraph colouring* problem. Let $H = (V, \mathcal{E})$ be a k-uniform hypergraph with maximum degree Δ. Let $(Q_v)_{v\in V}$ be a set of colour lists. We say $\boldsymbol{X} \in \otimes_{v\in V} Q_v$ is a proper list colouring if no hyperedge in H is monochromatic with respect to $\boldsymbol{X}$. Let μ denote the uniform distribution of all proper list hypergraph colourings. The following local uniformity property holds for the distribution μ. Its proof follows from the argument in [17].

▶ **Lemma 6** (local uniformity [17]). *Let $q_0 = \min_{v\in V} |Q_v|$ and $q_1 = \max_{v\in V} |Q_v|$. For any $r \geq k \geq 2$, if $q_0^k \geq \mathrm{e} q_1 r\Delta$, then for any $v \in V$ and $c \in Q_v$,*

$$\frac{1}{|Q_v|}\exp\left(-\frac{2}{r}\right) \leq \mu_v(c) \leq \frac{1}{|Q_v|}\exp\left(\frac{2}{r}\right),$$

where μ_v is the marginal distribution on v induced by μ.

3 Algorithm

Let $H = (V, \mathcal{E})$ be a k-uniform hypergraph and $[q]$ a set of colours. Let μ denote the uniform distribution of proper hypergraph colourings. Our algorithm is a variant of the projected dynamics from [8], using a particular projection scheme from [9]. We first introduce some basic definitions and notations, and then describe the sampling algorithm.

3.1 Projection scheme, projected distribution and conditional distribution

Our sampling algorithm is based on the following *projection scheme* introduced in [9].

▶ **Definition 7** (projection scheme [9]). *Let $1 \leq s \leq q$ be an integer. A (balanced) projection scheme with image size s is a function $h : [q] \to [s]$ such that for any $j \in [s]$, $\left|h^{-1}(j)\right| = \lfloor \frac{q}{s} \rfloor$ or $\left|h^{-1}(j)\right| = \lceil \frac{q}{s} \rceil$.*

For any $\boldsymbol{X} \in [q]^V$, define the projection *image* $\boldsymbol{Y} \in [s]^V$ of $\boldsymbol{X}$ by

$$\forall v \in V, \quad Y_v = h(X_v).$$

For simplicity, we often denote $\boldsymbol{Y} = h(\boldsymbol{X})$, and for any subset $\Lambda \subseteq V$, we denote $\boldsymbol{Y}_\Lambda = h(\boldsymbol{X}_\Lambda)$.

Given a projection scheme, the following *projected distribution* can be naturally defined.

▶ **Definition 8** (projected distribution). *Given a projection scheme h, the projected distribution ν is the distribution of $\boldsymbol{Y} = h(\boldsymbol{X})$, where $\boldsymbol{X} \sim \mu$.*

Given an image of the projection, we can define the following *conditional distribution* over $[q]^V$.

▶ **Definition 9** (conditional distribution). *Let $\Lambda \subseteq V$ be a subset of vertices. Given a (partial) image $\sigma_\Lambda \in [s]^\Lambda$, the conditional distribution μ^{σ_Λ} is the distribution of $\boldsymbol{X} \sim \mu$ conditional on $h(\boldsymbol{X}_\Lambda) = \sigma_\Lambda$.*

By definition, μ^{σ_Λ} is a distribution over $[q]^V$. We use $\mu_S^{\sigma_\Lambda}$ to denote the marginal distribution on $S \subseteq V$ projected from μ^{σ_Λ}, and we simply denote $\mu_{\{v\}}^{\sigma_\Lambda}$ by $\mu_v^{\sigma_\Lambda}$.

Algorithm 1 Sampling algorithm for hypergraph colouring.

Input: A hypergraph $H = (V, \mathcal{E})$, a set of colours $[q]$, an error bound $0 < \epsilon < 1$, and a balanced projection scheme $h : [q] \to [s]$, where $s = \lceil \sqrt{q} \rceil$
Output: A random colouring $\boldsymbol{X} \in [q]^V$
1 sample $\boldsymbol{Y} \in [s]^V$ uniformly at random;
2 **for** t *from* 1 *to* $T = \lceil 50n \log \frac{2n\Delta}{\epsilon} \rceil$ **do**
3 let v be the vertex with label $(t \mod n)$;
4 $X'_v \leftarrow \mathsf{Sample}\left(H, h, \{v\}, \boldsymbol{Y}_{V\setminus\{v\}}, \frac{\epsilon}{4T}\right)$;
`/* The Sample subroutine is given in Algorithm 2. */`
5 $Y_v \leftarrow h(X'_v)$;
6 **return** $\boldsymbol{X} \leftarrow \mathsf{Sample}\left(H, h, V, \boldsymbol{Y}, \frac{\epsilon}{4T}\right)$;

3.2 The sampling algorithm

In this section and what follows, we always assume that all vertices in V are labeled by $\{0, 1, \ldots, n-1\}$. We also fix the parameter $s = \lceil \sqrt{q} \rceil$. Given a projection scheme h with image size s, our sampling algorithm first samples $\boldsymbol{Y} \in [s]^V$ from the projected distribution ν, and then uses it to sample a random hypergraph colouring from the conditional distribution $\mu^{\boldsymbol{Y}}$. The pseudocode is given in Algorithm 1.

The main ingredient of Algorithm 1 is the part that samples $\boldsymbol{Y}$ (Line 1 to Line 5). It is basically a *systematic scan* version of the Glauber dynamics for ν. In order to update the state of a particular vertex, we invoke a subroutine Sample, given in Algorithm 2, to sample X'_v first from the distribution conditional on $\boldsymbol{Y}_{V\setminus\{v\}}$. Also, Sample is used to generate the random colouring conditional on $\boldsymbol{Y}$ in Line 6. The subroutine Sample in fact returns an approximate sample with high probability. Here we have to settle with some small error because exactly calculating the conditional distribution is intractable. To implement Sample, we use standard rejection sampling, which is described in Algorithm 3. Showing the correctness and efficiency of Algorithm 2 and Algorithm 3 is one of our main contributions.

In the following we flesh out the outline above. Let $\Lambda \subseteq V$ and $\boldsymbol{Y}_\Lambda \in [s]^\Lambda$. Note that during the execution of Algorithm 1, $\boldsymbol{Y}_\Lambda$ is a random input to Sample. Let $S \subseteq V$ and $\zeta \in (0, 1)$. The subroutine $\mathsf{Sample}\,(H, h, S, \boldsymbol{Y}_\Lambda, \zeta)$ in Algorithm 1 returns a random sample $\boldsymbol{X}_S \in [q]^S$ such that with probability at least $1 - \zeta$, the total variation distance between $\boldsymbol{X}_S$ and $\mu_S^{\boldsymbol{Y}_\Lambda}$ is at most ζ, where the probability is taken over the randomness of the input $\boldsymbol{Y}_\Lambda$.

In the t-th step of the systematic scan in Algorithm 1, we pick the vertex v with label $(t \mod n)$, and use Line 4 and Line 5 to update the value of Y_v. Ideally, we want to resample the value of Y_v according to the conditional distribution $\nu_v^{\boldsymbol{Y}_{V\setminus\{v\}}}$, where ν is the distribution projected from μ. However, exactly computing the conditional distribution is not tractable, and we approximate it by projecting from the random sample $X'_v \in [q]$ given by Sample in Line 4. It is straightforward to verify that Y_v approximately follows the law of $\nu_v^{\boldsymbol{Y}_{V\setminus\{v\}}}$ as long as X'_v approximately follows the law of $\mu_v^{\boldsymbol{Y}_{V\setminus\{v\}}}$. In the last step, we use Sample to draw approximate samples from the conditional distribution $\mu^{\boldsymbol{Y}}$.

We explain the details of $\mathsf{Sample}\,(H, h, S, \boldsymbol{Y}_\Lambda, \zeta)$ next. First we need some notations. Given a partial image $\boldsymbol{Y}_\Lambda$, we say an hyperedge $e \in \mathcal{E}$ is satisfied by $\boldsymbol{Y}_\Lambda$ if there exists $u, v \in e \cap \Lambda$ such that $Y_u \neq Y_v$. In other words, for all $\boldsymbol{X} \in [q]^V$ such that $\boldsymbol{Y}_\Lambda = h(\boldsymbol{X}_\Lambda)$, the hyperedge e is not monochromatic with respect to $\boldsymbol{X}$, and thus e is always "satisfied" given

Algorithm 2 Sample $(H, h, S, \boldsymbol{Y}_\Lambda, \zeta)$.

Input: A hypergraph $H = (V, \mathcal{E})$, a projection scheme $h : [q] \to [s]$, a subset $S \subseteq V$, a (partial) image $\boldsymbol{Y}_\Lambda \in [s]^\Lambda$ where $\Lambda \subseteq V$, and an error bound $\zeta \in (0, 1)$
Output: A random (partial) colouring $\boldsymbol{X}_S \in [q]^S$

remove all hyperedges in H that are satisfied by $\boldsymbol{Y}_\Lambda$ to obtain $H^{\boldsymbol{Y}_\Lambda} = (V, \mathcal{E}^{\boldsymbol{Y}_\Lambda})$;
let $H_i = (V_i, \mathcal{E}_i^{\boldsymbol{Y}_\Lambda})$ for $1 \le i \le \ell$ be the connected components such that $V_i \cap S \neq \emptyset$;
if $\exists 1 \le i \le \ell$ *such that* $|\mathcal{E}_i^{\boldsymbol{Y}_\Lambda}| > 4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)$ **then**
  return $\boldsymbol{X}_S \in [q]^S$ *uniformly at random;*
for *i from 1 to* ℓ **do**
  $\boldsymbol{X}_i \leftarrow \mathsf{RejectionSampling}(H_i, h, \boldsymbol{Y}_{\Lambda \cap V_i}, R)$, where $R = \left\lceil 10 \left(\frac{n\Delta}{\zeta}\right)^{\frac{1}{1000\eta}} \log \frac{n}{\zeta} \right\rceil$;
  `/* The RejectionSampling subroutine is given in Algorithm 3. */`
  if $\boldsymbol{X}_i = \perp$ **then**
    return $\boldsymbol{X}_S \in [q]^S$ *uniformly at random ;*
return $\boldsymbol{X}_S$ *where* $\boldsymbol{X} = \biguplus_{i=1}^{\ell} \boldsymbol{X}_i$;

$\boldsymbol{Y}_\Lambda$. Let $H^{\boldsymbol{Y}_\Lambda} = (V, \mathcal{E}^{\boldsymbol{Y}_\Lambda})$ be the hypergraph obtained from H by removing all hyperedges satisfied by $\boldsymbol{Y}_\Lambda$. Let $H_1^{\boldsymbol{Y}_\Lambda}, H_2^{\boldsymbol{Y}_\Lambda}, \ldots, H_m^{\boldsymbol{Y}_\Lambda}$ denote the connected components of $H^{\boldsymbol{Y}_\Lambda}$, where $H_i^{\boldsymbol{Y}_\Lambda} = (V_i, \mathcal{E}_i^{\boldsymbol{Y}_\Lambda})$. The following fact is straightforward to verify

$$\mu^{\boldsymbol{Y}_\Lambda} = \mu_1^{\boldsymbol{Y}_{\Lambda \cap V_1}} \times \mu_2^{\boldsymbol{Y}_{\Lambda \cap V_2}} \times \ldots \times \mu_m^{\boldsymbol{Y}_{\Lambda \cap V_m}},$$

where μ_i is the uniform distribution over proper q-colourings of the sub-hypergraph $H_i^{\boldsymbol{Y}_\Lambda}$ (namely, $\mu_i^{\boldsymbol{Y}_{\Lambda \cap V_i}}$ is the uniform distribution over list colourings of $H_i^{\boldsymbol{Y}_\Lambda}$ conditional on $\boldsymbol{Y}_{\Lambda \cap V_i}$). Without loss of generality, we assume $S \cap V_j \neq \emptyset$ for $1 \le j \le \ell$. To draw a random sample from $\mu_S^{\boldsymbol{Y}_\Lambda}$, it suffices to draw a random sample from the product distribution $\mu_1^{\boldsymbol{Y}_{\Lambda \cap V_1}} \times \mu_2^{\boldsymbol{Y}_{\Lambda \cap V_2}} \times \ldots \times \mu_\ell^{\boldsymbol{Y}_{\Lambda \cap V_\ell}}$, which we will do by drawing from each $\mu_i^{\boldsymbol{Y}_{\Lambda \cap V_i}}$ individually using standard rejection sampling (given in Algorithm 3).

One final detail about Algorithm 2 and Algorithm 3 is about their efficiency. Basically we set some thresholds to guard against two unlikely bad events. We break out from the normal execution immediately and return an arbitrary random sample if one of the following two bad events occur:

- for some $1 \le i \le \ell$, $|\mathcal{E}_i^{\boldsymbol{Y}_\Lambda}| > 4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)$;
- for some $1 \le i \le \ell$, the rejection sampling for $\mu_i^{\boldsymbol{Y}_{\Lambda \cap V_i}}$ fails after R trials, where

$$R := \left\lceil 10 \left(\frac{n\Delta}{\zeta}\right)^{\frac{1}{1000\eta}} \log \frac{n}{\zeta} \right\rceil \qquad \text{and} \qquad \eta := \frac{1}{\Delta} \left(\frac{q}{100}\right)^{\frac{k-3}{2}}. \tag{4}$$

In the analysis (see Lemma 13), we will show that both of the two bad events above occur with low probability, and thus with high probability the Sample subroutine returns an approximate sample with desired accuracy.

Algorithm 3 RejectionSampling$(H, h, \boldsymbol{Y}_\Lambda, R)$.

Input: A hypergraph $H = (V, \mathcal{E})$, a projection scheme $h : [q] \to [s]$, a (partial) image $\boldsymbol{Y}_\Lambda \in [s]^\Lambda$ where $\Lambda \subseteq V$ and an integer R
Output: A random colouring $\boldsymbol{X} \in [q]^V$ or a special symbol $\perp$

for each $v \in V$, let $Q_v \leftarrow h^{-1}(Y_v)$ if $v \in \Lambda$, and $Q_v \leftarrow [q]$ if $v \notin \Lambda$;
for *i from 1 to R* **do**
  sample $X_v \in Q_v$ uniformly at random for all $v \in V$ and let $\boldsymbol{X} = (X_v)_{v \in V}$;
  if $\boldsymbol{X}$ *is a proper hypergraph colouring of* H **then**
    return $\boldsymbol{X}$;
return $\perp$;

4 Proof of the main theorem

Let $H = (V, \mathcal{E})$ be a simple k-uniform hypergraph with maximum degree Δ. Let $[q]$ be a set of q colours. Recall $s = \lceil \sqrt{q} \rceil$, where s is the parameter of projection scheme h (Definition 7). To construct h, we partition $[q]$ into s intervals, where the first $(q \bmod s)$ of them contains $\lceil q/s \rceil$ elements each while the rest contains $\lfloor q/s \rfloor$ elements each. For each $i \in [q]$, set

$$h(i) = j \qquad \text{where } i \text{ belongs to the } j\text{-th interval.} \tag{5}$$

Note that this h satisfies Definition 7. In our algorithm, h is implemented as an oracle, supporting the following two types of queries.

- Evaluation: given i, the oracle returns $h(i)$.
- Inversion: given j, the oracle returns a uniform element in $h^{-1}(j)$.

Obviously, each query can be answered in time $O(\log q)$ because of the construction of h.

The next theorem is a stronger form of Theorem 1. It shows that our algorithm can run in time arbitrarily close to linear in n, the number of vertices, as long as sufficiently many colours are available.

▶ **Theorem 10.** *The following result holds for any $\delta > 0$ and $0 < \alpha \le 1$. Given any $\epsilon \in (0, 1)$, any q-colouring instance on k-uniform simple hypergraph $H = (V, E)$ with maximum degree Δ, and a balanced projection scheme, if $k \ge \frac{20(1+\delta)}{\delta}$ and $q \ge 100 \left(\frac{\Delta}{\alpha}\right)^{\frac{2+\delta}{k-4/\delta-4}}$, Algorithm 1 returns a random colouring that is ϵ-close to μ in total variation distance in time $O\left(\Delta^2 k^5 n \left(\frac{n\Delta}{\epsilon}\right)^{\alpha/100} \log^4\left(\frac{n\Delta q}{\epsilon}\right)\right)$.*

▶ Remark 11. The parameter α captures the relation between the local lemma condition and the running time of the algorithm. If α becomes smaller, the condition is more confined, and the running time is closer to linear. In particular, Theorem 1 is implied by setting $\alpha = 1$.

We need two lemmas to prove Theorem 10. The first lemma analyses the mixing time of the idealised systematic scan. Let ν be the projected distribution. The idealised systematic scan chain $\boldsymbol{P}_{scan}$ for ν is defined as follows. Initially, let $\boldsymbol{X}_0 \in [s]^V$ be an arbitrary initial configuration. In the t-th step, the systematic scan does the following update steps.

- Pick the vertex $v \in V$ with label $(t \mod n)$ and let $X_t(V \setminus \{v\}) \leftarrow X_{t-1}(V \setminus \{v\})$.
- Sample $X_t(v) \sim \nu_v^{\boldsymbol{X}_{t-1}(V \setminus \{v\})}$.

▶ **Lemma 12.** *If $q \ge 40\Delta^{\frac{2}{k-4}}$ and $k \ge 20$, the systematic scan chain $\boldsymbol{P}_{scan}$ for ν is irreducible, aperiodic and reversible with respect to ν. Furthermore, the mixing time satisfies*

$$\forall 0 < \epsilon < 1, \quad T_{\mathrm{mix}}(\boldsymbol{P}_{scan}, \epsilon) \le \left\lceil 50 n \log \frac{n\Delta}{\epsilon} \right\rceil.$$

Lemma 12 is shown in Section 6.

Our next lemma analyzes the Sample subroutine. Let $(\boldsymbol{Y}_t)_{t=0}^{T}$ denote the sequence of random configurations in $[s]^V$ generated by Algorithm 1, where $\boldsymbol{Y}_0 \in [s]^V$ is the initial configuration and $\boldsymbol{Y}_t$ is the configuration after the t-th iteration of the for-loop. For any $1 \le t \le T+1$, consider the t-th invocation of Sample and define the following two bad events:

- $\mathcal{B}_{\text{com}}(t)$: in the t-th invocation, $\boldsymbol{X}_S$ is returned by Line 4 in Algorithm 2;
- $\mathcal{B}_{\text{rej}}(t)$: in the t-th invocation, $\boldsymbol{X}_S$ is returned by Line 8 in Algorithm 2.

Note that the $(T+1)$-th invocation of the subroutine Sample is in Line 6 in Algorithm 1. Let $H = (V, \mathcal{E})$ denote the input hypergraph of Algorithm 1.

▶ **Lemma 13.** *For $1 \le t \le T+1$, the t-th invocation of the subroutine* Sample $(H, h, S, \boldsymbol{Y}_\Lambda, \zeta)$, *where h is given by (5), satisfies*

1. *the running time of the subroutine is bounded by $O\left(|S|\Delta^2 k^5 \left(\frac{n\Delta}{\zeta}\right)^{\frac{1}{1000\eta}} \log^3\left(\frac{n\Delta q}{\zeta}\right)\right)$;*
2. *conditional on neither $\mathcal{B}_{\text{com}}(t)$ nor $\mathcal{B}_{\text{rej}}(t)$ occurs, the subroutine returns a perfect sample from $\mu_S^{\boldsymbol{Y}_\Lambda}$;*
3. *if $q \ge 100\Delta^{\frac{2}{k-3}}$ and $k \ge 20$, then $\Pr[\mathcal{B}_{\text{rej}}(t)] \le \zeta$;*
4. *for any $\delta > 0$, if $k \ge \frac{20(\delta+1)}{\delta}$, $q \ge 100\Delta^{\frac{2+\delta}{k-4/\delta-3}}$, and H is simple, then $\Pr[\mathcal{B}_{\text{com}}(t)] \le \zeta$.*

Properties 1, 2 and 3 are very standard and hold in general hypergraphs. They can be established by mimicking the argument in [8, 9]. The challenge here is to prove Property 4, which is established in Section 5.

Given Lemma 12 and Lemma 13, it is straightforward to establish Theorem 10. Lemma 12 shows that the idealized systematic scan chain is rapidly mixing, and Lemma 13 shows that our implementation of the chain is efficient. Lemma 13 also shows that the exceptions and errors in our implementation have low probability to happen. Thus, by a coupling argument and the coupling lemma, Lemma 4, the output of our algorithm is within ϵ total variation distance to the desired one.

5 Analysis of connected components

In this section, we provide a proof sketch for Property 4 in Lemma 13. Note that this property needs the input hypergraph H being simple. Fix $1 \le t \le T+1$. Consider the t-th invocation of the subroutine Sample. If $1 \le t \le T$, we use v_t to denote the vertex picked by the t-th step of the systematic scan, i.e. v_t is the vertex with label $(t \mod n)$, and let $\Lambda := V \setminus \{v_t\}$. If $t = T+1$, let $\Lambda := V$. Recall that $\boldsymbol{Y}_t \in [s]^V$ is the random configuration generated by Algorithm 1 after the t-th iteration of the for-loop. To simplify the notation, we define $\boldsymbol{Y} = \boldsymbol{Y}_{t-1}(\Lambda)$, so that the input partial configuration to Sample is $\boldsymbol{Y}$ (see Algorithm 1). Hence, we consider the subroutine Sample $(H, h, S, \boldsymbol{Y}, \zeta)$. Note that $\boldsymbol{Y} \in [s]^\Lambda$ is a random configuration, and therefore $H^{\boldsymbol{Y}}$ is a random hypergraph, where $H^{\boldsymbol{Y}}$ is obtained by removing all the hyperedges in H satisfied by $\boldsymbol{Y}$. Fix an arbitrary vertex $v \in V$. We use $H_v^{\boldsymbol{Y}} = (V_v^{\boldsymbol{Y}}, \mathcal{E}_v^{\boldsymbol{Y}})$ to denote the connected component in $H^{\boldsymbol{Y}}$ that contains the vertex v. Note that $\mathcal{E}_v^{\boldsymbol{Y}}$ can be an empty set. A hyperedge $e \in \mathcal{E}$ is *incident* to v in the hypergraph H if $v \in e$. We prove the following lemma, which implies Property 4 by a straightforward union bound.

▶ **Lemma 14.** *For any $\delta > 0$, if $k \ge \frac{20(1+\delta)}{\delta}$, $q \ge 100\Delta^{\frac{2+\delta}{k-4/\delta-3}}$, and H is simple, then for any $v \in V$, any e incident to v in H, it holds that*

$$\Pr_{\boldsymbol{Y}}\left[e \in \mathcal{E}_v^{\boldsymbol{Y}} \wedge |\mathcal{E}_v^{\boldsymbol{Y}}| \ge 4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)\right] \le \frac{\zeta}{n\Delta}.$$

Denote by $L_H = (V_L, E_L) = \mathrm{Lin}(H)$ the line graph of H (recall Definition 3). Let e be the hyperedge in Lemma 14 and let $u = u_e$ be the vertex in L_H corresponding to e. Let $L_H^{\boldsymbol{Y}} = (V_L^{\boldsymbol{Y}}, E_L^{\boldsymbol{Y}})$ denote the line graph of $H^{\boldsymbol{Y}}$. Note that $L_H^{\boldsymbol{Y}}$ is random, and the randomness of $L_H^{\boldsymbol{Y}}$ is determined by the randomness of $\boldsymbol{Y}$.

Let $\mathcal{C} \subseteq V_L$ denote the random set of all vertices in the connected component of $L_H^{\boldsymbol{Y}}$ that contains the vertex u. If $u \notin V_L^{\boldsymbol{Y}}$, let $\mathcal{C} = \emptyset$. Define an integer parameter $\theta := \lceil \frac{4}{\delta} \rceil$. To prove Lemma 14, it suffices to show that

$$\forall M > \theta, \quad \Pr_{\boldsymbol{Y}}\left[|\mathcal{C}| \geq M\right] \leq \left(\frac{1}{2}\right)^{\frac{M}{2\theta k^2 \Delta}-1}. \tag{6}$$

This is because $k \geq \frac{20(\delta+1)}{\delta} > \lceil \frac{4}{\delta} \rceil + 1 = \theta + 1$, and setting $M = 4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)$ proves Lemma 14.

Define the following collection of subsets

$$\mathrm{Con}_u(M) := \{C \subseteq V_L \mid u \in C \wedge |C| = M \wedge L_H[C] \text{ is connected}\}.$$

It is straightforward to verify that

$$\Pr_{\boldsymbol{Y}}\left[|\mathcal{C}| \geq M\right] \leq \Pr_{\boldsymbol{Y}}\left[\exists C \in \mathrm{Con}_u(M) \text{ s.t. } C \subseteq V_L^{\boldsymbol{Y}}\right].$$

In our proof, we partition the set $\mathrm{Con}_u(M)$ into two disjoint subsets $\mathrm{Con}_u^{(1)}(M)$ and $\mathrm{Con}_u^{(2)}(M)$, and bound their corresponding probabilities separately, by showing

$$\Pr_{\boldsymbol{Y}}\left[\exists C \in \mathrm{Con}_u^{(1)}(M) \text{ s.t. } C \subseteq V_L^{\boldsymbol{Y}}\right] < \left(\frac{1}{2}\right)^{\frac{M}{2\theta k^2 \Delta}}; \tag{7}$$

$$\Pr_{\boldsymbol{Y}}\left[\exists C \in \mathrm{Con}_u^{(2)}(M) \text{ s.t. } C \subseteq V_L^{\boldsymbol{Y}}\right] < \left(\frac{1}{2}\right)^{M}. \tag{8}$$

We use Algorithm 4 to partition the set $\mathrm{Con}_u(M)$. Taking as an input any $C \in \mathrm{Con}_u(M)$, Algorithm 4 outputs an integer $\ell = \ell(C)$ and disjoint sets $C_1, C_2, \ldots, C_\ell \subseteq C$. The set C falls into $C \in \mathrm{Con}_u^{(1)}(M)$ if $\ell(C) \geq \frac{M}{2\theta k^2 \Delta}$, and $\mathrm{Con}_u^{(2)}(M)$ otherwise. We remark that Algorithm 4 is only used for analysis, and we do not need to implement this algorithm.

To characterise the output of Algorithm 4, we introduce a notion called "2-block-tree".

▶ **Definition 15** (2-block-tree). *Let $\theta \geq 1$ be an integer. Let $G = (V, E)$ be a graph. A set $\{C_1, C_2, \ldots, C_\ell\}$ is a* 2-block-tree *with block size θ and tree size ℓ in G if*

1. *for any $1 \leq i \leq \ell$, $C_i \subseteq V$, $|C_i| = \theta$, and the induced subgraph $G[C_i]$ is connected;*
2. *for any distinct $1 \leq i, j \leq \ell$, $\mathrm{dist}_G(C_i, C_j) \geq 2$;*
3. *$\{C_1, \cdots, C_\ell\}$ is connected on G^2. (Recall Definition 2 of graph powers.)*

One can easily observe that the notion of 2-block-trees is a generalisation of 2-trees in [1] by setting $\theta = 1$. According to the next proposition, the output of Algorithm 4 is a 2-block-tree in L_H. This explains the name "2-block-tree generator".

▶ **Proposition 16.** *The output $\{C_1, C_2, \ldots, C_\ell\}$ of Algorithm 4 satisfies that*

1. *$\{C_1, C_2, \ldots, C_\ell\}$ is a 2-block-tree in L_H with block size θ satisfying $u \in C_1$ and $\cup_{i=1}^{\ell} C_i \subseteq C$;*
2. *if all vertices in $\Gamma_G(C_i)$ are removed from G, where $G = L_H[C]$, then the resulting graph $G[C']$ is a collection of connected components whose sizes are at most θ, where $C' = C \setminus (\cup_{i=1}^{\ell} \Gamma_G(C_i))$.*

Algorithm 4 2-block-tree generator.

Input: the parameter $\delta \in (0,1)$ in Lemma 14, the line graph L_H, an integer $M > \theta$, a vertex u in L_H, and a subset $C \in \mathrm{Con}_u(M)$
Output: an integer ℓ and connected subgraphs $C_1, \cdots, C_\ell \subseteq C$

let $G = L_H[C] = (C, E_C)$ be the subgraph of L_H induced by C;
$\theta \leftarrow \lceil \frac{4}{\delta} \rceil$, $\ell \leftarrow 0$, $V \leftarrow C$;
while $|V| \geq \theta$ **do**
  $\ell \leftarrow \ell + 1$;
  if $\ell = 1$ **then** $u_\ell \leftarrow u$;
  if $\ell > 1$ **then** let u_ℓ be an arbitrary vertex in $\Gamma_G(C \setminus V)$;
  let $C_\ell \subseteq V$ be an arbitrary connected subgraph in G such that $|C_\ell| = \theta$ and $u_\ell \in C_\ell$;
  $V \leftarrow V \setminus (C_\ell \cup \Gamma_G(C_\ell))$;
  for *each connected component* $G' = (V', E')$ *in* $G[V]$ *such that* $|V'| < \theta$ **do**
    $V \leftarrow V \setminus V'$;
return $\ell, \{C_1, C_2, \ldots, C_\ell\}$;

To prove (7), it is not hard to see that for any $C \in \mathrm{Con}_u^{(1)}(M)$, there is a 2-block-tree tree $\{C_1, C_2, \ldots, C_\ell\}$ in the line graph L_H with block size θ and tree size ℓ satisfying

$$u \in C_1 \cup C_2 \cup \ldots \cup C_\ell \quad \text{and} \quad C_1 \cup C_2 \cup \ldots \cup C_\ell \subseteq C \tag{9}$$

where $\ell = \lceil \frac{M}{2\theta k^2 \Delta} \rceil$. We denote a 2-block-tree tree with block size θ and tree size ℓ by (θ, ℓ)-2-block-tree. This implies that

$$\begin{aligned} &\Pr_{\boldsymbol{Y}}\left[\exists\, C \in \mathrm{Con}_u^{(1)}(M) \text{ s.t. } C \subseteq V_L^{\boldsymbol{Y}}\right] \\ &\leq \Pr_{\boldsymbol{Y}}\left[\exists\, (\theta,\ell)\text{-2-block-tree } \{C_i\} \text{ in } L_H \text{ satisfying (9) s.t. } \forall 1 \leq i \leq \ell, C_i \subseteq V_L^{\boldsymbol{Y}}\right]. \end{aligned}$$

We apply a union bound on the RHS over all possible 2-block-trees. The probability of any 2-block-tree is bounded by

$$\Pr_{\boldsymbol{Y}}\left[\forall 1 \leq i \leq \ell, C_i \subseteq V_L^{\boldsymbol{Y}}\right] \leq \left((\mathrm{e}s)^\theta \left(\frac{1}{s} + \frac{1}{q}\right)^{\theta(k-\theta)}\right)^\ell. \tag{10}$$

To bound the number of 2-block-trees, we use the following lemma.

▶ **Lemma 17.** *Let $\theta \geq 1$ be an integer. Let $G = (V, E)$ be a graph with maximum degree d. For any integer $\ell \geq 1$, any vertex $v \in V$, the number of 2-block-trees $\{C_1, C_2, \ldots, C_\ell\}$ with block size θ and tree size ℓ such that $v \in \cup_{i=1}^{\ell} C_i$ is at most $(\theta \mathrm{e}^\theta d^{\theta+1})^\ell$.*

One may attempt Lemma 17 using the count of connected components with a degree bound in [4] based on [33]. However, it is too loose to our need, and our refined estimation in Lemma 17 is shown by a more complicated encoding argument. Our encoding has $\ell + 1$ components. The first one encodes how C_i's are connected in G^2, and the rest encodes each individual C_i. To encode, we need to carefully perform a depth-first-search (DFS) in G and generate an encoding for C_i whenever we meet one. The DFS generates a tree encoding how C_i's are connected in G^2. This way, we can uniquely recover the original 2-block-tree and map each component to some subtree of a suitable infinite tree to bound their numbers.

The inequality (7) follows directly from (10) and Lemma 17.

To prove (8), we take a union bound directly over connected components of size M. Using Algorithm 4, we can get a good lower bound on the number of distinct vertices in the original hypergraph for a component. This implies that each component in $\mathrm{Con}_u^{(2)}(M)$ happens with probability much smaller than the total number of such components, and the union bound succeeds.

6 Mixing of systematic scan

In this section, we give the proof sketch of the mixing lemma for the projected systematic scan Markov chain of hypergraph colourings (Lemma 12). It is straightforward to verify that the systematic scan is aperiodic and reversible with respect to ν. Irreducibility follows from the local lemma, Theorem 5. More precisely, Theorem 5 implies that for any $\tau \in [s]^V$, $\nu(\tau) > 0$ if $q \geq 40\Delta^{\frac{2}{k-4}}$ and $k \geq 20$. Hence, the systematic scan has the unique stationary distribution ν.

For the mixing time, the analysis is based on an information percolation argument. Define a coupling $\mathcal{C}$ of the systematic scan $(\boldsymbol{X}_t, \boldsymbol{Y}_t)_{t\geq 0}$. Let $\boldsymbol{X}_0, \boldsymbol{Y}_0 \in [s]^V$ be two arbitrary initial configurations. In the t-th transition step,

- let $v \in V$ be the vertex with label $(t \mod n)$ and set $(X_t(u), Y_t(u)) \leftarrow (X_{t-1}(u), Y_{t-1}(u))$ for all other vertices $u \in V \setminus \{v\}$;
- sample $(X_t(v), Y_t(v))$ from the optimal coupling between $\nu_v^{X_{t-1}(V\setminus\{v\})}$ and $\nu_v^{Y_{t-1}(V\setminus\{v\})}$.

We prove the following lemma in this section.

▶ **Lemma 18.** *Suppose $k \geq 20$ and $q \geq 40\Delta^{\frac{2}{k-4}}$. For any initial configurations $\boldsymbol{X}_0, \boldsymbol{Y}_0 \in [s]^V$, any $\epsilon \in (0,1)$, let $T = \lceil 50n \log \frac{n\Delta}{\epsilon} \rceil$, it holds that*

$$\forall v \in V, \quad \Pr_{\mathcal{C}}\left[X_T(v) \neq Y_T(v)\right] \leq \frac{\epsilon}{n}.$$

By Lemma 18, a union bound over all vertices and the coupling lemma (Lemma 4), it holds that $\max_{\boldsymbol{X}_0, \boldsymbol{Y}_0 \in [s]^V} d_{\mathrm{TV}}(\boldsymbol{X}_T, \boldsymbol{Y}_T) \leq \Pr_{\mathcal{C}}\left[\boldsymbol{X}_T \neq \boldsymbol{Y}_T\right] \leq \epsilon$, which proves the mixing time part of Lemma 12 via (1). In the rest of this section, we use the information percolation technique to analyse the coupling $\mathcal{C}$ and prove Lemma 18.

Consider the coupling procedure $(\boldsymbol{X}_t, \boldsymbol{Y}_t)_{t\geq 0}$. For each $t \geq 1$, let v_t denote the vertex picked in the t-th step of systematic scan, namely, v_t is the vertex with label $(t \mod n)$. Consider the t-th transition step, where $t > 0$. Define the set of agreement vertices when updating v_t at time t by $A_t := \{v \in V \setminus \{v_t\} \mid X_{t-1}(v) = Y_{t-1}(v)\}$. We say a hyperedge $e \in \mathcal{E}$ is satisfied by A_t if there exist two distinct vertices $u, v \in e \cap A_t$ such that $X_{t-1}(u) \neq X_{t-1}(v)$ (and hence $Y_{t-1}(u) \neq Y_{t-1}(v)$). We remove all the hyperedges $e \in \mathcal{E}$ satisfied by A_t to obtain a sub-hypergraph H_t. Let H_t^v denote the connected component in H_t containing v.

▶ **Lemma 19.** *If $X_t(v_t) \neq Y_t(v_t)$ for some $t \geq 1$, then there exists $u \neq v_t$ in $H_t^{v_t}$ such that $X_{t-1}(u) \neq Y_{t-1}(u)$.*

Lemma 19 can be proved by contradiction. Note that $X_t(v_t)$ (resp. $Y_t(v_t)$) depends only on the configuration of X_{t-1} (resp. $Y_{t-1}(v_t)$) restricted on the vertices in $H_t^{v_t}$. If X_{t-1} and Y_{t-1} are the same on the vertices in $H_t^{v_t}$, then $X_t(v_t)$ and $Y_t(v_t)$ must be coupled perfectly.

We say that a hyperedge sequence $e_1, e_2, \ldots, e_\ell$ is a path in a hypergraph if for each $1 < i \leq \ell$, $e_{i-1} \cap e_i \neq \emptyset$ and $e_{i-1} \neq e_i$. The following result is a straightforward corollary of Lemma 19.

▶ **Corollary 20.** *Let $t \geq 1$. If $X_t(v_t) \neq Y_t(v_t)$, then there exists a vertex $u \neq v_t$ satisfying $X_{t-1}(u) \neq Y_{t-1}(u)$ and a path $e_1, e_2, \ldots, e_\ell$ in hypergraph H such that*

- *$v \in e_1$ and $u \in e_\ell$;*
- *for any hyperedge e_i in the path, there exists $c \in [s]$ such that for all vertex $w \in e_i$ and $w \neq v_t$, either $X_{t-1}(w) = Y_{t-1}(w) = c$ or $X_{t-1}(w) \neq Y_{t-1}(w)$.*

Corollary 20 is a key result for the information percolation analysis. For any time $0 \leq t \leq T$, any vertex $v \in V$, define the set of previous update times by $S(t, v) := \{1 \leq i \leq t \mid v_i = v\}$, where v_i is the vertex picked in the i-th transition step. Define the last update time for v up to t by

$$\mathsf{time}_{\mathsf{ud}}(t, v) := \begin{cases} \max_{i \in S(t,v)} i & \text{if } S(t, v) \neq \emptyset; \\ 0 & \text{otherwise.} \end{cases}$$

By Corollary 20, if the coupling on vertex v failed at time t, then there must exist a vertex u such that the coupling on u failed at time $t' = \mathsf{time}_{\mathsf{ud}}(t, u)$. We apply Corollary 20 recursively until we find a vertex w such that $X_0(w) \neq Y_0(w)$. This gives us an update time sequence $t = t_1 > t_2 > \ldots > t_\ell = 0$ such that the coupling of each t_i-th transition fails, together with a set of paths satisfying the properties in Corollary 20. We will show that such an update time sequence and the set of paths occur with small probability, which bounds the probability of $X_t(v_t) \neq Y_t(v_t)$. For this analysis, we will use the notions of extended hyperedges and extended hypergraphs introduced by He, Sun, and Wu [19].

Fix an integer $T \geq 1$ to be the total number of transitions of the systematic scan. Define the set of extended vertex V^{ext} by

$$V^{\text{ext}} = \{(t, v_t) \mid 1 \leq t \leq T\} \cup \{(0, v) \mid v \in V\},$$

where v_t is the vertex with label ($t \mod n$). Each vertex $(t, u) \in V^{\text{ext}}$ represents an update, i.e. u is updated at the t-th transition step. We regard all vertices "updated" at the initial step ($t = 0$). Consider the systematic scan process $(\boldsymbol{X}_t)_{t \geq 0}$. For any hyperedge $e \in \mathcal{E}$, the configuration $X_t(e)$ of e at time t satisfies for all $u \in e$, $X_t(u) = X_{t'}(u)$, where $t' = \mathsf{time}_{\mathsf{ud}}(t, u)$, namely, the value of u at time t is the same as the value of u at time $t' = \mathsf{time}_{\mathsf{ud}}(t, u)$. Besides, the configuration of hyperedge e remains unchanged until some vertex in e is updated. This motivates the following definition.

▶ **Definition 21.** *The set $\mathcal{E}^{\text{ext}}$ of* extended hyperedges *is defined by $\mathcal{E}^{\text{ext}} := \cup_{t=0}^{T} \mathcal{E}_t^{\text{ext}}$, where*

$$\mathcal{E}_0^{\text{ext}} := \bigcup_{e \in \mathcal{E}} \{(0, v) \mid v \in e\},$$

$$\forall 1 \leq t \leq T, \quad \mathcal{E}_t^{\text{ext}} := \bigcup_{e : v_t \in e} \{(t', v) \mid v \in e \wedge t' = \mathsf{time}_{\mathsf{ud}}(t, v)\}.$$

The extended hypergraph *is $H^{\text{ext}} = (V^{\text{ext}}, \mathcal{E}^{\text{ext}})$.*

At the beginning, each hyperedge $e \in \mathcal{E}$ takes its initial value, and thus we add all the extended hyperedges with $t = 0$ to $\mathcal{E}_0^{\text{ext}}$. For each update at time $1 \leq t \leq T$, only the value of v_t is updated. Thus the configurations of only the hyperedges containing v_t are updated, and we add only those to $\mathcal{E}_t^{\text{ext}}$.

Corollary 20 shows that for any $t \geq 1$, if the coupling in the t-th transition step fails (i.e. $X_t(v_t) \neq Y_t(v_t)$), then we can find a specific path in the hypergraph H. Our next lemma lifts such a path to H^{ext}. Note that there is a slight difference regarding v_t comparing to Corollary 20.

▶ **Lemma 22.** *Let $1 \le t \le T$ be an integer. Suppose $X_t(v_t) \neq Y_t(v_t)$. There exist a vertex $(t', u) \in V^{\text{ext}}$ satisfying $t' < t$ and $X_{t'}(u) \neq Y_{t'}(u)$, together with a path $e_1^{\text{ext}}, e_2^{\text{ext}}, \ldots, e_\ell^{\text{ext}}$ in H^{ext} such that*

- *$(t, v_t) \in e_1^{\text{ext}}$ and $(t', u) \in e_\ell^{\text{ext}}$;*
- *for any hyperedge e_i^{ext} in the path, there exists $c \in [s]$ such that for all $(j, w) \in e_i^{\text{ext}}$, either $X_j(w) = Y_j(w) = c$ or $X_j(w) \neq Y_j(w)$.*

We may repeatedly apply Lemma 22 to trace a discrepancy from some time t to time 0. By pruning "cycles" (in some hypergraph sense) from such a path from t to 0 in H^{ext}, we can find a path satisfying the properties in the following lemma.

▶ **Lemma 23.** *Let $1 \le t \le T$ be an integer. Suppose $X_t(v_t) \neq Y_t(v_t)$. There exists a path $e_1^{\text{ext}}, e_2^{\text{ext}}, \ldots, e_\ell^{\text{ext}}$ in the extended hypergraph H^{ext} such that*

- *$(t, v_t) \in e_1^{\text{ext}}$, $\min\{j \mid (j, w) \in e_i^{\text{ext}}\} > 0$ for all $i < \ell$ and $\min\{j \mid (j, w) \in e_\ell^{\text{ext}}\} = 0$;*
- *for any $1 \le i, i' \le \ell$ satisfying $|i - i'| \ge 2$, $e_i^{\text{ext}} \cap e_{i'}^{\text{ext}} = \emptyset$;*
- *for any hyperedge e_i^{ext} in the path, there exists $c \in [s]$ such that for all $(j, w) \in e_i^{\text{ext}}$, either $X_j(w) = Y_j(w) = c$ or $X_j(w) \neq Y_j(w)$.*

Finally, we give the proof sketch of Lemma 18.

Proof sketch of Lemma 18. Let $t = \mathsf{time}_{\mathsf{ud}}(T, v) \ge \left\lceil 40n \log \frac{n}{\epsilon} \right\rceil$. We only need to bound the probability of $X_t(v) \neq Y_t(v)$. Fix a path $\mathcal{P} = e_1^{\text{ext}}, e_2^{\text{ext}}, \ldots, e_\ell^{\text{ext}}$ satisfying the first two properties in Lemma 23. Call $\mathcal{P}$ bad if $\mathcal{P}$ satisfies the third property in Lemma 23. We bound the probability of $\mathcal{P}$ being bad, and then take a union bound over all possible paths.

To bound the probability, we truncate the last extended hyperedge e_ℓ^{ext} in $\mathcal{P}$ to obtain a new path $\mathcal{P}'$ of length $\ell - 1$. By using the local lemma, we can show that

$$\Pr[\mathcal{P} \text{ is bad}] \le \left(\frac{1.16}{\sqrt{q}} \left(1 + \frac{5}{k} \right) \right)^{N(\mathcal{P}')},$$

where $N(\mathcal{P}')$ denotes the number of distinct extended vertices in $\mathcal{P}'$ and the constant 1.16 comes from comparing $s = \lceil \sqrt{q} \rceil$ and $\sqrt{q}$.

In our analysis, we need to use some detailed structure of the extended hypergraph. Each $e^{\text{ext}} \in \mathcal{E}^{\text{ext}}$ corresponds to a unique hyperedge $\mathsf{edge}(e^{\text{ext}}) \in \mathcal{E}$ in the input hypergraph, or more formally, $\mathsf{edge}(e^{\text{ext}}) := \{v \mid (t, v) \in e^{\text{ext}}\}$. For two adjacent extended hyperedges $e^{\text{ext}}, f^{\text{ext}} \in \mathcal{E}^{\text{ext}}$, e^{ext} is an out-neighbor of f^{ext} if $\mathsf{edge}(e^{\text{ext}}) \neq \mathsf{edge}(f^{\text{ext}})$, and e^{ext} is a self-neighbor of f^{ext} if $\mathsf{edge}(e^{\text{ext}}) = \mathsf{edge}(f^{\text{ext}})$. For each e^{ext}, the number of out-neighbours is at most Δk^2, whereas the number of self-neighbours is at most $2k$.

Back to the path $\mathcal{P}'$. Consider two adjacent extended hyperedges e_i^{ext} and e_{i+1}^{ext} in $\mathcal{P}'$.

- If e_i^{ext} is an out-neighbour of e_{i+1}^{ext}, then e_i^{ext} and e_{i+1}^{ext} share at most one extended vertex because the input hypergraph is simple. In this case, $\mathcal{P}'$ has many distinct extended vertices. Hence, we can control the probability that $\mathcal{P}$ is bad.
- If e_i^{ext} is an self-neighbour of e_{i+1}^{ext}, then e_i^{ext} and e_{i+1}^{ext} may share a lot of extended vertices. In this case we have to choose non-consecutive hyperedges to bound the number of distinct extended vertices. However, thankfully, given e_i^{ext}, there are at most $2k$ choices for e_{i+1}^{ext}, and there are not very many paths with many self-neighbours inside.

With these two cases analysed, to apply the union bound and finish the proof, we parametrise the number of self-neighbours in a path and show that the number of all paths does not outweigh the probability of them being bad. ◀

References

1 Noga Alon. A parallel algorithmic version of the local lemma. *Random Struct. Algorithms*, 2(4):367–378, 1991.

2 József Beck. An algorithmic approach to the Lovász local lemma. I. *Random Struct. Algorithms*, 2(4):343–366, 1991.

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

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

5 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.

6 P. Erdős and L. 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. North-Holland, 1975.

7 Weiming Feng, Heng Guo, and Jiaheng Wang. Improved bounds for randomly colouring simple hypergraphs. *CoRR*, abs/2202.05554, 2022. `arXiv:2202.05554`.

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

9 Weiming Feng, Kun He, and Yitong Yin. Sampling constraint satisfaction solutions in the local lemma regime. *arXiv*, abs/2011.03915, 2020. `arXiv:2011.03915`.

10 Alan Frieze and Dhruv Mubayi. Coloring simple hypergraphs. *J. Combin. Theory Ser. B*, 103(6):767–794, 2013.

11 Alan M. Frieze and Michael Anastos. Randomly coloring simple hypergraphs with fewer colors. *Inf. Process. Lett.*, 126:39–42, 2017. `doi:10.1016/j.ipl.2017.06.005`.

12 Alan M. Frieze and Páll Melsted. Randomly coloring simple hypergraphs. *Inf. Process. Lett.*, 111(17):848–853, 2011. `doi:10.1016/j.ipl.2011.06.001`.

13 Andreas Galanis, Leslie Ann Goldberg, Heng Guo, and Kuan Yang. Counting solutions to random CNF formulas. *SIAM J. Comput.*, 50(6):1701–1738, 2021.

14 Andreas Galanis, Heng Guo, and Jiaheng Wang. Inapproximability of counting hypergraph colourings. *arXiv preprint*, 2021. `arXiv:2107.05486`.

15 Heidi Gebauer, Tibor Szabó, and Gábor Tardos. The local lemma is asymptotically tight for SAT. *J. ACM*, 63(5):43:1–43:32, 2016.

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

17 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.

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

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

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

21 Mark Huber. Exact sampling and approximate counting techniques. In *STOC*, pages 31–40. ACM, 1998.

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

23 Vishesh Jain, Huy Tuan Pham, and Thuy Duong Vuong. Towards the sampling Lovász local lemma. *arXiv*, abs/2011.12196, 2020. `arXiv:2011.12196`.

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

25 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.

26 Vladimir Kolmogorov. A faster approximation algorithm for the Gibbs partition function. In *COLT*, pages 228–249. PMLR, 2018. URL: `http://proceedings.mlr.press/v75/kolmogorov18a.html`.

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

28 Michael Molloy and Bruce A. Reed. Further algorithmic aspects of the local lemma. In *STOC*, pages 524–529. ACM, 1998.

29 Robin A. Moser. A constructive proof of the Lovász local lemma. In *STOC*, pages 343–350. ACM, 2009.

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

31 James G. Propp and David B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures Algorithms*, 9(1-2):223–252, 1996.

32 Aravind Srinivasan. Improved algorithmic versions of the Lovász local lemma. In *SODA*, pages 611–620. SIAM, 2008.

33 Richard P. Stanley. *Enumerative combinatorics. Vol. 2*, volume 62 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1999.

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

35 Ian M Wanless and David R Wood. A general framework for hypergraph colouring. *arXiv preprint*, 2020. `arXiv:2008.00775`.