# Improved bounds for randomly colouring simple hypergraphs*

Weiming Feng† Heng Guo† Jiaheng Wang†

**Abstract**

We study the problem of sampling almost uniform proper $q$-colourings in $k$-uniform simple hypergraphs with maximum degree $\Delta$. 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).

## 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 [EL75] 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 [Bec91, Alo91, MR98, CS00, Sri08, Mos09], the celebrated work of Moser and Tardos [MT10] 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 the sampling problem can be **NP**-hard even when the criterion of the local lemma is satisfied and the corresponding searching problem lies in **P** [BGG+19, GGW21]. 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 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 [Moi19, GJL19, GLLZ19, JPV20], and in particular the projected Markov chain approach [FGYZ21, FHY20, JPV21, HSW21]. For searching problems, the local lemma is known to give a sharp computational transition threshold from **P** to **NP**-hard [MT10, GST16] 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 [EL75], and has received considerable recent attention. 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

*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)

†School of Informatics, University of Edinburgh, Informatics Forum, Edinburgh, EH8 9AB, United Kingdom. E-mails: wfeng@ed.ac.uk, hguo@inf.ed.ac.uk, jiaheng.wang@ed.ac.uk

from $q$, $k$, and $\Delta$) for $k$-uniform hypergraphs with maximum degree $\Delta$ [JPV21, HSW21], while the sampling problem is **NP**-hard whenever $q \lesssim \Delta^{2/k}$ for even $q$ [GGW21]. For comparison, the local lemma shows that a proper $q$-colouring exists if $q \gtrsim \Delta^{1/(k-1)}$ (see also [WW20] 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[1] has already been studied [FM11, FA17]. In particular, Frieze and Anastos [FA17] 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 [JPV21, HSW21] for general hypergraphs, and does not require unbounded number of colours, unlike in [FM11, FA17]. Let $\mu$ 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 $\Delta$, 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 $\epsilon$-close to $\mu$ 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 [Hub98] based on coupling from the past (CFTP) [PW96], following the same lines of [HSW21]. Moreover, using known reductions from approximate counting to sampling [JVV86, ŠVV09, Hub15, Kol18] (see [FGYZ21] 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 [FGYZ21] with state compression [FHY20]. 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 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[2] 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 [HSZ19, JPV21, HSW21], 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

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

[2]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.

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 [HSZ19], 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 [Alo91]. 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 $\theta$ 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 $\Delta$ 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 [GGW21] for general $q$, and by the algorithm of Frieze and Anastos [FA17] for $q = \Omega(\log n)$. Note that for a simple $k$-uniform hypergraph with maximum degree $\Delta$, Frieze and Mubayi [FM13] showed that the chromatic number $\chi(H) \le C_k \left(\frac{\Delta}{\log \Delta}\right)^{\frac{1}{k-1}}$ 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 may deteriorate slightly but would still be better than those for general hypergraphs. 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 [GJL19] almost matches the hardness result [GGW21]. 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.

## 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.

### 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 $G$.

- $\Gamma_G^i(A)$: 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 $\Omega$ and two distributions $\mu$ and $\nu$ over it. The *total variation distance* between $\mu$ and $\nu$ is defined by

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

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

**Lemma 4** (Coupling lemma). *For any coupling $(X, Y)$ between between $\mu$ and $\nu$,*

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

*Moreover, there exists an optimal coupling reaching the equality.*

Given a finite state space $\Omega$, a discrete-time *Markov chain* is a sequence $\{X_t\}_{t \ge 0}$ 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 $\Omega$ 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 $\pi$, if

$$\pi(X)\boldsymbol{P}(X, Y) = \pi(Y)\boldsymbol{P}(Y, X) \qquad \forall X, Y \in \Omega.$$

This equation is usually known as the *detailed balance condition.*

A distribution $\pi$ is *stationary* for $\boldsymbol{P}$, if $\pi\boldsymbol{P} = \pi$ (regarding $\pi$ 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 $\pi$. The speed of convergence towards $\pi$ 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 $(X_t, Y_t)_{t \ge 0}$ is a *coupling of Markov chain* $\boldsymbol{P}$ if $(X_t)_{t \ge 0}$ and $(Y_t)_{t \ge 0}$ individually follow the transition rule of $\boldsymbol{P}$, and if $X_i = Y_i$ then $X_j = Y_j$ for all $j \ge i$. By the coupling lemma, for any coupling $(X_t, Y_t)_{t \ge 0}$ of $\boldsymbol{P}$, it holds that

$$d_{\mathrm{TV}}(P^t(X_0, \cdot), P^t(Y_0, \cdot)) \le \Pr[X_t \neq Y_t].$$

Hence, the mixing time of $\boldsymbol{P}$ can be bounded by

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

### 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 [HSS11].

**Theorem 5** ([EL75, HSS11]). *If there exists a function $x : \mathcal{B} \to (0, 1)$ such that for any bad event $B \in \mathcal{B}$,*

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

*then it holds that*

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

*Moreover, for any event $A$,*

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

### 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 $\Delta$. 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 $\mu$ denote the uniform distribution of all proper list hypergraph colourings. The following local uniformity property holds for the distribution $\mu$. Its proof follows from the argument in [GLLZ19]. We include it here for completeness.

**Lemma 6** (local uniformity [GLLZ19]). *Let $q_0 = \min_{v \in V} |Q_v|$ and $q_1 = \max_{v \in V} |Q_v|$. For any $r \ge k \ge 2$, if $q_0^k \ge \mathrm{e} q_1 r \Delta$, the for any $v \in V$ and $c \in Q_v$,*

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

*where $\mu_v$ is the marginal distribution on $v$ induced by $\mu$.*

*Proof.* Let $\mathcal{D}$ denote the product distribution where each $v \in V$ samples a colour in $Q_v$ uniformly at random. For each $e \in \mathcal{E}$, let $B_e$ be the bad event that $e$ is monochromatic. Let $x(e) = \frac{1}{r\Delta}$ for all $e \in \mathcal{E}$. Note that $r \ge k$. We have

$$\Pr_{\mathcal{D}}[B_e] \le \frac{q_1}{q_0^k} \le \frac{1}{\mathrm{e} r \Delta} \le \frac{1}{r\Delta}\left(1 - \frac{1}{r\Delta}\right)^{k(\Delta-1)} \le x(B_e) \prod_{B \in \Gamma(B_e)} (1 - x(B)).$$

By Theorem 5, it holds that

$$\mu_v(c) \le \frac{1}{|Q_v|}\left(1 - \frac{1}{r\Delta}\right)^{-\Delta} \le \frac{1}{|Q_v|} \exp\left(\frac{2}{r}\right).$$

For the lower bound, consider each hyperedge $e$ such that $v \in e$. Let $\mathsf{Block}_e$ be the event that all vertices in $e$ except $v$ have the colour $c$. If none of $\mathsf{Block}_e$ occurs, then $v$ has colour $c$ with probability at least $1/|Q_v|$. By Theorem 5, we have

$$\mu_v(c) \ge \frac{1}{|Q_v|} \Pr_\mu\left[\bigwedge_{e \ni v} \overline{\mathsf{Block}_e}\right] \ge \frac{1}{|Q_v|}\left(1 - \sum_{e \ni v} \Pr_\mu[\mathsf{Block}_e]\right).$$

Note that $\Pr_{\mathcal{D}}[\mathsf{Block}_e] \le q_0^{-k+1}$ and $|\Gamma(\mathsf{Block}_e)| \le k(\Delta-1)+1$. We have

$$\Pr_{\mu}[\mathsf{Block}_e] \le q_0^{-k+1}\left(1-\frac{1}{r\Delta}\right)^{-k(\Delta-1)-1} \le q_0^{-k+1}\mathrm{e} \le \frac{1}{r\Delta},$$

where the last inequality holds because $q_0^{-k+1}\mathrm{e} \le q_0^{-k}q_1\mathrm{e} \le \frac{1}{r\Delta}$, which implies

$$\mu_v(c) \ge \frac{1}{|Q_v|}\left(1-\sum_{e\ni v}\Pr_{\mu}[\mathsf{Block}_e]\right) \ge \frac{1}{|Q_v|}\left(1-\frac{1}{r}\right) \ge \frac{1}{|Q_v|}\exp\left(-\frac{2}{r}\right). \qquad \square$$

# 3 Algorithm

Let $H = (V, \mathcal{E})$ be a $k$-uniform hypergraph and $[q]$ a set of colours. Let $\mu$ denote the uniform distribution of proper hypergraph colourings. Our algorithm is a variant of the projected dynamics from [FGYZ21], using a particular projection scheme from [FHY20]. 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 [FHY20].

**Definition 7** (projection scheme [FHY20])**.** *Let $1 \le s \le q$ 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 $\nu$ 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 $\mu^{\sigma_\Lambda}$ is the distribution of $\boldsymbol{X} \sim \mu$ conditional on $h(\boldsymbol{X}_\Lambda) = \sigma_\Lambda$.*

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

## 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 $s = \left\lceil \sqrt{q} \right\rceil$. Given a projection scheme $h$ with image size $s$, our sampling algorithm first samples $Y \in [s]^V$ from the projected distribution $\nu$, and then uses it to sample a random hypergraph colouring from the conditional distribution $\mu^Y$. The pseudocode is given in Algorithm 1.

---

**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 $X \in [q]^V$

1 sample $Y \in [s]^V$ uniformly at random;

2 **for** $t$ *from* 1 *to* $T = \lceil 50 n \log \frac{2n\Delta}{\epsilon} \rceil$ **do**

3     let $v$ be the vertex with label $(t \mod n)$;

4     $X'_v \leftarrow \mathsf{Sample}\left(H, h, \{v\}, Y_{V \setminus \{v\}}, \frac{\epsilon}{4T}\right)$;

    `/* The Sample subroutine is given in Algorithm 2. */`

5     $Y_v \leftarrow h(X'_v)$;

6 **return** $X \leftarrow \mathsf{Sample}\left(H, h, V, Y, \frac{\epsilon}{4T}\right)$;

---

The main ingredient of Algorithm 1 is the part that samples $Y$ (Line 1 to Line 5). It is basically a *systematic scan* version of the Glauber dynamics for $\nu$. 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 $Y_{V \setminus \{v\}}$. Also, Sample is used to generate the random colouring conditional on $Y$ 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 $Y_\Lambda \in [s]^\Lambda$. Note that during the execution of Algorithm 1, $Y_\Lambda$ is a random input to Sample. Let $S \subseteq V$ and $\zeta \in (0, 1)$. The subroutine $\mathsf{Sample}\,(H, h, S, Y_\Lambda, \zeta)$ in Algorithm 1 returns a random sample $X_S \in [q]^S$ such that with probability at least $1 - \zeta$, the total variation distance between $X_S$ and $\mu_S^{Y_\Lambda}$ is at most $\zeta$, where the probability is taken over the randomness of the input $Y_\Lambda$.

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 $\nu_v^{Y_{V\setminus\{v\}}}$, where $\nu$ is the distribution projected from $\mu$. 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 $\nu_v^{Y_{V\setminus\{v\}}}$ as long as $X'_v$ approximately follows the law of $\mu_v^{Y_{V\setminus\{v\}}}$. In the last step, we use Sample to draw approximate samples from the conditional distribution $\mu^Y$.

We explain the details of $\mathsf{Sample}\,(H, h, S, Y_\Lambda, \zeta)$ next. First we need some notations. Given a partial image $Y_\Lambda$, we say an hyperedge $e \in \mathcal{E}$ is satisfied by $Y_\Lambda$ if there exists $u, v \in e \cap \Lambda$ such that $Y_u \neq Y_v$. In other words, for all $X \in [q]^V$ such that $Y_\Lambda = h(X_\Lambda)$, the hyperedge $e$ is not monochromatic with respect to $X$, and thus $e$ is always "satisfied" given $Y_\Lambda$. Let $H^{Y_\Lambda} = (V, \mathcal{E}^{Y_\Lambda})$ be the hypergraph obtained from $H$ by removing all hyperedges satisfied by $Y_\Lambda$. Let $H_1^{Y_\Lambda}, H_2^{Y_\Lambda}, \ldots, H_m^{Y_\Lambda}$ denote the connected components of $H^{Y_\Lambda}$, where $H_i^{Y_\Lambda} = (V_i, \mathcal{E}_i^{Y_\Lambda})$. The following fact is straightforward to verify

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

where $\mu_i$ is the uniform distribution over proper $q$-colourings of the sub-hypergraph $H_i^{Y_\Lambda}$ (namely, $\mu_i^{Y_{\Lambda \cap V_i}}$ is the uniform distribution over list colourings of $H_i^{Y_\Lambda}$ conditional on $Y_{\Lambda \cap V_i}$). Without loss of generality, we assume $S \cap V_j \neq \emptyset$ for $1 \leq j \leq \ell$. To draw a random sample from $\mu_S^{Y_\Lambda}$, it suffices to draw a random sample from the product distribution $\mu_1^{Y_{\Lambda \cap V_1}} \times \mu_2^{Y_{\Lambda \cap V_2}} \times \ldots \times \mu_\ell^{Y_{\Lambda \cap V_\ell}}$, which we will do by drawing from each $\mu_i^{Y_{\Lambda \cap V_i}}$ 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 $1 \le i \le \ell$, $|\mathcal{E}_i^{Y_\Lambda}| > 4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)$;
- for some $1 \le i \le \ell$, the rejection sampling for $\mu_i^{Y_{\Lambda\cap V_i}}$ 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 12), 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 2:** Sample $(H, h, S, Y_\Lambda, \zeta)$

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

**Output:** A random (partial) colouring $X_S \in [q]^S$

remove all hyperedges in $H$ that are satisfied by $Y_\Lambda$ to obtain $H^{Y_\Lambda} = (V, \mathcal{E}^{Y_\Lambda})$;
let $H_i = (V_i, \mathcal{E}_i^{Y_\Lambda})$ for $1 \le i \le \ell$ be the connected components such that $V_i \cap S \ne \emptyset$;
**if** $\exists 1 \le i \le \ell$ *such that* $|\mathcal{E}_i^{Y_\Lambda}| > 4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)$ **then**
  **return** $X_S \in [q]^S$ *uniformly at random;*
**for** $i$ *from 1 to* $\ell$ **do**
  $X_i \leftarrow$ RejectionSampling$(H_i, h, Y_{\Lambda\cap V_i}, R)$, 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** $X_i = \perp$ **then**
    **return** $X_S \in [q]^S$ *uniformly at random ;*
**return** $X_S$ *where* $X = \biguplus_{i=1}^{\ell} X_i$*;*

**Algorithm 3:** RejectionSampling$(H, h, Y_\Lambda, R)$

**Input:** A hypergraph $H = (V, \mathcal{E})$, a projection scheme $h : [q] \to [s]$, a (partial) image $Y_\Lambda \in [s]^\Lambda$ where $\Lambda \subseteq V$ and an integer $R$

**Output:** A random colouring $X \in [q]^V$ 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 $X = (X_v)_{v\in V}$;
  **if** $X$ *is a proper hypergraph colouring of* $H$ **then**
    **return** $X$*;*
**return** $\perp$*;*

# 4 Proof of the main theorem

Let $H = (V, \mathcal{E})$ be a simple $k$-uniform hypergraph with maximum degree $\Delta$. 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 ($q$ mod $s$) 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 $\Delta$, 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 $\epsilon$-close to $\mu$ 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.** *The parameter $\alpha$ captures the relation between the local lemma condition and the running time of the algorithm. If $\alpha$ 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 $\nu$ be the projected distribution. The idealised systematic scan for $\nu$ 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 $X_t(v) \sim \nu_v^{X_{t-1}(V\setminus\{v\})}$.

**Lemma 11.** *If $q \ge 40\Delta^{\frac{2}{k-4}}$ and $k \ge 20$, the systematic scan chain $\boldsymbol{P}_{scan}$ for $\nu$ is irreducible, aperiodic and reversible with respect to $\nu$. Furthermore, the mixing time satisfies*

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

Lemma 11 is shown in Section 7.

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:

- $\mathcal{B}_{\mathrm{com}}(t)$: in the $t$-th invocation, $\boldsymbol{X}_S$ is returned by Line 4 in Algorithm 2;
- $\mathcal{B}_{\mathrm{rej}}(t)$: 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 12.** *For any $1 \le t \le T+1$, the $t$-th invocation of the subroutine* Sample $(H, h, S, Y_\Lambda, \zeta)$, *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 $\mathcal{B}_{\mathrm{com}}(t)$ nor $\mathcal{B}_{\mathrm{rej}}(t)$ occurs, the subroutine returns a perfect sample from $\mu_S^{Y_\Lambda}$;*
3. *if $q \ge 100\Delta^{\frac{2}{k-3}}$ and $k \ge 20$, then $\Pr[\mathcal{B}_{\mathrm{rej}}(t)] \le \zeta$;*

4. *for any $\delta > 0$, if $k \geq \frac{20(\delta+1)}{\delta}$, $q \geq 100\Delta^{\frac{2+\delta}{k-4/\delta-3}}$, and $H$ is simple, then* $\Pr[\mathcal{B}_{\text{com}}(t)] \leq \zeta$.

Lemma 12 is proved in Section 5 and 6.

Now we are ready to prove our main result, Theorem 10.

*Proof of Theorem 10.* First note that the condition in Theorem 10 implies all the conditions in Lemma 11 and Lemma 12. Denote the output of Algorithm 1 by $X_{\mathsf{alg}}$. To prove the correctness of our algorithm, the goal is to show

$$d_{\mathrm{TV}}(X_{\mathsf{alg}}, \mu) \leq \epsilon.$$

We first consider an idealized algorithm which, instead of simulating the transitions by the Sample subroutine, is able to run the ideal Glauber dynamics to obtain $Y_{\mathsf{ideal}}$ before sampling $X_{\mathsf{ideal}}$ from the distribution $\mu^{Y_{\mathsf{ideal}}}$. By Lemma 11, running this systematic scan for $T = \lceil 50n \log \frac{2n\Delta}{\epsilon} \rceil$ steps ensures $d_{\mathrm{TV}}(Y_{\mathsf{ideal}}, \nu) \leq \frac{\epsilon}{2}$. On the other hand, a perfect sample $X \sim \mu$ can be drawn by sampling $Y \sim \nu$ first, followed by sampling $X \sim \mu^Y$ based on that. The upper bound on total variation distance allows us to couple the perfect $Y$ and $Y_{\mathsf{ideal}}$ such that $Y \neq Y_{\mathsf{ideal}}$ with probability no more than $\frac{\epsilon}{2}$. Conditional on $Y = Y_{\mathsf{ideal}}$, the samples $X$ and $X_{\mathsf{ideal}}$ on original distribution can be perfectly coupled. Together with the coupling lemma (Lemma 4), we have

$$d_{\mathrm{TV}}(X_{\mathsf{ideal}}, \mu) \leq \frac{\epsilon}{2}.$$

Hereinafter, we couple the idealized algorithm with Algorithm 1. The nature of systematic scan warrants that both algorithms pick the same vertex in the same step on Line 3. We then try to couple the vertex update as much as possible. That is, at Step $t$, if none of $\mathcal{B}_{\text{com}}(t)$ or $\mathcal{B}_{\text{rej}}(t)$ happens, then the output of Sample subroutine at Line 4 in Algorithm 1 is perfect, and hence we can couple it with the idealized systematic scan perfectly. The remaining coupling error emerges from the occurrence of $\mathcal{B}_{\text{com}}(t)$ or $\mathcal{B}_{\text{rej}}(t)$. By the coupling lemma (Lemma 4) and Lemma 12, we have

$$d_{\mathrm{TV}}(X_{\mathsf{alg}}, X_{\mathsf{ideal}}) \leq \Pr\left[\bigvee_{i=1}^{T} \left(\mathcal{B}_{\text{com}}(t) \vee \mathcal{B}_{\text{rej}}(t)\right)\right] = 2T\zeta = \frac{\epsilon}{2}$$

where the last equality is due to the selection of $\zeta$ in Algorithm 1. Finally, a straightforward application of triangle inequality yields

$$d_{\mathrm{TV}}(X_{\mathsf{alg}}, \mu) \leq d_{\mathrm{TV}}(X_{\mathsf{alg}}, X_{\mathsf{ideal}}) + d_{\mathrm{TV}}(X_{\mathsf{ideal}}, \mu) = \epsilon$$

as desired.

There are $T + 1$ invocations to the Sample subroutine in total, with the first $T$ calls each costing

$$T_{\mathsf{step}} := O\left(\Delta^2 k^5 \left(\frac{n\Delta}{\epsilon/4T}\right)^{\frac{1}{1000\eta}} \log^3\left(\frac{n\Delta q}{\epsilon/4T}\right)\right)$$

and the final call on Line 6 costing

$$T_{\mathsf{final}} := O\left(n\Delta^2 k^5 \left(\frac{n\Delta}{\epsilon/4T}\right)^{\frac{1}{1000\eta}} \log^3\left(\frac{n\Delta q}{\epsilon/4T}\right)\right).$$

Summing up, the total running time is

$$T_{\mathsf{total}} = T \cdot T_{\mathsf{step}} + T_{\mathsf{final}} = O\left((T+n)\Delta^2 k^5 \left(\frac{n\Delta}{\epsilon/4T}\right)^{\frac{1}{1000\eta}} \log^3\left(\frac{n\Delta q}{\epsilon/4T}\right)\right) \tag{6}$$

where

$$T = 50n \log \frac{2n\Delta}{\epsilon} \qquad \text{and} \qquad \eta = \frac{1}{\Delta}\left(\frac{q}{100}\right)^{\frac{k-3}{2}}. \tag{7}$$

Note that the condition $q \geq 100\left(\frac{\Delta}{\alpha}\right)^{\frac{2+\delta}{k-4/\delta-4}}$ implies

$$\eta = \frac{1}{\Delta}\left(\frac{q}{100}\right)^{\frac{k-3}{2}} \geq \frac{1}{\Delta}\left(\left(\frac{\Delta}{\alpha}\right)^{\frac{2+\delta}{k-4/\delta-4}}\right)^{\frac{k-3}{2}} \geq \frac{1}{\alpha}\Delta^{\frac{(k-3)(1+\delta/2)}{k-4/\delta-4}-1} \geq \frac{1}{\alpha}$$

and hence

$$\left(\frac{n\Delta}{\epsilon/4T}\right)^{\frac{1}{1000\eta}} \leq \left(\frac{200n^2\Delta\log\frac{2n\Delta}{\epsilon}}{\epsilon}\right)^{\alpha/1000} = O\left(\left(\frac{n\Delta}{\epsilon}\right)^{\alpha/100}\right). \tag{8}$$

Plugging (7) and (8) back into (6), we get

$$T_{\mathsf{total}} = O\left(\Delta^2k^5n\left(\frac{n\Delta}{\epsilon}\right)^{\alpha/100}\log^4\left(\frac{n\Delta q}{\epsilon}\right)\right)$$

as desired. $\square$

# 5 Analysis of the Sample subroutine

In this section, we analyse the subroutine Sample and prove Lemma 12. Properties 1, 2, and 3 in Lemma 12 can be proved using techniques developed in [FGYZ21, FHY20]. The proofs are given in Section 5.1 and Section 5.2. We remark that proofs of the first three properties in Lemma 12 hold for general hypergraphs, not necessarily simple hypergraphs. It is property 4 that requires a simple hypergraph as the input. The proof of property 4 is quite involved and is left to Section 6.

## 5.1 Proof of running time and correctness

*Proof of Property 1 and 2, Lemma 12.* Property 2 is straightforwardly implied by the nature of rejection sampling. We now deal with Property 1.

Assume all hypergraphs are stored as incidence lists. We first calculate the time cost of Line 2. Starting from each $v \in S$, we perform depth-first search (DFS) on $H$, and for each edge we encounter, we can check whether it is in $H^{Y_\Lambda}$ in time $O(k)$. This procedure can work simultaneously with Line 3, that once the current component reaches size $4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)$, the subroutine exits in Line 4. The number of visits by DFS itself will be upper-bounded by the number of edges times maximum edge degree which is no larger than $\Delta k$. In all, the time complexity of DFS has a crude upper bound

$$T_{\mathsf{DFS}} = O\left(|S| \cdot k \cdot 4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right) \cdot \Delta k\right) = O\left(|S|\Delta^2k^5\log\left(\frac{n\Delta}{\zeta}\right)\right).$$

For the time cost of Line 6, be aware $\ell$ is at most $|S|$. Suppose the cost of sampling a uniformly random colour from a colour list $Q \subseteq [q]$ is $O(\log q)$. Each invocation of RejectionSampling contains $R$ rounds, each of which colours the subgraph $H_i$ and check if it is a proper colouring. The cost depends to the number of vertices in $H_i$, which is upper-bounded by $k \cdot 4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)$. The total cost is then

$$T_{\mathsf{Rej}} = O\left(|S| \cdot R \cdot \Delta k^4 \log\left(\frac{n\Delta}{\zeta}\right)\log q\right) \leq O\left(|S|\Delta k^4\left(\frac{n\Delta}{\zeta}\right)^{\frac{1}{1000\eta}}\log^3\left(\frac{n\Delta q}{\zeta}\right)\right).$$

The total running time of Sample is hence given by

$$T_{\mathsf{Sample}} = T_{\mathsf{DFS}} + T_{\mathsf{Rej}} = O\left(|S|\Delta^2 k^5 \left(\frac{n\Delta}{\zeta}\right)^{\frac{1}{1000\eta}} \log^3\left(\frac{n\Delta q}{\zeta}\right)\right). \qquad \square$$

### 5.2 Bound the probability of $\mathcal{B}_{\mathrm{rej}}(t)$

*Proof of Property 3, Lemma 12.* By the definition of $\eta$ in (4) and the condition in Lemma 12, it holds that

$$q = 100(\eta\Delta)^{\frac{2}{k-3}}, \quad \eta \geq 1, \quad \text{and} \quad q \geq 100.$$

Consider Line 6 in Algorithm 2. In the rejection sampling, the input is a hyperedge $H = (V, \mathcal{E})$ with at most $4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)$ hyperedges. The size of the color list for each vertex $v \in V$ satisfies

$$|Q_v| \geq \left\lfloor \frac{q}{s} \right\rfloor = \left\lfloor \frac{q}{\lceil q \rceil} \right\rfloor \overset{(*)}{\geq} \frac{4}{5}\sqrt{q},$$

where inequality $(*)$ holds because $q \geq 100$.

Let $\mathcal{D}$ denote the product distribution that each $v \in V$ samples a colour from $Q_v$ uniformly at random. For each hyperedge $e \in \mathcal{E}$, let $\mathcal{B}_e$ denote the bad event that $e$ is monochromatic. Note that $|Q_v| \leq q$ for all $v \in V$. We have for any $e \in \mathcal{E}$,

$$\mathrm{Pr}_{\mathcal{D}}[\mathcal{B}_e] \leq \frac{q}{(\frac{4}{5}\sqrt{q})^{k-1}} = \left(\frac{5}{4}\right)^{k-1} q^{\frac{3-k}{2}} = \left(\frac{5}{4}\right)^{k-1} 100^{\frac{3-k}{2}} \frac{1}{\eta\Delta} \leq \frac{1}{10000\mathrm{e}k^3\eta\Delta},$$

where the last inequality holds because $k \geq 20$. For each $e \in \mathcal{E}$, define $x(e) = \frac{1}{10000\eta\Delta k^3}$. Note that $\eta \geq 1$. It is straightforward to verify that

$$\mathrm{Pr}_{\mathcal{D}}[\mathcal{B}_e] \leq x(e) \prod_{e':\mathcal{B}_{e'} \in \Gamma(B_e)} (1 - x(e')).$$

By Lovász local lemma in Theorem 5, it holds that

$$\mathrm{Pr}_{\mathcal{D}}\left[\bigwedge_{e \in \mathcal{E}} \overline{\mathcal{B}(e)}\right] \geq \left(1 - \frac{1}{10000\eta\Delta k^3}\right)^{\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)} \geq \exp\left(-\frac{\log\left(\frac{n\Delta}{\zeta}\right)}{5000\eta}\right) \geq \left(\frac{\zeta}{n\Delta}\right)^{\frac{1}{1000\eta}}.$$

The rejection sampling repeats for $R = \left\lceil 10\left(\frac{n\Delta}{\zeta}\right)^{\frac{1}{1000\eta}} \log\frac{n}{\zeta}\right\rceil$ times. Hence, the probability that the rejection sampling fails on one connected component is at most

$$\left(1 - \left(\frac{\zeta}{n\Delta}\right)^{\frac{1}{1000\eta}}\right)^R \leq \exp\left(-10\log\frac{n}{\zeta}\right) \leq \left(\frac{\zeta}{n}\right)^2.$$

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

$$\Pr[\mathcal{B}_{\mathrm{rej}}(t)] \leq \zeta. \qquad \square$$

# 6 Analysis of connected components

In this section, we prove Property 4 in Lemma 12. We assume that the input hypergraph $H$ is simple in this section. Fix $1 \le t \le T+1$. Consider the $t$-th invocation of the subroutine $\mathsf{Sample}$. 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 ($t \mod n$). Recall that $Y_t \in [s]^V$ is the random configuration generated by Algorithm 1 after the $t$-th iteration of the for-loop. Denote

$$\Lambda = \begin{cases} V \setminus \{v_t\} & \text{if } 1 \le t \le T \\ V & \text{if } t = T+1 \end{cases} \quad \textit{and} \quad Y = Y_{t-1}(\Lambda), \tag{9}$$

so that the input partial configuration to $\mathsf{Sample}$ is $Y$ (see Algorithm 1). Hence, we consider the subroutine $\mathsf{Sample}\,(H, h, S, Y, \zeta)$, where $Y \in [s]^\Lambda$ is a random configuration.

Let $H = (V, \mathcal{E})$ denote the input simple hypergraph. Since $Y \in [s]^\Lambda$ is a random configuration, $H^Y$ is a random hypergraph, where $H^Y$ is obtained by removing all the hyperedges in $H$ satisfied by $Y$. Fix an arbitrary vertex $v \in V$. We use $H_v^Y = (V_v^Y, \mathcal{E}_v^Y)$ to denote the connected component in $H^Y$ that contains the vertex $v$. Note that $\mathcal{E}_v^Y$ 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.

**Lemma 13.** *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*

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

We now show that property 4 is a corollary of Lemma 13. Since there are at most $\Delta$ hyperedges incident to $v$, by a union bound, we have for all $v \in V$,

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

By a union bound over all vertices $v \in V$, we have

$$\Pr_Y\left[\exists v \in V \text{ s.t. } |\mathcal{E}_v^Y| \ge 4\Delta k^3 \log\left(\frac{n\Delta}{\zeta}\right)\right] \le \zeta.$$

This implies the property 4 in Lemma 12. The rest of this section is dedicated to the proof of Lemma 13.

## 6.1 Proof of Lemma 13

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 13 and let $u = u_e$ be the vertex in $L_H$ corresponding to $e$. Let $L_H^Y = (V_L^Y, E_L^Y)$ denote the line graph of $H^Y$. Note that $L_H^Y$ is random, and the randomness of $L_H^Y$ is determined by the randomness of $Y$. Equivalently, the graph $L_H^Y$ can be generated as follows:

- remove all vertices $w \in V_L$ such that the corresponding hyperedges in $H$ are satisfied by $Y$; let $V_L^Y \subseteq V_L$ denote the set of remaining vertices;
- let $L_H^Y = L_H[V_L^Y]$ be the subgraph of $L_H$ induced by $V_L^Y$.

Let $C \subseteq V_L$ denote the random set of all vertices in the connected component of $L_H^Y$ that contains the vertex $u$. If $u \notin V_L^Y$, let $C = \emptyset$. Define an integer parameter $\theta := \lceil \frac{4}{\delta} \rceil$. To prove Lemma 13, it suffices to show that

$$\forall M > \theta, \quad \Pr_Y\left[|C| \ge M\right] \le \left(\frac{1}{2}\right)^{\frac{M}{2\theta k^2 \Delta} - 1}. \tag{10}$$

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 13.

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

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

In our proof, we partition the set $\mathrm{Con}_u(M)$ into two disjoint subsets

$$\mathrm{Con}_u(M) = \mathrm{Con}_u^{(1)}(M) \uplus \mathrm{Con}_u^{(2)}(M),$$

and we bound the probability separately

$$\Pr_Y[|C| \geq M] \leq \Pr_Y\left[\exists C \in \mathrm{Con}_u^{(1)}(M) \text{ s.t. } C \subseteq V_L^Y\right] + \Pr_Y\left[\exists C \in \mathrm{Con}_u^{(2)}(M) \text{ s.t. } C \subseteq V_L^Y\right]. \tag{11}$$

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 $C_1, C_2, \ldots, C_\ell \subseteq C$. Let

$$\forall C \in \mathrm{Con}_u(M), \quad C \in \begin{cases} \mathrm{Con}_u^{(1)}(M) & \text{if } \ell(C) \geq \frac{M}{2\theta k^2 \Delta}; \\ \mathrm{Con}_u^{(2)}(M) & \text{if } \ell(C) < \frac{M}{2\theta k^2 \Delta}. \end{cases} \tag{12}$$

We remark that Algorithm 4 is only used for analysis, and we do not need to implement this algorithm.

**Algorithm 4:** 2-block-tree generator

**Input:** the parameter $\delta \in (0,1)$ in Lemma 13, 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 $\ell$ and connected subgraphs $C_1, \cdots, C_\ell \subseteq C$

1 let $G = L_H[C] = (C, E_C)$ be the subgraph of $L_H$ induced by $C$;
2 $\theta \leftarrow \lceil \frac{4}{\delta} \rceil$, $\ell \leftarrow 0$, $V \leftarrow C$;
3 **while** $|V| \geq \theta$ **do**
4     $\ell \leftarrow \ell + 1$;
5     **if** $\ell = 1$ **then** $u_\ell \leftarrow u$;
6     **if** $\ell > 1$ **then** let $u_\ell$ be an arbitrary vertex in $\Gamma_G(C \setminus V)$;
7     let $C_\ell \subseteq V$ be an arbitrary connected subgraph in $G$ such that $|C_\ell| = \theta$ and $u_\ell \in C_\ell$;
8     $V \leftarrow V \setminus (C_\ell \cup \Gamma_G(C_\ell))$;
9     **for** *each connected component* $G' = (V', E')$ *in* $G[V]$ *such that* $|V'| < \theta$ **do**
10         $V \leftarrow V \setminus V'$;
11 **return** $\ell, \{C_1, C_2, \ldots, C_\ell\}$;

In Line 6 and Line 7 of Algorithm 4, we may use a specific rule to choose the vertex $u_\ell$ and the connected subgraph $C_\ell$ (e.g. pick the element with the smallest index according to an arbitrary but predetermined ordering). To explain this algorithm concretely, consider the first round of the **while**-loop running on the graph in Figure 1, with the parameter $\theta$ set to 3.

In Line 7, the algorithm picks the connected subgraph $C_1$ containing $u$, represented by black circles. Then in Line 8, the algorithm removes $C_1$, together with its neighbours, depicted by circles in dark grey, from the vertex set $V$. Afterwards, the algorithm checks all remaining connected components, and removes those with size less than $\theta = 3$ from $V$ in Line 10. In this example, the algorithm captures and deletes the component in the dotted box. Be aware that their neighbours (dark grey circles) have already been removed from $V$. As the algorithm goes into the second round of the **while**-loop, the next candidate starting point $u_2$ is selected, as of in Line 6, among the vertices depicted by white circles.

To formalize the properties of Algorithm 4, we begin with the following proposition, which asserts that Algorithm 4 is well defined. The proof is given in Section 6.2.

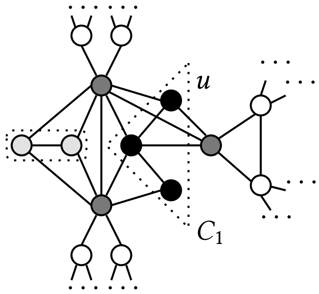


Figure 1: The example graph where Algorithm 4 runs on.

**Proposition 14.** *Given the input $\delta$, $L_H$, $M$, $u$, and $C \in \mathrm{Con}_u(M)$, Algorithm 4 terminates and generates a unique output. Moreover, when Algorithm 4 terminates, $V = \emptyset$.*

The next proposition, yet of more importance, establishes a few properties of the output of Algorithm 4. They will eventually be used to bound the probabilities on the right hand side (RHS) of (11). Before characterising these properties, 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 $\theta$ and tree size $\ell$ in $G$ if*
*(B1) for any $1 \leq i \leq \ell$, $C_i \subseteq V$, $|C_i| = \theta$, and the induced subgraph $G[C_i]$ is connected;*
*(B2) for any distinct $1 \leq i, j \leq \ell$, $\mathrm{dist}_G(C_i, C_j) \geq 2$;*
*(B3) $\{C_1, \cdots, C_\ell\}$ 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 [Alo91] by setting $\theta = 1$. 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 $\theta$ 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 $\theta$, where $C' = C \setminus (\cup_{i=1}^{\ell} \Gamma_G(C_i))$.*

In Proposition 16, Item 1 is stated with respect to the line graph $L_H$, but Item 2 is stated with respect to the induced subgraph $L_H[C]$. The proof of Proposition 16 is also given in Section 6.2.

Finally, to bound the probabilities on the RHS of (11), we need the following lemma about the random configuration $\boldsymbol{Y} \in [s]^\Lambda$. The proof of Lemma 17 is given in Section 6.3.

**Lemma 17.** *If $\lfloor q/s \rfloor^k \geq 2\mathrm{e}qk\Delta$, then for any $R \subseteq \Lambda$, any $\sigma \in [s]^R$, it holds that*

$$\Pr[\boldsymbol{Y}_R = \sigma] \leq \left(\frac{1}{s} + \frac{1}{q}\right)^{|R|} \exp\left(\frac{|R|}{k}\right).$$

The following result is a straightforward corollary of Lemma 17.

**Corollary 18.** *Let $\delta > 0$ and $R_1, R_2, \ldots, R_\ell \subseteq \Lambda$ be disjoint subsets. For each $1 \leq i \leq \ell$, let $\mathcal{S}_i \subseteq [s]^{R_i}$ be a subset of configurations (namely an event). If $k \geq \frac{20(\delta+1)}{\delta}$ and $q \geq 100\Delta^{\frac{2+\delta}{k-4/\delta-3}}$, then it holds that*

$$\Pr\left[\bigwedge_{i=1}^{\ell} (\boldsymbol{Y}_{R_i} \in \mathcal{S}_i)\right] \leq \prod_{i=1}^{\ell} |\mathcal{S}_i| \left(\frac{1}{s} + \frac{1}{q}\right)^{|R_i|} \exp\left(\frac{|R_i|}{k}\right).$$

*Proof.* Let $R = R_1 \uplus R_2 \uplus \ldots \uplus R_\ell$. Note that $\bigwedge_{i=1}^{\ell} \left(Y_{R_i} \in \mathcal{S}_i\right)$ if and only if $Y_R \in \mathcal{S}_1 \otimes \mathcal{S}_2 \otimes \ldots \otimes \mathcal{S}_\ell$, where

$$\mathcal{S}_1 \otimes \mathcal{S}_2 \otimes \ldots \otimes \mathcal{S}_\ell := \left\{\sigma \in [s]^R \mid \forall 1 \le i \le \ell, \sigma_{R_i} \in \mathcal{S}_i\right\}.$$

We now verify the condition in Lemma 17 that $\lfloor q/s \rfloor^k \ge 2eqk\Delta$. Since $s = \left\lceil \sqrt{q} \right\rceil$ and $q \ge 100$, $\lfloor q/s \rfloor \ge \sqrt{q}/4$. Thus it suffices to verify $(\sqrt{q}/4)^k \ge 2eqk\Delta$. The condition in Corollary 18 implies that $q \ge 100\Delta^{\frac{2}{k-2}}$ and $k \ge 20$, which implies $(\sqrt{q}/4)^k \ge 2eqk\Delta$. Hence, the condition in Lemma 17 holds. We have

$$\Pr\left[\bigwedge_{i=1}^{\ell} \left(Y_{R_i} \in \mathcal{S}_i\right)\right] = \sum_{\sigma \in \mathcal{S}_1 \uplus \mathcal{S}_2 \uplus \ldots \uplus \mathcal{S}_\ell} \Pr\left[Y_R = \sigma\right] \le \prod_{i=1}^{\ell} |\mathcal{S}_i| \left(\frac{1}{s} + \frac{1}{q}\right)^{|R_i|} \exp\left(\frac{|R_i|}{k}\right). \qquad \square$$

Now, we are ready to bound the probabilities on the RHS of (11). We handle the two terms separately:

$$\Pr_Y\left[\exists C \in \mathrm{Con}_u^{(1)}(M) \text{ s.t. } C \subseteq V_L^Y\right] < \left(\frac{1}{2}\right)^{\frac{M}{2\theta k^2 \Delta}}; \tag{13}$$

$$\Pr_Y\left[\exists C \in \mathrm{Con}_u^{(2)}(M) \text{ s.t. } C \subseteq V_L^Y\right] < \left(\frac{1}{2}\right)^{M}. \tag{14}$$

Combining (11) with (13) and (14), we have

$$\begin{aligned}\Pr_Y\left[|C| \ge M\right] &\le \Pr_Y\left[\exists C \in \mathrm{Con}_u^{(1)}(M) \text{ s.t. } C \subseteq V_L^Y\right] + \Pr_Y\left[\exists C \in \mathrm{Con}_u^{(2)}(M) \text{ s.t. } C \subseteq V_L^Y\right] \\ &\le \left(\frac{1}{2}\right)^{\frac{M}{2\theta k^2 \Delta}} + \left(\frac{1}{2}\right)^{M} \le \left(\frac{1}{2}\right)^{\frac{M}{2\theta k^2 \Delta} - 1}.\end{aligned}$$

This proves the desired inequality (10).

In the next two subsections, we give proofs of (13) and (14).

### 6.1.1 Proof of inequality (13)

We first prove (13). We need to use the following two properties of 2-block-trees, the proofs of which are deferred till Section 6.4.

**Lemma 19.** *Let $\theta \ge 1$ be an integer. Let $G = (V, E)$ be a graph. For any integer $\ell \ge 2$, any vertex $v \in V$, if $G$ has a 2-block-tree $\{C_1, C_2, \ldots, C_\ell\}$ with block size $\theta$ and tree size $\ell$ such that $v \in \cup_{i=1}^{\ell} C_i$, then there exists an index $1 \le i \le \ell$ such that $\{C_1, C_2, \ldots, C_\ell\} \setminus \{C_i\}$ is a 2-block-tree in $G$ with block size $\theta$ and tree size $\ell - 1$ and $v \in \cup_{1 \le j \le \ell : j \ne i} C_j$.*

**Lemma 20.** *Let $\theta \ge 1$ be an integer. Let $G = (V, E)$ be a graph with maximum degree $d$. For any integer $\ell \ge 1$, any vertex $v \in V$, the number of 2-block-trees $\{C_1, C_2, \ldots, C_\ell\}$ with block size $\theta$ and tree size $\ell$ such that $v \in \cup_{i=1}^{\ell} C_i$ is at most $(\theta \mathrm{e}^{\theta} d^{\theta+1})^{\ell}$.*

In the rest of this subsection we fix $\ell = \left\lceil \frac{M}{2\theta k^2 \Delta} \right\rceil$. By (12), Proposition 16, and Lemma 19, 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 $\theta$ and tree size $\ell$ satisfying:

(P1) $u \in C_1 \cup C_2 \cup \ldots \cup C_\ell$;

(P2) $C_1 \cup C_2 \cup \ldots \cup C_\ell \subseteq C$.

We denote a 2-block-tree tree with block size $\theta$ and tree size $\ell$ by $(\theta,\ell)$-2-block-tree. This implies that

$$\begin{aligned}&\Pr_Y\left[\exists C\in \mathrm{Con}_u^{(1)}(M) \text{ s.t. } C\subseteq V_L^Y\right]\\ &\le \Pr_Y\left[\exists\,(\theta,\ell)\text{-2-block-tree } \{C_1,C_2,\dots,C_\ell\} \text{ in } L_H \text{ satisfying (P1) s.t. } \forall 1\le i\le \ell, C_i\subseteq V_L^Y\right].\end{aligned}\tag{15}$$

Note that we only need to consider $(\theta,\ell)$-2-block trees satisfying (P1), because (P2) implies the event that $\forall 1\le i\le \ell, C_i\subseteq V_L^Y$.

To bound the probability, we fix a $(\theta,\ell)$-2-block tree $\{C_1,C_2,\dots,C_\ell\}$ in $L_H$ satisfying (P1). Fix an index $1\le j\le \ell$. By Definition 15, $|C_j|=\theta$. Note that each vertex in $C_j$ represents a hyperedge in the input hypergraph $H=(V,\mathcal{E})$. Let the hyperedges in $C_j$ be $e_1^j,e_2^j,\dots,e_\theta^j$. For each $1\le t\le\theta$, we define a subset of vertices $R_t^j\subseteq\Lambda$ (in $H$) by

$$S_t^j := e_t^j\setminus\left(\bigcup_{i\in[\theta]:i\neq t} e_i^j\right)\quad\text{and}\quad R_t^j:=S_t^j\cap\Lambda,$$

where $\Lambda$ is defined in (9). By definition, $R_t^j\subseteq e_t^j$ is a subset of vertices of the input hypergraph $H=(V,\mathcal{E})$, and $R_t^j\cap e_i^j=\emptyset$ for any $i\neq t$. This implies that $R_1^j,R_2^j,\dots,R_\theta^j$ are mutually disjoint. Furthermore, since $H$ is simple and $|\Lambda|\ge|V|-1$, we have

$$\forall 1\le t\le\theta:\quad \left|R_t^j\right|\ge k-(\theta-1)-1=k-\theta.\tag{16}$$

The above inequality holds because (1) $|e_t^j|=k$; (2) for each $e_i^j$ with $i\neq t$, the intersection between $e_t^j$ and $e_i^j$ is at most one vertex; and (3) $|\Lambda|\ge|V|-1$. By Definition 15 of 2-block-trees, for $i\neq j$, $\mathrm{dist}_{L_H}(C_i,C_j)\ge 2$. Let $e\in\mathcal{E}$ be a hyperedge in $C_i$ and $e'\in\mathcal{E}$ be a hyperedge in $C_j$, this implies that $e$ and $e'$ are not adjacent in the line graph $L_H$, and thus $e\cap e'=\emptyset$. Hence,

$$(R_t^j)_{1\le j\le\ell,1\le t\le\theta} \text{ are mutually disjoint.}\tag{17}$$

We now bound the probability of $C_j\subseteq V_L^Y$ for all $1\le j\le\ell$. For all $1\le j\le\ell$ and $1\le t\le\theta$, since $C_j\subseteq V_L^Y$, the hyperedge $e_t^j$ is not satisfied by $\boldsymbol{Y}$, thus $e_t^j$ is monochromatic with respect to $\boldsymbol{Y}$, i.e. for all $v,v'\in e_t^j$, it holds that $Y_v=Y_{v'}$. Note that $R_t^j\subseteq e_t^j$. We have the following bound

$$\Pr_Y\left[\forall 1\le j\le\ell, C_j\subseteq V_L^Y\right]\le\Pr_Y\left[\forall 1\le j\le\ell,1\le t\le\theta, R_t^j \text{ is monochromatic w.r.t. } \boldsymbol{Y}\right].\tag{18}$$

Let $\mathcal{S}_t^j$ be the set of all $s$ monochromatic configurations of $R_t^j$ (i.e. all vertices in $R_t^j$ take the same value $c$, where $c\in[s]$), or more formally,

$$\mathcal{S}_t^j=\{\sigma\in\{c\}^{R_t^j}\mid c\in[s]\}.$$

In particular, $\left|\mathcal{S}_t^j\right|=s$. By Corollary 18, (16), (17), and (18), it holds that

$$\begin{aligned}\Pr_Y\left[\forall 1\le i\le\ell, C_i\subseteq V_L^Y\right]&\le\Pr_Y\left[\bigwedge_{j=1}^{\ell}\bigwedge_{t=1}^{\theta}\left(Y_{R_t^j}\in\mathcal{S}_t^j\right)\right]\le\prod_{i=1}^{\ell}\prod_{t=1}^{\theta}s\left(\frac1s+\frac1q\right)^{\left|R_t^j\right|}\exp\left(\frac{|R_t^j|}{k}\right)\\ &\le s^{\ell\theta}\prod_{i=1}^{\ell}\prod_{t=1}^{\theta}\left(\frac1s+\frac1q\right)^{\left|R_t^j\right|}\exp\left(\frac{|R_t^j|}{k}\right)\\ \left(\text{as } k-\theta\le|R_t^j|\le k\right)\quad&\le(\mathrm{e}s)^{\ell\theta}\left(\frac1s+\frac1q\right)^{\ell\theta(k-\theta)}=\left((\mathrm{e}s)^{\theta}\left(\frac1s+\frac1q\right)^{\theta(k-\theta)}\right)^{\ell}.\end{aligned}$$

Note that the maximum degree of $L_H$ is no more than $k\Delta$. By Lemma 20 and a union bound over all possible 2-block-trees, we have

$$\begin{aligned}&\Pr_Y\left[\exists\, (\theta,\ell)\text{-2-block-tree } \{C_1,C_2,\ldots,C_\ell\} \text{in } L_H \text{ satisfying (P1) s.t. } \forall 1\le i\le \ell, C_i\subseteq V_L^Y\right]\\ &\le \left(\theta \mathrm{e}^{2\theta}(k\Delta)^{\theta+1}s^{\theta}\left(\frac{1}{s}+\frac{1}{q}\right)^{\theta(k-\theta)}\right)^{\ell} \le \left(\theta\mathrm{e}^{2\theta}2^{\theta(k-\theta)}(k\Delta)^{\theta+1}s^{\theta-\theta(k-\theta)}\right)^{\ell},\end{aligned} \tag{19}$$

where the last inequality uses the fact that $\frac{1}{s}+\frac{1}{q}\le\frac{2}{s}$. We will show that

$$\theta\mathrm{e}^{2\theta}2^{\theta(k-\theta)}(k\Delta)^{\theta+1}s^{\theta-\theta(k-\theta)}\le\frac{1}{2}. \tag{20}$$

Recall that $k>\theta+1$, and consequently $\theta(k-\theta)-\theta>0$. It implies that

$$\theta\mathrm{e}^{2\theta}2^{\theta(k-\theta)}(k\Delta)^{\theta+1}s^{\theta-\theta(k-\theta)}\le\frac{1}{2}\quad\Longleftrightarrow\quad s\ge\theta^{\frac{1}{\theta(k-\theta)-\theta}}\mathrm{e}^{\frac{2\theta}{\theta(k-\theta)-\theta}}2^{\frac{\theta(k-\theta)+1}{\theta(k-\theta)-\theta}}(k\Delta)^{\frac{\theta+1}{\theta(k-\theta)-\theta}}.$$

Recall that $s=\lceil\sqrt{q}\rceil\ge q^{1/2}$. It suffices to show that

$$q\ge\theta^{\frac{2}{\theta(k-\theta)-\theta}}\mathrm{e}^{\frac{4\theta}{\theta(k-\theta)-\theta}}2^{\frac{2\theta(k-\theta)+2}{\theta(k-\theta)-\theta}}(k\Delta)^{\frac{2\theta+2}{\theta(k-\theta)-\theta}}=\theta^{\frac{2}{\theta(k-\theta)-\theta}}\mathrm{e}^{\frac{4}{k-\theta-1}}2^{\frac{2(k-\theta)+2/\theta}{k-\theta-1}}(k\Delta)^{\frac{2+2/\theta}{k-\theta-1}}.$$

Recall that $\theta=\lceil\frac{4}{\delta}\rceil$. If $\delta\ge 4$, then $\theta=1$. In this case, we only need to show that

$$q\ge\mathrm{e}^{\frac{4}{k-2}}2^{\frac{2k}{k-2}}k^{\frac{4}{k-2}}\Delta^{\frac{2+\delta/2}{k-2}}.$$

Otherwise $0<\delta<4$, in which case we only need to show that

$$q>2\mathrm{e}^{\frac{4}{k-4/\delta-2}}2^{\frac{2k-8/\delta+\delta/2}{k-4/\delta-2}}(k\Delta)^{\frac{2+\delta/2}{k-4/\delta-2}},$$

as $\theta^{\frac{2}{\theta(k-\theta)-\theta}}<2$ and $4/\delta\le\theta<4/\delta+1$. The conditions $k\ge\frac{20(\delta+1)}{\delta}$ and $q\ge 100\Delta^{\frac{2+\delta}{k-4/\delta-3}}$ imply both conditions above. This finishes the proof of (20). Finally, (13) follows from combining (15), (19), and (20).

### 6.1.2 Proof of inequality (14)

We continue to show (14). Fix a connected component $C\in\mathrm{Con}_u^{(2)}(M)$. We analyse the probability of $C\subseteq V_L^Y$. We run Algorithm 4 with the input $C$. The algorithm outputs an integer $\ell<\frac{M}{2\theta k^2\Delta}$ and a set of connected components $C_1,C_2,\ldots,C_\ell$. Let $G=L_H[C]$ be the subgraph of $L_H$ induced by $C$. By Proposition 16, after removing all vertices of $\Gamma_G(C_i)$ for all $1\le i\le\ell$, the graph $G$ is decomposed into connected components with vertex sets $D_1,D_2,\ldots,D_m\subseteq C$ such that $|D_i|\le\theta$ for all $1\le j\le m$. Note that given $C\in\mathrm{Con}_u^{(2)}(M)$, all the sets $D_1,D_2,\ldots,D_m\subseteq C$ are uniquely determined by Algorithm 4. We have

$$\Pr_Y[C\subseteq V_L^Y]\le\Pr_Y\left[\bigwedge_{j=1}^{m}\left(D_j\subseteq V_L^Y\right)\right].$$

We then use an analysis similar to the last subsection but focused on the $D_j$'s. For each $1\le j\le m$, each vertex in $D_j$ represents a hyperedge in the input hypergraph $H=(V,\mathcal{E})$. Let $d(j)=|D_j|$. Let $e_1^j,e_2^j,\ldots,e_{d(j)}^j$ denote the hyperedges in $D_j$. For each $1\le t\le d(j)$, we define

$$S_t^j:=e_t^j\setminus\left(\bigcup_{i\in[d(j)]:i\ne t}e_i^j\right)\quad\text{and}\quad R_t^j:=S_t^j\cap\Lambda.$$

Since $H$ is simple, $|D_j| \le \theta$, and $|\Lambda| \ge |V| - 1$, it holds that

$$\forall 1 \le t \le d(j): \quad \left|R_t^j\right| \ge k - (\theta - 1) - 1 = k - \theta. \tag{21}$$

Next, note that $D_1, D_2, \ldots, D_m \subseteq C$ is a set of disjoint connected components in the induced subgraph $G[D]$, where $D = C \setminus (\cup_{i=1}^{\ell} \Gamma_G(C_i)) = \cup_{i=1}^{m} D_i$. For any two distinct $1 \le i, j \le m$, $\mathrm{dist}_G(D_i, D_j) \ge 2$, as otherwise $D_i$ and $D_j$ must have been merged into one component. As $G = L_H[C]$ is a subgraph of $L_H$ induced by $C$, for any two distinct $1 \le i, j \le m$, $\mathrm{dist}_{L_H}(D_i, D_j) \ge 2$. Hence, for any hyperedge $e \in \mathcal{E}$ in $D_i$, any hyperedge $e' \in \mathcal{E}$ in $D_j$, it holds that $e \cap e' = \emptyset$. It implies that

$$(R_t^j)_{1 \le j \le m, 1 \le t \le d(j)} \text{ are mutually disjoint.} \tag{22}$$

Again, let $\mathcal{S}_t^j$ denote the set of all $s$ monochromatic configurations of $R_t^j$ (i.e. all vertices in $R_t^j$ taking the same value $c$, where $c \in [s]$). By Corollary 18 and (22), it holds that

$$\begin{aligned}\Pr_Y[C \subseteq V_L^Y] &\le \Pr_Y\left[\bigwedge_{j=1}^{m}\left(D_j \subseteq V_L^Y\right)\right] \le \Pr_Y\left[\bigwedge_{j=1}^{m}\bigwedge_{t=1}^{d(j)}\left(R_t^j \subseteq V_L^Y\right)\right] = \Pr_Y\left[\bigwedge_{j=1}^{m}\bigwedge_{t=1}^{d(j)}\left(Y_{R_t^j} \in \mathcal{S}_t^j\right)\right] \\ &\le \prod_{j=1}^{m}\prod_{t=1}^{d(j)}\left(s\left(\frac{1}{s}+\frac{1}{q}\right)^{|R_t^j|}\exp\left(\frac{|R_t^j|}{k}\right)\right) \le \prod_{j=1}^{m}\prod_{t=1}^{d(j)}\left(\mathrm{e}s\left(\frac{1}{s}+\frac{1}{q}\right)^{|R_t^j|}\right),\end{aligned}$$

where the last equation holds because $|R_t^j| \le k$. Define

$$R := \bigcup_{j=1}^{m}\bigcup_{t=1}^{d(j)} R_t^j$$

as the (disjoint) union of all $R_t^j$. By the lower bound in (21), we have

$$|R| \ge \sum_{j=1}^{m}\sum_{t=1}^{d(j)}(k-\theta) = (k-\theta)\sum_{j=1}^{m} d(j) = (k-\theta)\left(M - \left|\bigcup_{i=1}^{\ell}\Gamma_G(C_i)\right|\right),$$

where the last equation holds because $\{D_i\}_{1 \le i \le m}$ is a partition of $C \setminus (\cup_{i=1}^{\ell}\Gamma_G(C_i))$ and $|C| = M$. Note that for any $1 \le i \le \ell$, $|C_i| = \theta$ and the maximum degree of the line graph $L_H$ is at most $k\Delta$. We have

$$|R| \ge (k-\theta)\left(M - \ell\theta k\Delta\right).$$

This implies

$$\Pr_Y[C \subseteq V_L^Y] \le \prod_{j=1}^{m}\prod_{t=1}^{d(j)}\left(\mathrm{e}s\left(\frac{1}{s}+\frac{1}{q}\right)^{|R_t^j|}\right) = (\mathrm{e}s)^{\sum_{i=1}^{m} d(j)}\left(\frac{1}{s}+\frac{1}{q}\right)^{|R|} \le (\mathrm{e}s)^{M}\left(\frac{1}{s}+\frac{1}{q}\right)^{(k-\theta)(M-\ell\theta k\Delta)},$$

where we use the fact $\sum_{i=1}^{m} d(j) \le M$ in the last inequality. Since $C \in \mathrm{Con}_u^{(2)}(M)$, it holds that $\ell < \frac{M}{2\theta k^2\Delta}$. Combining with the fact that $\frac{1}{s} + \frac{1}{q} \le \frac{2}{s}$, we have

$$\Pr_Y[C \subseteq V_L^Y] \le (\mathrm{e}s)^{M}\left(\frac{2}{s}\right)^{(k-\theta)\left(M - \frac{M}{2k}\right)} \le (\mathrm{e}s)^{M}\left(\frac{2}{s}\right)^{(k-\theta)M}\left(\frac{s}{2}\right)^{\frac{M}{2}}.$$

In order to give a rough bound on the number of connected subgraphs containing $u$, we will use the following well-known result by Borgs, Chayes, Kahn, and Lovász [BCKL13].

**Lemma 21** ([BCKL13, Lemma 2.1])**.** *Let $G = (V, E)$ be a graph with maximum degree $d$ and $v \in V$ be a vertex. Then the number of connected induced subgraphs of size $\ell$ containing $v$ is at most $(ed)^{\ell-1}/2$.*

The maximum degree of $L_H$ is at most $k\Delta$. By Lemma 21, the number of connected subgraphs of size $M$ containing $u$ in $L_H$ is at most $(\mathrm{e}\Delta k)^{M-1}/2$. Hence $\left|\mathrm{Con}_u^{(2)}(M)\right| < (\mathrm{e}\Delta k)^M$. By a union bound over all $C \in \mathrm{Con}_u^{(2)}(M)$, we have

$$\Pr_Y\left[\exists C \in \mathrm{Con}_u^{(2)}(M) \text{ s.t. } C \subseteq V_L^Y\right] \le (\mathrm{e}\Delta k)^M (\mathrm{e}s)^M \left(\frac{2}{s}\right)^{(k-\theta)M} \left(\frac{s}{2}\right)^{\frac{M}{2}} = \left(\mathrm{e}^2 s\Delta k \left(\frac{2}{s}\right)^{(k-\theta)}\right)^M \left(\frac{s}{2}\right)^{\frac{M}{2}}.$$

We claim that

$$\mathrm{e}^2 s\Delta k \left(\frac{2}{s}\right)^{(k-\theta)} \le \frac{1}{s}.$$

Since $s = \lceil\sqrt{q}\rceil$, it suffices to show that

$$q \ge \mathrm{e}^{\frac{4}{k-\theta-2}} 2^{\frac{2(k-\theta)}{k-\theta-2}} k^{\frac{2}{k-\theta-2}} \Delta^{\frac{2}{k-\theta-2}},$$

which is, in turn, implied by $\theta = \left\lceil\frac{4}{\delta}\right\rceil$, $k \ge \frac{20(\delta+1)}{\delta}$ and $q \ge 100\Delta^{\frac{2+\delta}{k-4/\delta-3}}$. Hence, we have

$$\Pr_Y\left[\exists C \in \mathrm{Con}_u^{(2)}(M) \text{ s.t. } C \subseteq V_L^Y\right] \le \left(\frac{1}{s}\right)^M \left(\frac{s}{2}\right)^{\frac{M}{2}} \le \left(\frac{1}{2}\right)^M,$$

where the last inequality holds because $s \ge \sqrt{q} \ge 10$.

### 6.2 Properties of the 2-block-tree generator

We begin with validating Algorithm 4, namely proving Proposition 14.

*Proof of Proposition 14.* We claim that the algorithm always succeeds in Line 6 and Line 7, which implies that the size of $V$ strictly decreases in every step and the algorithm halts eventually. Moreover, if $|V| < \theta$, then all vertices in $V$ will be removed in Line 9 and Line 10. Also, so long as $u_\ell$ and $C_\ell$ are selected according to some (arbitrary but) deterministic rule, the output is deterministic.

For the claim, first notice that $V \subseteq C$ throughout the algorithm. For Line 6, since $G = L_H[C]$ is connected and $V \neq \emptyset$, $\Gamma_G(C \setminus V) \neq \emptyset$ and thus $u_\ell$ exists. For Line 7, $C_\ell$ exists as long as the connected component containing $u_\ell$ in $G[V]$ has size at least $\theta$. In the first iteration of the while-loop, this holds true as $|V| = |C| = M > \theta$ and $G[V] = G$ is connected. In all iterations thereafter, the size of the component cannot be smaller than $\theta$, as otherwise it would have been removed in the previous iteration at Line 9 and Line 10. ◻

We then prove Proposition 16. The following observation will be useful.

**Proposition 22.** *Let $\ell > 1$ and $u_\ell$ be the vertex selected in Line 6. Then there exists some $1 \le j < \ell$ such that $\mathrm{dist}_G(C_j, u_\ell) = 2$.*

*Proof.* Assume for contradiction that $\mathrm{dist}_G(C_j, u_\ell) > 2$ for all $1 \le j < \ell$. Consider the set $V$ when $u_\ell$ is selected. Because of Line 6, we can find one of $u_\ell'$s neighbours that is in $C \backslash V$, say $v$. Consider the reason why $v$ was removed from $V$. If this happened on Line 8, then there must have been some $i$ such that $v \in C_i$ or $v \in \Gamma_G(C_i)$. The former case implies that $u_\ell$ must have been removed from $V$, which is impossible. The latter case indicates $\mathrm{dist}_G(C_i, u_\ell) = 2$, a contradiction. Therefore, $v$ was removed in Line 10. However, this implies that $u_\ell$ would have been removed from $V$ too, because $u_\ell$ and $v$ must have been in the same component $V'$, which is also a contradiction. ◻

*Proof of Proposition 16.* The first part of this proposition requires us to verify that $\{C_1, \cdots, C_\ell\}$ is a 2-block-tree in $L_H$. To do so, we verify Items (B1), (B2), and (B3) of Definition 15 next. Notice that what we need to prove here is with respect to $L_H$, instead of $G = L_H[C]$.

- Item (B1) holds due to how $C_i$ is constructed in Line 7.
- For Item (B2), we first show $\mathrm{dist}_G(C_i, C_j) \geq 2$. For any $C_i$ generated by Algorithm 4, it is ensured that $\Gamma_G(C_i)$ gets removed from $V$, and therefore, no vertex in $\Gamma_G(C_i)$ will be in $C_j$ for any other $j$. To show $\mathrm{dist}_{L_H}(C_i, C_j) \geq 2$, note that $G$ is an induced subgraph of $L_H$. Any two vertices of distance more than 1 in $G$ cannot be neighbours in $L_H$, and this implies $\mathrm{dist}_{L_H}(C_i, C_j) \geq 2$.
- To verify (B3), it suffices to show that $\{C_1, \cdots, C_\ell\}$ is connected in $G^2$, because $G$ is a subgraph of $L_H$. This follows from a simple induction. Suppose $\{C_1, \cdots, C_i\}$, in the order of being generated by the algorithm, is connected in $G^2$. The base case of $i = 1$ holds since $C_1$ is connected. Now consider $C_{i+1}$. By Proposition 22, there exists some $j$ such that $\mathrm{dist}_G(C_{i+1}, C_j) = 2$, which implies that $\{C_1, \cdots, C_{i+1}\}$ is connected in $G^2$ as well.

For the second part, suppose towards contradiction that there is some connected component $C^*$ in $G[C']$ of size greater than $\theta$. All vertices in $C$ must have been removed from $V$ when the algorithm halts, according to Proposition 14. However, $C^*$ cannot be $C_i$ for any $i$, because $|C_i| = \theta$. It cannot contain any vertex in $\Gamma_G(C_i)$ either by the definition of $C'$. Thus, no vertex in $C^*$ can be removed in Line 8, and all vertices in $C^*$ must have been removed from $V$ in Line 10. Because $C^*$ does not contain any vertex from either $C_i$ or $\Gamma_G(C_i)$, it does not split into smaller components whilst the algorithm is executed. Thus, the whole $C^*$ must have been removed from $V$ in a single step, which means $|C^*| < \theta$, a contradiction. ◻

### 6.3 Property of random configurations

*Proof of Lemma 17.* Recall that $\boldsymbol{Y} \in [s]^\Lambda$, defined in (9), is the configuration at time $t-1$ on $\Lambda$. For each vertex $w \in V$, let $t(w)$ denote $\max_{1 \leq t' < t}$ such that vertex $w$ is updated by the systematic scan in the $t'$-th step (i.e. the label of $w$ is $t' \mod n$), and let $t(w) = 0$ when such $t'$ does not exist. With this notation $Y_w = Y_{t(w)}(w)$ for all $w \in \Lambda$. We assume $R = \{w_1, w_2, \ldots, w_{|R|}\}$ such that $t(w_1) \leq t(w_2) \leq \ldots \leq t(w_{|R|})$. By the chain rule, we have $\Pr[\boldsymbol{Y}_R = \sigma] = \prod_{i=1}^{|R|} p_i$, where $p_i = \Pr\left[Y_{w_i} = \sigma_{w_i} \mid \bigwedge_{j=1}^{i-1} Y_{w_j} = \sigma_{w_j}\right]$. We now bound the value of each $p_i$ as follows. If $t(w_i) = 0$, then it holds that $p_i \leq \frac{\lceil q/s \rceil}{q}$. If $t(w_i) > 0$, then in the $t(w_i)$-th iteration, the algorithm first samples $X'_{w_i}$ using Sample, and then sets $Y_{w_i} = h(X'_{w_i})$. Denote $\boldsymbol{Y}' = \boldsymbol{Y}_{t(w_i)-1}(V \setminus \{w_i\})$. There are two sub-cases:

- if $X'_{w_i}$ is returned by Line 4 or Line 8 in Sample, then $X'_{w_i}$ is sampled uniformly at random from $[q]$, which implies that $p_i \leq \frac{\lceil q/s \rceil}{q}$;
- if $X'_{w_i}$ is returned by Line 9 in Sample, by property 2 of Lemma 12, $X'_{w_i}$ is sampled from the correct conditional distribution $\mu_{w_i}^{\boldsymbol{Y}'}$. Note that for any $\tau \in [s]^{V \setminus \{w_i\}}$, $\mu_{w_i}^{\tau}$ is the marginal distribution induced by a list hypergraph colouring instance where the colour list of any $w \neq w_i$ is $h^{-1}(\tau(w))$, where $h$ is the projection scheme, and $w_i$'s colour list is $[q]$. By Definition 7 of projection schemes, for any $w \neq w_i$, $\left|h^{-1}(\tau(w))\right| \geq \lfloor q/s \rfloor$. In other words, the upper bound on the size of the lists is $q$ and the lower bound is $\lfloor q/s \rfloor$. Since $\lfloor q/s \rfloor^k \geq 2eqk\Delta$, by Lemma 6, it holds that for all $\tau \in [s]^{V \setminus \{w_i\}}, c \in [q]$,

$$\Pr\left[X'_w = c \mid \boldsymbol{Y}' = \tau \wedge X'_{w_i} \text{ is returned by Line 9}\right] \leq \frac{1}{q} \exp\left(\frac{1}{k}\right),$$

  which implies $p_i \leq \frac{\lceil q/s \rceil}{q} \exp\left(\frac{1}{k}\right)$.

Combining all the cases together, we have

$$\Pr[\boldsymbol{Y}_R = \sigma] \leq \left(\frac{\lceil q/s \rceil}{q}\right)^{|R|} \exp\left(\frac{|R|}{k}\right) \leq \left(\frac{q/s+1}{q}\right)^{|R|} \exp\left(\frac{|R|}{k}\right) = \left(\frac{1}{s} + \frac{1}{q}\right)^{|R|} \exp\left(\frac{|R|}{k}\right).$$ ◻

## 6.4 Properties of 2-block-trees

In this subsection, we show Lemma 19 and Lemma 20. We begin with the first one, which is a simple observation.

*Proof of Lemma 19.* Given a 2-block-tree $\{C_1, \cdots, C_\ell\}$ of $G$ and the vertex $v$, construct the following graph $G_C$. Each vertex $u_j$ of $G_C$ corresponds to a block $C_j$, and two vertices $u_j, u_{j'}$ are adjacent if and only if $\mathrm{dist}_G(C_j, C_{j'}) = 2$. By the definition of 2-block-tree, the graph $G_C$ is connected. Therefore, we can take an arbitrary spanning tree of it. To select the $C_i$ to drop, note that any tree containing at least 2 vertices has at least 2 vertices of degree 1. Therefore, we just choose $u_i$ to be one such vertex where $v \notin C_i$. The rest of the tree is still connected, and so is $G_C - u_i$, which indicates that $\{C_1, \cdots, C_\ell\} - C_i$ still forms a 2-block-tree that contains $v$. □

We proceed to show Lemma 20. We may apply Lemma 21 on $G^2$ due to property (B3). Unfortunately, this yields roughly $(ed^2)^{\theta\ell}$ and does not suffice for our purpose. Here, we give a refined estimation inspired by the original embedding argument of [Sta99, BCKL13].

Let $d' := (ed)^{\theta-1}/2$, which, by Lemma 21, upper bounds the number of size-$\theta$ connected induced subgraphs containing a given vertex in a graph with maximum degree $d$. Therefore, given $v$, we can encode each connected induced subgraph containing $v$ with a positive integer $\Xi \in [d']$. In other words, there exists an injective mapping $\Upsilon_v$ from all connected induced subgraphs of $G$ containing $v$ to $\{v\} \times [d']$.

Our counting argument will be based on encoding the whole 2-block-tree. Intuitively, the encoding contains $\ell + 1$ components. The first one encodes how $C_i$'s are connected in $G^2$, and the rest encodes each individual $C_i$ by an integer in $[d']$.

Let $\mathbb{T}_{\theta d^2}$ to be the infinite $\theta d^2$-ary tree. In the first step, the relation between blocks is encoded by a subtree of $\mathbb{T}_{\theta d^2}$ containing its root, which is basically a DFS tree. However, the order of visiting will affect the DFS tree we construct. For this reason, we need to specify this ordering. First, we order the vertices by their indices. That is, $v_i \prec v_j$ if $i < j$. Given a subset $C$ of vertices, consider the set $\Gamma^2(C)$ containing vertices of distance 2 from $C$. We can sort this set according to the ordering of vertices, and hence any vertex $u \in \Gamma^2(C)$ has a rank among $\Gamma^2(C)$, denoted by $\mathtt{Rank}_C(u)$. Suppose at some stage of our DFS algorithm, we have just finished handling some block $C$. Then we find the next unvisited vertex in $\Gamma^2(C)$, say $v'$, which is in some block $C'$ that needs to be encoded. Then $C'$ will be encoded as the $\mathtt{Rank}_C(v')$-th child of current vertex in the DFS tree, together with the integer $\Upsilon_{v'}(C') \in [d']$. The key of our proof is to show that this encoding is injective, i.e., no two distinct 2-block-trees share the same encoding.

With all the preparation, we give the encoding algorithm as Algorithm 5. Once again, Algorithm 5 is for analysis only and does not need to be implemented.

**Lemma 23.** *Fix a graph $G$ and a vertex $v$. Any 2-block-tree $\{C_1, \cdots, C_\ell\}$ of block size $\theta$ and tree size $\ell$ containing $v$ can be encoded by a tuple $(T, \Xi_1, \cdots, \Xi_\ell)$, where $T$ is a subtree of $\mathbb{T}_{\theta d^2}$ of size $\ell$ containing its root, and $\Xi_i \in [d']$. Moreover, no two distinct 2-block-trees share the same encoding.*

*Proof.* The first part of this lemma follows by going through Algorithm 5. There are two things to verify:

- The algorithm will always halt, outputting $\ell$ $\Xi_i$'s. To show this, one only needs to check that every $C_i$ will be visited exactly once, which is true due to property (B3) of Definition 15 and Line 17 of Algorithm 5.
- The algorithm can find such $w'$ on Line 18, or equivalently, $\mathrm{Rank}_{C_i}(u') \in [\theta d^2]$. This follows after a trivial upper bound on the number of distance-2 neighbours that $|\Gamma^2(C_i)| \le \theta d^2$.

To prove the second part, suppose there are two 2-block-trees $\{C_1, \cdots, C_\ell\}$ and $\{C'_1, \cdots, C'_\ell\}$ with the same encoding $(T, \Xi_1, \cdots, \Xi_\ell)$. Without loss of generality, we can assume $C_1, \cdots, C_\ell$ (resp. $C'_1, \cdots, C'_\ell$)

**Algorithm 5:** Encoding

**Input:** A graph $G$, a vertex $v \in G$, a 2-block-tree $\{C_1, \cdots, C_\ell\}$ of block size $\theta$ and tree size $\ell$

**Output:** An encoding $(T, \Xi_1, \cdots, \Xi_\ell)$, where $T$ is a subtree of $\mathbb{T}_{\theta d^2}$ of size $\ell$

1 initialize `visited[1..`$\ell$`]` to be all `False`;
2 let $C_j$ be the component containing $v$;
3 let $r$ be the root of $\mathbb{T}_{\theta d^2}$;
4 let $T$ be an empty subtree;
5 $t \leftarrow 0$;
6 DFS-Encode($j$,$v$,$r$);
7 **return** $(T, \Xi_1, \cdots, \Xi_\ell)$;

8 **Procedure** *DFS-Encode(i,u,w)*:
9     `visited[i]` $\leftarrow$ **True**;
10     $t \leftarrow t+1$;
11     $\Xi_t \leftarrow \Upsilon_u(C_i)$;
12     add $w$ into $T$;
13     **for** $u' \in \Gamma^2(C_i)$ **do** `// enumerate` $u' \in \Gamma^2(C_i)$ `in order`
14         **if** *there does not exist any* $i'$ *such that* $C_{i'} \ni u'$ **then**
15             **continue**;
16         let $i'$ be the index such that $C_{i'} \ni u'$;
17         **if** `visited[i']=False` **then**
18             let $w'$ be the $\mathtt{Rank}_{C_i}(u')$-th child of $w$ in $\mathbb{T}_{\theta d^2}$;
19             DFS-Encode($i'$,$u'$,$w'$);

are sorted in the order of being visited by Algorithm 5. The goal is then to prove $C_i = C'_i$ for all $i \in [\ell]$. To show this, we do a simple induction argument. More precisely, denote by $T_t$ and $T'_t$ the subtrees constructed by the first $t$ calls to DFS-Encode respectively. We induce on $t$ to show that

$$C_i = C'_i \text{ for all } i \in [t], \text{ and } T_t = T'_t. \tag{IH}$$

**Base case $t = 1$.** Note that $C_1 = C'_1$ follows from the injectivity of $\Upsilon_v$, and $T_1 = T'_1$ as they both contain only the root.

**Induction step.** Suppose (IH) holds for $t-1$. At this stage, we compare the progress of two copies of Encoding running on $C$ and $C'$ respectively. Right before the **for**-loop in the $(t-1)$-th call to DFS-Encode, both copies get the same $w$ by (IH). Again by (IH), both copies get the same $C_{t-1}$ in the condition of the **for**-loop. In the enumeration of **for**-loop, both copies skip or keep the $u'$ in Line 14 simultaneously, because $C_i = C'_i$ for all $i \in [t-1]$. Note that each vertex of $\mathbb{T}_{\theta d^2}$ can be visited at most once. This means that if the two copies get different $u'$ in Line 18, then the final subtree will be different. Therefore, they must get the same $u'$ and $i'$, and hence the same $w'$ because they have the same $C_{t-1}$, implying $T_t = T'_t$. Moreover, the next calls to DFS-Encode have an identical input in both copies. Thus, $\Xi_t = \Upsilon_u(C_t)$ and $\Xi'_t = \Upsilon_u(C'_t)$. By assumption $\Xi_t = \Xi'_t$. Injectivity of $\Upsilon_u$ implies that $C_t = C'_t$, finishing the proof. □

We conclude this subsection by proving Lemma 20.

*Proof of Lemma 20.* By Lemma 23, the number of 2-block-trees can be upperbounded by the number of possible encodings. To count the number of possible subtrees $T$, we simply apply Lemma 21, which gives $(e\theta d^2)^{\ell-1}/2$. The number of possible $\Xi_i$ sequences is $d'^{\ell} = (ed)^{\ell(\theta-1)}/2^{\ell}$. Combining both parts yields the upper bound $\theta^{\ell-1}e^{\theta\ell-1}d^{(\theta+1)\ell-2}/2^{\ell+1}$. □

# 7 Mixing of systematic scan

In this section, we prove the mixing lemma for the projected systematic scan Markov chain of hypergraph colourings (Lemma 11). First, we verify that the systematic scan is irreducible, aperiodic and reversible with respect to $\nu$. This implies that the systematic scan has the unique stationary distribution $\nu$. Aperiodicity and reversibility are straightforward to verify. For irreducibility, it suffices to show that for any $\tau \in [s]^V$, $\nu(\tau) > 0$, as our chain is a Glauber dynamics for $\nu$. Fix an arbitrary configuration $\tau \in [s]^V$. We show that there exists a proper colouring $\sigma \in [q]^V$ such that $h(\sigma) = \tau$, where $h$ is the projection scheme. This implies $\nu(\tau) > 0$. To prove the existence of such a proper colouring, consider the list hypergraph colouring instance $(H, (Q_v)_{v\in V})$, where $Q_v = h^{-1}(\tau_v)$ for all $v \in V$. We only need to show that this list colouring instance has a feasible solution. Note that $|Q_v| \geq \lfloor q/\lceil\sqrt{q}\rceil\rfloor \geq \sqrt{q}/2$ for $q \geq 20$. By the Lovász local lemma, Theorem 5, we only need to verify that

$$eq\left(\frac{2}{\sqrt{q}}\right)^k \Delta k \leq 1,$$

which follows from $q \geq 40\Delta^{\frac{2}{k-4}}$ and $k \geq 20$.

Next, we prove the mixing time result in Lemma 11. The analysis is based on an information percolation argument. We first define a coupling $C$ 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 24.** *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*

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

By Lemma 24, a union bound over all vertices and the coupling lemma (Lemma 4), it holds that

$$\max_{\boldsymbol{X}_0, \boldsymbol{Y}_0 \in [s]^V} d_{\mathrm{TV}}(\boldsymbol{X}_T, \boldsymbol{Y}_T) \leq \Pr_C\left[\boldsymbol{X}_T \neq \boldsymbol{Y}_T\right] \leq \epsilon,$$

which proves the mixing time part of Lemma 11 via (1). In the rest of this section, we use the information percolation technique to analyse the coupling $C$ and prove Lemma 24.

## 7.1 Information percolation analysis

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 25.** *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 $X_{t-1}(u) \neq Y_{t-1}(u)$.*

*Proof.* Note that $X_t(v_t)$ and $Y_t(v_t)$ are sampled from $\nu_{v_t}^{X_{t-1}(V\setminus\{v_t\})}$ and $\nu_{v_t}^{Y_{t-1}(V\setminus\{v_t\})}$ respectively. Let $\mu'$ denote the uniform distribution of proper colourings of $H_t^v$. Let $\pi$ denote the projected distribution induced by $\mu'$ and the projection scheme $h$. Let $V_t^{v_t}$ denote the vertex set of $H_t^{v_t}$ and let $S = V_t^{v_t} \setminus \{v_t\}$. We claim that (1) $\nu_{v_t}^{X_{t-1}(V\setminus\{v_t\})}$ and $\pi_{v_t}^{X_{t-1}(S)}$ are identical distributions; (2) $\nu_{v_t}^{Y_{t-1}(V\setminus\{v_t\})}$ and $\pi_{v_t}^{Y_{t-1}(S)}$ are identical distributions. Hence, if $X_{t-1}(u) = Y_{t-1}(u)$ for all $u \neq v_t$ in $H_t^{v_t}$ , then $X_t(v_t)$ and $Y_t(v_t)$ must be perfectly coupled.

We verify that $\nu_{v_t}^{X_{t-1}(V\setminus\{v_t\})}$ and $\pi_{v_t}^{X_{t-1}(S)}$ are identical distributions. The claim for $\nu_{v_t}^{Y_{t-1}(V\setminus\{v_t\})}$ and $\pi_{v_t}^{Y_{t-1}(S)}$ can be verified by a similar proof. Consider the list colouring instance $(H, (Q_v)_{v\in V})$, where $Q_v = [q]$ if $v = v_t$ and $Q_v = h^{-1}(X_{t-1}(v))$ if $v \neq v_t$. Let $\mu_{\text{list}}$ denote the uniform distribution of all proper list colourings. If $X \sim \mu_{\text{list}}$, then $h(X_{v_t}) \sim \nu_{v_t}^{X_{t-1}(V\setminus\{v_t\})}$. For any hyperedge $e$ satisfied by $A_t$, it holds that for any colouring $X \in \otimes_{v\in V} Q_v$, $e$ is not monochromatic. Let $H_t$ denote the hypergraph obtained from $H$ by removing all hyperedges satisfied by $A_t$. Hence, $(H, (Q_v)_{v\in V})$ and $(H_t, (Q_v)_{v\in V})$ have the same set of proper list colourings. Recall that $H_t^{v_t}$ is the connected component in $H_t$ containing vertex $v_t$. Let $\mu_{\text{list}}^{\text{com}}$ denote the uniform distribution over all proper list colourings of $(H_t^{v_t}, (Q_v)_{v\in V_t^{v_t}})$. Hence, $\mu_{\text{list}}$ projected on $v_t$ is the same distribution as $\mu_{\text{list}}^{\text{com}}$ projected on $v_t$. If $X \sim \mu_{\text{list}}^{\text{com}}$, then $h(X_{v_t}) \sim \pi_{v_t}^{X_{t-1}(S)}$. This implies that $\nu_{v_t}^{X_{t-1}(V\setminus\{v_t\})}$ and $\pi_{v_t}^{X_{t-1}(S)}$ are identical distributions. □

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 25.

**Corollary 26.** *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 $X_{t-1}(w) \neq Y_{t-1}(w)$.*

*Proof.* By Lemma 25, there is a vertex $u \neq v_t$ such that $X_{t-1}(u) \neq Y_{t-1}(u)$ and $u \in H_t^{v_t}$. As $u$ and $v_t$ are in the same connected component, there exist a path from $v_t$ to $u$. Moreover, for each hyperedge $e_i$ on this path, since $e_i$ is in $H_t^{v_t}$, it is not satisfied by $A_t$. This implies that for all $w \neq v_t \in e_i$ such that $X_{t-1}(w) = Y_{t-1}(w)$, their values in both chains must be the same $c \in [s]$. Lastly, note that any path in $H_t^{v_t}$ is also a path in $H$. This proves the corollary. □

Corollary 26 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

$$S(v,t) := \{1 \leq i \leq t \mid v_i = v\},$$

where $v_i$ is the vertex picked in the $i$-th transition step. Define the last update time for $v$ up to $t$ by

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

By Corollary 26, 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 $t' = \mathsf{time}_{\mathsf{ud}}(u,t)$. We apply Corollary 26 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 26. We will show that such a 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 [HSW21].

### 7.2 Extended hyperedges and the extended hypergraph

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

$$\forall u \in e, \quad X_t(u) = X_{t'}(u), \quad \text{where } t' = \mathsf{time}_{\mathsf{ud}}(u, t),$$

namely, the value of $u$ at time $t$ is the same as the value of $u$ at time $t' = \mathsf{time}_{\mathsf{ud}}(u, t)$. Besides, the configuration of hyperedge $e$ remains unchanged until some vertex in $e$ is updated. This motivates the following definition of extended hyperedges and the extended hypergraph, introduced by He, Sun, and Wu [HSW21].

**Definition 27.** *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*

$$\mathcal{E}_0^{\text{ext}} := \bigcup_{e \in \mathcal{E}} \{(0, v) \mid v \in e\},$$
$$\forall 1 \leq t \leq T, \quad \mathcal{E}_t^{\text{ext}} := \bigcup_{e : v_t \in e} \{(t', v) \mid v \in e \wedge t' = \mathsf{time}_{\mathsf{ud}}(v, t)\}.$$

*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 26 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 $H^{\text{ext}}$.

**Lemma 28.** *Let $1 \leq t \leq T$ 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 $H^{\text{ext}}$ such that*
- *$(t, v_t) \in e_1^{\text{ext}}$ and $(t', u) \in e_\ell^{\text{ext}}$;*
- *for any hyperedge $e_i^{\text{ext}}$ 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 $X_j(w) \neq Y_j(w)$.*

*Proof.* Let $u$ and $e_1, e_2, \ldots, e_\ell$ denote the vertex and the path in Corollary 26 respectively. For each vertex $w \in V$, let $t_w = \mathsf{time}_{\mathsf{ud}}(w, t)$. For each $1 \leq i \leq \ell$, define

$$e_i^{\text{ext}} = \{(t_w, w) \mid w \in e_i\}.$$

To show that $e_1^{\text{ext}}, e_2^{\text{ext}}, \ldots, e_\ell^{\text{ext}}$ is a path in $H^{\text{ext}}$, we need to verify that each $e_i^{\text{ext}}$ defined above belongs to $\mathcal{E}^{\text{ext}}$ in Definition 27. Fix an $e_i^{\text{ext}}$. Let $t_{\max} = \max\{t \mid (t, w) \in e_i^{\text{ext}}\}$. It is straightforward to verify that $e_i^{\text{ext}} \in \mathcal{E}_{t_{\max}}^{\text{ext}}$.

Next, we show that $t' < t$ and $X_{t'}(u) \neq Y_{t'}(u)$. By definition, we have $t' = t_u = \mathsf{time}_{\mathsf{ud}}(u, t) < t$. As the value of any vertex does not change until the next update, we have that

$$\forall w \in V \setminus \{v_t\}, \quad X_{t-1}(w) = X_{t_w}(w) \text{ and } Y_{t-1}(w) = Y_{t_w}(w). \tag{23}$$

By Corollary 26, it holds that $X_{t-1}(u) \neq Y_{t-1}(u)$. By (23), it holds that $X_{t'}(u) \neq Y_{t'}(u)$.

Finally, we verify the two properties of the path. The first property $(t, v_t) \in e_1^{\text{ext}}$ and $(t', u) \in e_\ell^{\text{ext}}$ follows from the way $e_i^{\text{ext}}$ is constructed. By Corollary 26, for any $e_i$ in the path, there exists $c \in [s]$ such that for all vertices $w \in e_i \setminus \{v_t\}$, either $X_{t-1}(w) = Y_{t-1}(w) = c$ or $X_{t-1}(w) \neq Y_{t-1}(w)$. By (23), for all extended vertices $(i, w) \in e_i^{\text{ext}}$ with $w \neq v_t$, either $X_i(w) = Y_i(w) = c$ or $X_i(w) \neq Y_i(w)$. Finally, consider the extended vertex $(t, v_t)$. By our assumption in the lemma, we have that $X_t(v_t) \neq Y_t(v_t)$. □

We may repeatedly apply Lemma 28 to trace a discrepancy from some time $t$ to time 0.

**Lemma 29.** *Let $1 \le t \le T$ be an integer. Suppose $X_t(v_t) \neq Y_t(v_t)$. There exists a path $e_1^{\text{ext}}, e_2^{\text{ext}}, \dots, e_\ell^{\text{ext}}$ in the extended hypergraph $H^{\text{ext}}$ 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* $e_i^{\text{ext}}$ *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* $X_j(w) \neq Y_j(w)$.

*Proof.* We use Lemma 28 recursively. Namely, we use Lemma 28 for $(t, v_t)$ to find $(t', u)$. If $t' \neq 0$, we apply Lemma 28 on $(t', u)$ again to find the previous discrepancy. Repeat this process until we find $(t'', w)$ such that $t'' = 0$. This gives a path $f_1^{\text{ext}}, f_2^{\text{ext}}, \dots, f_m^{\text{ext}}$ in the extended hypergraph $H^{\text{ext}}$ such that $(t, v_t) \in f_1^{\text{ext}}$ and $\min\{j \mid (j, w) \in f_m^{\text{ext}}\} = 0$. By Lemma 28, this path $f_1^{\text{ext}}, f_2^{\text{ext}}, \dots, f_m^{\text{ext}}$ satisfies the last property in Lemma 29.

We then construct the path $e_1^{\text{ext}}, e_2^{\text{ext}}, \dots, e_\ell^{\text{ext}}$. First let $e_1^{\text{ext}} = f_1^{\text{ext}}$, $\ell = 1$, and $p = 1$. While $\min\{i \mid (i, w) \in e_\ell^{\text{ext}}\} > 0$, we repeat the following process:

- let $p + 1 \le j \le m$ be the largest index satisfying $f_j^{\text{ext}} \cap e_\ell^{\text{ext}} \neq \emptyset$;
- let $\ell \leftarrow \ell + 1$, $e_\ell^{\text{ext}} \leftarrow f_j^{\text{ext}}$ and $p \leftarrow j$.

When the above process ends, we get the path $e_1^{\text{ext}}, e_2^{\text{ext}}, \dots, e_\ell^{\text{ext}}$.

We first show that the process above is well-defined. Consider the beginning of each iteration of the while-loop. It holds that $e_\ell^{\text{ext}} = f_p^{\text{ext}}$. Since $\min\{i \mid (i, w) \in e_\ell^{\text{ext}}\} > 0$, we know that $p < m$. The index $p + 1 \le j \le m$ such that $f_j^{\text{ext}} \cap e_\ell^{\text{ext}} \neq \emptyset$ must exist because $f_{p+1}^{\text{ext}} \cap e_\ell^{\text{ext}} = f_{p+1}^{\text{ext}} \cap f_p^{\text{ext}} \neq \emptyset$. The while-loop must terminate eventually because $p$ always increase and $\min\{i \mid (i, w) \in f_m^{\text{ext}}\} = 0$.

We claim that $e_1^{\text{ext}}, e_2^{\text{ext}}, \dots, e_\ell^{\text{ext}}$ is indeed a path. We only need to show that for all $2 \le i \le \ell$, it holds that $e_{i-1}^{\text{ext}} \cap e_i^{\text{ext}} \neq \emptyset$ and $e_{i-1}^{\text{ext}} \neq e_i^{\text{ext}}$. The construction process guarantees that $e_{i-1}^{\text{ext}} \cap e_i^{\text{ext}} \neq \emptyset$. Suppose there is an index $2 \le i \le \ell$ such that $e_{i-1}^{\text{ext}} = e_i^{\text{ext}} = f_{i'}^{\text{ext}}$ for some $i' \le m$. Since the construction process finds $e_i^{\text{ext}}$, we know that $\min\{t \mid (t, w) \in e_{i-1}^{\text{ext}}\} > 0$. Thus $i' < m$ and $f_{i'+1}^{\text{ext}}$ exists. Since $f_1^{\text{ext}}, f_2^{\text{ext}}, \dots, f_m^{\text{ext}}$ is a path, we know that $f_{i'}^{\text{ext}} \cap f_{i'+1}^{\text{ext}} \neq \emptyset$, which implies that $e_{i-1}^{\text{ext}} \cap f_{i'+1}^{\text{ext}} \neq \emptyset$. When constructing $e_i^{\text{ext}}$, we look for the largest $j$ such that $e_{i-1}^{\text{ext}} \cap f_j^{\text{ext}} \neq \emptyset$. Hence, $e_i^{\text{ext}} \neq f_{i'}^{\text{ext}}$, a contradiction.

Lastly, we verify the properties of the path.

- Since $e_1^{\text{ext}} = f_1^{\text{ext}}$ and $(t, v_t) \in f_1^{\text{ext}}$, $(t, v_t) \in e_1^{\text{ext}}$. The while-loop terminates once $\min\{j \mid (j, w) \in e_\ell^{\text{ext}}\} > 0$. Hence, $\min\{j \mid (j, w) \in e_i^{\text{ext}}\} > 0$ for all $i < \ell$ and $\min\{j \mid (j, w) \in e_\ell^{\text{ext}}\} = 0$.
- For any $1 \le i, i' \le \ell$ with $|i - i'| \ge 2$, consider how $e_{i+1}^{\text{ext}}$ is constructed. We choose the largest index $j \le m$ such that $f_j^{\text{ext}} \cap e_\ell^{\text{ext}} \neq \emptyset$ and $e_{i+1}^{\text{ext}} \leftarrow f_j^{\text{ext}}$. In other words, for any $j' > j$, $f_{j'}^{\text{ext}} \cap e_i^{\text{ext}} = \emptyset$. Since there is $j'$ such that $e_{i'}^{\text{ext}} = f_{j'}^{\text{ext}}$, $e_i^{\text{ext}} \cap e_{i'}^{\text{ext}} = \emptyset$.
- Since $e_1^{\text{ext}}, e_2^{\text{ext}}, \dots, e_\ell^{\text{ext}}$ is a subsequence of $f_1^{\text{ext}}, f_2^{\text{ext}}, \dots, f_m^{\text{ext}}$, the last property is satisfied as well.

□

### 7.3 Proof of Lemma 24

Recall that $T = \lceil 50n \log \frac{n}{\epsilon} \rceil$ in Lemma 24. To prove Lemma 24, we need to show that

$$\forall v \in V, \quad \Pr_C [X_T(v) \neq Y_T(v)] \le \frac{\epsilon}{n}.$$

Fix a vertex $v$. By the same reason as (23), we only need to prove $\Pr_C[X_T(v) \neq Y_T(v)] \leq \frac{\epsilon}{n}$ for a new $T$, where

$$T = \mathsf{time}_{\mathsf{ud}}\left(v, \left\lceil 50n \log \frac{n}{\epsilon} \right\rceil\right) \geq \left\lceil 40n \log \frac{n}{\epsilon} \right\rceil. \tag{24}$$

Note that $v$ is updated at time $T$, i.e. $v = v_T$.

Fix $T$ defined in (24). Define the following information percolation path (IPP).

**Definition 30.** *We say a path $e_1^{\text{ext}}, e_2^{\text{ext}}, \dots, e_\ell^{\text{ext}}$ of length $\ell$ in the extended hypergraph $H^{\text{ext}}$ is an* information percolation path *(IPP) if the following two properties are satisfied:*

- *$(T, v_T) \in e_1^{\text{ext}}$, $\min\{j \mid (j, w) \in e_i^{\text{ext}}\} > 0$ for all $i < \ell$ and $\min\{j \mid (j, w) \in e_\ell^{\text{ext}}\} = 0$;*
- *for any $1 \leq i, j \leq \ell$ such that $|i - j| \geq 2$, $e_i^{\text{ext}} \cap e_j^{\text{ext}} = \emptyset$.*

Suppose $X_T(v) \neq Y_T(v)$. By Lemma 29, we can find an IPP $e_1^{\text{ext}}, e_2^{\text{ext}}, \dots, e_\ell^{\text{ext}}$ in extended hypergraph $H^{\text{ext}}$. The following lemma lower bounds the length of the IPP.

**Lemma 31.** *For any IPP of length $\ell$, $\ell \geq \lceil T/n \rceil$.*

*Proof.* For any extended hyperedge $e_i^{\text{ext}}$, define the maximum and minimum update times in $e_i^{\text{ext}}$ by $t_{\max}^{(i)} = \max\{t \mid (t, w) \in e_i^{\text{ext}}\}$ and $t_{\min}^{(i)} = \min\{t \mid (t, w) \in e_i^{\text{ext}}\}$. In the systematic scan, we update vertices in order of their labels. By Definition 27, it holds that for any $i$,

$$t_{\max}^{(i)} - t_{\min}^{(i)} \leq n - 1 \leq n.$$

Note that $e_i^{\text{ext}} \cap e_{i+1}^{\text{ext}} \neq \emptyset$, which implies

$$t_{\min}^{(i)} \leq t_{\max}^{(i+1)} \leq t_{\min}^{(i+1)} + n.$$

Note that $t_{\min}^{(1)} \geq t_{\max}^{(1)} - n = T - n$. We have

$$T - n \leq t_{\min}^{(1)} \leq t_{\min}^{(\ell)} + (\ell - 1)n = (\ell - 1)n,$$

where the last equation holds because $t_{\min}^{(\ell)} = 0$. Since $\ell$ is an integer, we have $\ell \geq \lceil T/n \rceil$. □

Now fix an integer $\ell \geq T/n$ and an IPP $\mathcal{P} = e_1^{\text{ext}}, e_2^{\text{ext}}, \dots, e_\ell^{\text{ext}}$ of length $\ell$. We define the bad event $\mathcal{B}(\mathcal{P})$ as: for any hyperedge $e_i^{\text{ext}}$ 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 $X_j(w) \neq Y_j(w)$. Namely, $\mathcal{B}(\mathcal{P})$ that implies $\mathcal{P}$ satisfies the third property in Lemma 29. By Lemma 29, Lemma 31 and a union bound over all IPPs of length at least $\ell$, the probability of $X_T(v) \neq Y_T(v)$ can be bounded as follows

$$\Pr_C[X_T(v) \neq Y_T(v)] \leq \sum_{\ell \geq \lceil T/n \rceil} \sum_{\mathcal{P}\text{: IPP of length } \ell} \Pr_C[\mathcal{B}(\mathcal{P})]. \tag{25}$$

We bound $\Pr_C[\mathcal{B}(\mathcal{P})]$ in the RHS of (25) next. We need to use more delicate structures of the extended hypergraph $H^{\text{ext}} = (V^{\text{ext}}, \mathcal{E}^{\text{ext}})$. By Definition 27, each extended hyperedge $e^{\text{ext}} \in \mathcal{E}^{\text{ext}}$ corresponds to a unique hyperedge $\mathsf{edge}\left(e^{\text{ext}}\right) \in \mathcal{E}$ in the input hypergraph, or more formally,

$$\mathsf{edge}\left(e^{\text{ext}}\right) := \{v \mid (t, v) \in e^{\text{ext}}\}.$$

We remark that different extended hyperedges may correspond to the same hyperedge. For each extended hyperedge $e^{\text{ext}} \in \mathcal{E}^{\text{ext}}$, we use $N(e^{\text{ext}})$ to denote the neighbour extended hyperedges:

$$N(e^{\text{ext}}) := \{f^{\text{ext}} \in \mathcal{E}^{\text{ext}} \mid f^{\text{ext}} \cap e^{\text{ext}} \neq \emptyset \text{ and } f^{\text{ext}} \neq e^{\text{ext}}\}.$$

The following observation is straightforward to verify.

**Observation 32.** *For any $e^{\text{ext}} \in \mathcal{E}^{\text{ext}}$ and $f^{\text{ext}} \in N(e^{\text{ext}})$, $\mathsf{edge}\left(e^{\text{ext}}\right) \cap \mathsf{edge}\left(f^{\text{ext}}\right) \neq \emptyset$.*

We further partition $N(e^{\text{ext}})$ into self-neighbours and outside-neighbours as follows,

$$N_{\mathsf{self}}(e^{\text{ext}}) := \left\{f^{\text{ext}} \in N(e^{\text{ext}}) \mid \mathsf{edge}\left(e^{\text{ext}}\right) = \mathsf{edge}\left(f^{\text{ext}}\right)\right\};$$
$$N_{\mathsf{out}}(e^{\text{ext}}) := \left\{f^{\text{ext}} \in N(e^{\text{ext}}) \mid \mathsf{edge}\left(e^{\text{ext}}\right) \neq \mathsf{edge}\left(f^{\text{ext}}\right)\right\}.$$

**Observation 33.** *For any $e^{\text{ext}} \in \mathcal{E}^{\text{ext}}$ and $f^{\text{ext}} \in N_{\mathsf{out}}(e^{\text{ext}})$, $\left|e^{\text{ext}} \cap f^{\text{ext}}\right| = 1$.*

*Proof.* Let $e = \mathsf{edge}\left(e^{\text{ext}}\right)$ and $f = \mathsf{edge}\left(f^{\text{ext}}\right)$. Since $f^{\text{ext}} \in N_{\mathsf{out}}(e^{\text{ext}})$, by Observation 32 and the fact that the input hypergraph is simple, $|e \cap f| = 1$, which implies $\left|e^{\text{ext}} \cap f^{\text{ext}}\right| = 1$. □

The following lemma bounds the degree of the extended hypergraph.

**Lemma 34.** *Let $\Delta$ be the maximum degree of the input hypergraph $H = (V, \mathcal{E})$. Then,*

1. *given $(t, v) \in V^{\text{ext}}$ and $e \in \mathcal{E}$ such that $v \in e$, the number of $e^{\text{ext}}$ such that $(t, v) \in e^{\text{ext}}$ and $\mathsf{edge}\left(e^{\text{ext}}\right) = e$ is at most $k$;*
2. *for any extended vertex $(t, v) \in V^{\text{ext}}$, the number of extended hyperedges incident to $(v, t)$ is at most $d_{\mathsf{vtx}} := \Delta k$;*
3. *for any extended hyperedge $e^{\text{ext}} \in \mathcal{E}^{\text{ext}}$, $N_{\mathsf{self}}(e^{\text{ext}}) \le d_{\mathsf{self}} := 2k$, $N_{\mathsf{out}}(e^{\text{ext}}) \le d_{\mathsf{out}} := \Delta k^2$.*

*Proof.* For Item 1, suppose such $e^{\text{ext}}$ is $\{(t_j, u_j) \mid 1 \le j \le k\}$ and $t_1 \le t_2 \le \ldots \le t_k$. Moreover, for all $j$ such that $t_j = 0$, we order $u_j$ according to their original label in $H$. As $(v, t) \in e^{\text{ext}}$, $t$ equals one of $t_j$. Then observe that $e^{\text{ext}}$ is uniquely determined if we know $t = t_j$ for some $1 \le j \le k$, and there are at most $k$ choices of $j$ (the number of choices can be less than $k$ if $t = 0$). This shows the claim.

For Item 2, if $e^{\text{ext}}$ is incident to $(v, t)$, then $\mathsf{edge}\left(e^{\text{ext}}\right) = e$ for some $e \ni v$. There are at most $\Delta$ choices of such hyperedge $e$ in $H$. Then the bound follows from Item 2.

For Item 3, let $e = \mathsf{edge}\left(e^{\text{ext}}\right)$, and again assume $e^{\text{ext}}$ is $\{(t_j, u_j) \mid 1 \le j \le k\}$ and $t_1 \le t_2 \le \ldots \le t_k$ as in the proof of Item 1.

To bound the number of self-neighbours, suppose $f^{\text{ext}} \in N_{\mathsf{self}}(e^{\text{ext}})$ such that $\mathsf{edge}\left(f^{\text{ext}}\right) = e$. Let $t_{\max} = \max\{t \mid (t, w) \in f^{\text{ext}}\}$ and $t_{\min} = \min\{t \mid (t, w) \in f^{\text{ext}}\}$. Note that if $t_{\max} \le t_k$, then there are at most $k - 1$ choices of $t_{\max}$, namely $t_1, t_2, \ldots, t_{k-1}$. Otherwise $t_{\max} > t_k$. Note that if $t_{\max} \ge t_k + n$, then $t_{\min} \ge t_{\max} - (n-1) > t_k$, which contradicts to $e^{\text{ext}} \cap f^{\text{ext}} \neq \emptyset$. It must hold that $t_k + 1 \le t_{\max} \le t_k + n - 1$. In the interval $[t_k + 1, t_k + n - 1]$, there are at most $k - 1$ times so that one of the vertices in $e$ is updated (this vertex cannot be $t_k$ as its update times are $t_k$ and $t_k + n$). Thus, there are $k - 1$ choices of $t_{\max}$ again. Once $t_{\max}$ is fixed, since $\mathsf{edge}\left(f^{\text{ext}}\right) = e$, $f^{\text{ext}}$ is also fixed. Overall, the number of $f^{\text{ext}} \in N_{\mathsf{self}}(e^{\text{ext}})$ is at most $2(k-1) \le 2k$.

To bound the number of outside-neighbours. We first choose one of the $k$ extended vertices in $e^{\text{ext}}$, say $(t_i, u_i)$. Then consider $f^{\text{ext}} \in N_{\mathsf{out}}(e^{\text{ext}})$ such that $(t_i, u_i) \in f^{\text{ext}}$. By Item 2, the number of such $f^{\text{ext}}$ is at most $\Delta k$, implying the overall bound of $\Delta k^2$. □

Consider the IPP $\mathcal{P} = e_1^{\text{ext}}, e_2^{\text{ext}}, \ldots, e_\ell^{\text{ext}}$. Define the parameters $R_{\mathsf{self}}$ and $R_{\mathsf{out}}$ by

$$R_{\mathsf{self}} := \left|\left\{2 \le i \le \ell \mid e_i^{\text{ext}} \in N_{\mathsf{self}}(e_{i-1}^{\text{ext}})\right\}\right|;$$
$$R_{\mathsf{out}} := \left|\left\{2 \le i \le \ell \mid e_i^{\text{ext}} \in N_{\mathsf{out}}(e_{i-1}^{\text{ext}})\right\}\right|.$$

By definition, $R_{\mathsf{self}}$ counts the number of consecutive self neighbours in $\mathcal{P}$ and $R_{\mathsf{out}}$ counts the number of consecutive outside neighbours in $\mathcal{P}$. It holds that $R_{\mathsf{self}} + R_{\mathsf{out}} = \ell - 1$. We have the following lemma.

**Lemma 35.** *Suppose $k \ge 20$ and $q \ge 40\Delta^{\frac{2}{k-4}}$. For any IPP $\mathcal{P} = e_1^{\text{ext}}, e_2^{\text{ext}}, \ldots, e_\ell^{\text{ext}}$, it holds that*

$$\Pr_C\left[\mathcal{B}(\mathcal{P})\right] \le 10^3 \Delta k^6 \left(\frac{1}{10^3 \Delta k^6}\right)^{R_{\mathsf{out}} + \frac{1}{3}(R_{\mathsf{self}} - b)},$$

*where $b$ is an integer satisfying $0 \le b \le \min\{R_{\mathsf{self}}, 2R_{\mathsf{out}}\}$.*

The proof of Lemma 35 is given in Section 7.4, where we will specify the value of the integer $b$. Now, we use Lemma 35 to prove Lemma 24. We remark that in the proof of Lemma 24, we do not use the specific value of $b$, we only use the fact that $0 \le b \le \min\{R_{\mathrm{self}}, 2R_{\mathrm{out}}\}$.

*Proof of Lemma 24.* First fix an integer $\ell \ge \lfloor T/n \rfloor$ and an integer $0 \le r \le \ell - 1$. Consider the IPP $\mathcal{P}$ of length $\ell$ such that $R_{\mathrm{out}} = r$ and $R_{\mathrm{self}} = \ell - 1 - r$. By the definition of IPP (Definition 30) together with Lemma 34, the number of such path $\mathcal{P}$ is at most

$$\binom{\ell-1}{r} d_{\mathrm{vtx}} d_{\mathrm{out}}^{r} d_{\mathrm{self}}^{\ell-1-r} \le \Delta k \binom{\ell-1}{r} \left(\Delta k^2\right)^r (2k)^{\ell-1-r}.$$

By Lemma 35 and the union bound in (25), we have

$$\begin{aligned}\Pr_C\left[X_T(v) \ne Y_T(v)\right] &\le \sum_{\ell \ge \lceil T/n \rceil} \sum_{\mathcal{P}:\text{ IPP of length } \ell} \Pr_C\left[\mathcal{B}(\mathcal{P})\right] \\ &\le \sum_{\ell \ge \lceil T/n \rceil} \sum_{r=0}^{\ell-1} \Delta k \binom{\ell-1}{r} \left(\Delta k^2\right)^r (2k)^{\ell-1-r} \cdot 10^3 \Delta k^6 \left(\frac{1}{10^3 \Delta k^6}\right)^{r + \frac{1}{3}(\ell-1-r-b(\ell,r))},\end{aligned}$$

where $b(\ell, r)$ is an integer satisfying $0 \le b(\ell, r) \le \min\{\ell-1-r, 2r\}$. Since $b(\ell, r) \le \ell - 1 - r$, it holds that $\left(\frac{1}{10^3\Delta k^6}\right)^{\frac{\ell-1-r-b(\ell,r)}{3}} \le \left(\frac{1}{10^3 k^6}\right)^{\frac{\ell-1-r-b(\ell,r)}{3}}$, which implies

$$\begin{aligned}\Pr_C\left[X_T(v) \ne Y_T(v)\right] &\le \sum_{\ell \ge \lceil T/n \rceil} \sum_{r=0}^{\ell-1} \Delta k \binom{\ell-1}{r} \left(\Delta k^2\right)^r (2k)^{\ell-1-r} \cdot 10^3 \Delta k^6 \left(\frac{1}{10^3 \Delta k^6}\right)^r \left(\frac{1}{10^3 k^6}\right)^{\frac{\ell-1-r-b(\ell,r)}{3}} \\ &= 10^3 \Delta^2 k^7 \sum_{\ell \ge \lceil T/n \rceil} \sum_{r=0}^{\ell-1} \binom{\ell-1}{r} \left(\frac{1}{5k}\right)^{\ell-1-r} \left(\frac{1}{10^3 k^4}\right)^r \left(\frac{1}{10k^2}\right)^{-b(\ell,r)}.\end{aligned}$$

Note that $k \ge 20$. Since $0 \le b(\ell, r) \le 2r$, we have $\left(\frac{1}{10k^2}\right)^{-b(\ell,r)} \le \left(\frac{1}{10k^2}\right)^{-2r} = (100k^4)^r$, which imples

$$\begin{aligned}\Pr_C\left[X_T(v) \ne Y_T(v)\right] &\le 10^3 \Delta^2 k^7 \sum_{\ell \ge \lceil T/n \rceil} \left(\frac{1}{10}\right)^{\ell-1} \sum_{r=0}^{\ell-1} \binom{\ell-1}{r} = 10^3 \Delta^2 k^7 \sum_{\ell \ge \lceil T/n \rceil} \left(\frac{1}{5}\right)^{\ell-1} \\ &\le 10^3 \Delta^2 k^7 \left(\frac{1}{2}\right)^{T/n}.\end{aligned}$$

Note that $T \ge 40n \log \frac{n\Delta}{\epsilon}$ and $k \le n$. We have

$$\Pr_C\left[X_T(v) \ne Y_T(v)\right] \le \frac{\epsilon}{n}.$$

□

## 7.4 Proof of Lemma 35

Fix an IPP $\mathcal{P} = e_1^{\mathrm{ext}}, e_2^{\mathrm{ext}}, \ldots, e_\ell^{\mathrm{ext}}$. We define a total ordering among all extended hyperedges in $\mathcal{P}$. For any two extended hyperedges $e_i^{\mathrm{ext}}$ and $e_j^{\mathrm{ext}}$ in $\mathcal{P}$, we say $e_i^{\mathrm{ext}} < e_j^{\mathrm{ext}}$ if and only if $i < j$.

**Lemma 36.** *There exists a subsequence $f_1^{\mathrm{ext}} < f_2^{\mathrm{ext}} < \ldots < f_m^{\mathrm{ext}}$ in IPP $\mathcal{P}$ such that*

- *for any $1 \le i, j \le m$ satisfying $|i - j| \ge 2$, $f_i^{\mathrm{ext}} \cap f_j^{\mathrm{ext}} = \emptyset$;*
- *for any $2 \le i \le m$, $\left|f_i^{\mathrm{ext}} \cap f_{i-1}^{\mathrm{ext}}\right| \le 1$;*
- *$m \ge R_{\mathrm{out}} + \frac{1}{3}\left(R_{\mathrm{self}} - b\right)$ for some integer $0 \le b \le \min\{R_{\mathrm{self}}, 2R_{\mathrm{out}}\}$.*

Note that $\{f_i^{\text{ext}}\}$ given in Lemma 36 is not necessarily a path. What we do in Lemma 36 is to prune certain self-neighbours from $\mathcal{P}$ so that the second property holds. To be more precise, for a maximal sequence of consecutive self-neighbouring hyperedges, we prune all hyperedges that are in even positions of this sequence. We give a formal proof below.

*Proof of Lemma 36.* There are $\ell - 1$ pairs of adjacent extended hyperedges, i.e. $e_{i-1}^{\text{ext}}$ and $e_i^{\text{ext}}$ are adjacent for $2 \le i \le \ell$. Define

$$S_{\text{out}} := \left\{ \text{integer } i \in [2, \ell] \mid e_i^{\text{ext}} \in N_{\text{out}}(e_{i-1}^{\text{ext}}) \right\}.$$

Note that $|S_{\text{out}}| = R_{\text{out}}$. Denote $R = R_{\text{out}}$. Suppose the elements in $S_{\text{out}}$ are $2 \le i_1 < i_2 < \ldots < i_R \le \ell$. In addition, we define $i_0 = 1$ and $i_{R+1} = \ell + 1$, although $i_0 \notin S_{\text{out}}$ and $i_{R+1} \notin S_{\text{out}}$. Removing all the elements in $S_{\text{out}}$, the integers in the interval $[2, \ell]$ splits into a set $I_{\text{self}}$ of sub-intervals:

$$I_{\text{self}} := \{[l, r] \mid \exists j \text{ s.t. } 0 \le j \le R,\ l = i_j + 1,\ r = i_{j+1} - 1, \text{ and } l \le r\}.$$

Equivalently, $I_{\text{self}}$ can be constructed by going through all $j$ from $0$ to $R$, and adding the interval $[i_j + 1, i_{j+1} - 1]$ to the set $I_{\text{self}}$ if $i_j + 1 \le i_{j+1} - 1$. For each interval $[l, r] \in I_{\text{self}}$, the following properties hold

1. for each integer $i \in [l, r]$, $e_i^{\text{ext}} \in N_{\text{self}}\left(e_{i-1}^{\text{ext}}\right)$;
2. either $l = 2$ or $e_{l-1}^{\text{ext}} \in N_{\text{out}}\left(e_{l-2}^{\text{ext}}\right)$;
3. either $r = \ell$ or $e_{r+1}^{\text{ext}} \in N_{\text{out}}\left(e_r^{\text{ext}}\right)$.

In other words, each interval $[l, r] \in I_{\text{self}}$ represents a sequence of consecutive extended hyperedges in the IPP $\mathcal{P}$ of length $r - l + 1$ such that each extended hyperedge is a self-neighbour of its predecessor in $\mathcal{P}$, and this sequence is maximal.

Suppose the intervals in $I_{\text{self}}$ are $[l_1, r_1], [l_2, r_2], \ldots, [l_a, r_a]$ such that $l_1 \le r_1 < l_2 \le r_2 < \ldots < l_a \le r_a$, where $a = |I_{\text{self}}|$. It is straightforward to verify that

$$\sum_{i=1}^{a} (r_i - l_i + 1) = R_{\text{self}}. \tag{26}$$

Define a subset $I_{\text{self}}^{(1)} \subseteq I_{\text{self}}$ by

$$I_{\text{self}}^{(1)} := \{[l, r] \in I_{\text{self}} \mid l = r\}.$$

The quantity $b$ is the size of $I_{\text{self}}^{(1)}$, i.e. $b := \left|I_{\text{self}}^{(1)}\right|$. Since $I_{\text{self}}^{(1)}$ is a subset of $I_{\text{self}}$, by (26), we have

$$b \le R_{\text{self}}. \tag{27}$$

Note that $\ell \ge T/n \ge 40 \log n \ge 20$. If $R_{\text{out}} = 0$, then $I_{\text{self}}$ contains only a single interval $[2, \ell]$. Thus $b = 0$ and we have $b \le 2R_{\text{out}}$. Otherwise $R_{\text{out}} \ge 1$. By property 3 above, for each $j \in [a]$, it holds that either $r_j = \ell$ or $e_{r_j+1}^{\text{ext}} \in N_{\text{out}}\left(e_{r_j}^{\text{ext}}\right)$ (namely $r_j + 1 \in S_{\text{out}}$). This implies $b \le R + 1 = R_{\text{out}} + 1 \le 2R_{\text{out}}$, because there are at most one $(l_j, r_j) \in I_{\text{self}}^{(1)}$ satisfying $l_j = r_j = \ell$. Hence, in both cases, we have

$$b \le 2R_{\text{out}}. \tag{28}$$

Combining (27) and (28) proves that $b \le \min\{R_{\text{self}}, 2R_{\text{out}}\}$.

Finally, we construct the the subsequence $f_1^{\text{ext}} < f_2^{\text{ext}} < \ldots < f_m^{\text{ext}}$ from IPP $\mathcal{P}$. We construct a subset $\mathcal{F}$ by the following procedure.

- For each $i \in S_{\text{out}}$, we add $e_i^{\text{ext}}$ into $\mathcal{F}$.

- For each interval $[l, r] \in I_{\text{self}}$, for all integers $j \in [l, r]$ such that $(j - l)$ is an odd number, we add $e^{\text{ext}}_j$ into $\mathcal{F}$. Note that by property 2, if $l > 2$, $e^{\text{ext}}_{l-1}$ is always in $\mathcal{F}$ because of the previous rule.
- To finish, we sort all extended hyperedges in $\mathcal{F}$ to obtain $f^{\text{ext}}_1 < f^{\text{ext}}_2 < \ldots < f^{\text{ext}}_m$.

We now verify the three properties in Lemma 36.

- By the definition of IPP, for any $1 \leq i, j \leq \ell$ satisfying $|i - j| \geq 2$, $e^{\text{ext}}_i \cap e^{\text{ext}}_j = \emptyset$. Since $f^{\text{ext}}_1 < f^{\text{ext}}_2 < \ldots < f^{\text{ext}}_m$ is a subsequence of $\mathcal{P}$, the first property holds.
- Fix an index $2 \leq j \leq m$. Suppose $f^{\text{ext}}_{j-1} = e^{\text{ext}}_{j_1}$ and $f^{\text{ext}}_j = e^{\text{ext}}_{j_2}$. If $|j_1 - j_2| \geq 2$, then $\left|f^{\text{ext}}_i \cap f^{\text{ext}}_{i-1}\right| = 0$. Assume $j_1 + 1 = j_2$, which means that $e^{\text{ext}}_{j_1}$ and $e^{\text{ext}}_{j_2}$ are neighbours in extended hypergraph. If $e^{\text{ext}}_{j_2} \in N_{\text{out}}(e^{\text{ext}}_{j_1})$, by Observation 33, it holds that $\left|f^{\text{ext}}_i \cap f^{\text{ext}}_{i-1}\right| = 1$. Otherwise, $e^{\text{ext}}_{j_2} \in N_{\text{self}}(e^{\text{ext}}_{j_1})$. There must exist an interval $[l, r] \in I_{\text{self}}$ such that either $j_1, j_2 \in [l, r]$ or $j_1 \notin [l, r]$ but $j_2 \in [l, r]$. The first case is impossible because we do not add two consecutive indices in any interval of $I_{\text{self}}$. The second case is also impossible because it implies $j_1 = l - 1$ and $j_2 = l$, but $l$ cannot be added.
- All extendeds hyperedge in $S_{\text{out}}$ are added into $\mathcal{F}$. For each interval $[l, r] \in I_{\text{self}}$, $\lfloor \frac{r-l+1}{2} \rfloor$ extended hyperedges in $[l, r]$ are added into $\mathcal{F}$. Hence, if $l \neq r$, the number of vertices in $[l, r]$ added to $\mathcal{F}$ is at least $(r-l+1)/3$ (with $r = l+2$ being the worst case). By (26), we have $m \geq R_{\text{out}} + \frac{1}{3}(R_{\text{self}} - b)$.

Hence, the subsequence $f^{\text{ext}}_1 < f^{\text{ext}}_2 < \ldots < f^{\text{ext}}_m$ satisfies all the properties in Lemma 36. □

Now we are ready to prove Lemma 35.

*Proof of Lemma 35.* Let $f^{\text{ext}}_1 < f^{\text{ext}}_2 < \ldots < f^{\text{ext}}_m$ be the subsequence given in Lemma 36. For each $f^{\text{ext}}_i$ and $c \in [s]$, define a bad event $\mathcal{B}_i(c)$ that for all $(j, w) \in f^{\text{ext}}_i$, either $X_j(w) \neq Y_j(w)$ or $X_j(w) = Y_j(w) = c$. Note that $f^{\text{ext}}_1 < f^{\text{ext}}_2 < \ldots < f^{\text{ext}}_m$ is a subsequence in IPP $\mathcal{P}$, the probability of $\mathcal{B}(\mathcal{P})$ can be bounded as follows

$$\Pr_C[\mathcal{B}(\mathcal{P})] \leq \Pr_C[\forall i \in [m], \exists c_i \in [s] \text{ s.t. } \mathcal{B}_i(c_i)].$$

By (24), it holds that $\ell \geq T/n \geq 40 \log n \geq 20$. By the last property in Lemma 36, $m \geq \frac{1}{3}(R_{\text{out}} + R_{\text{self}}) = \frac{\ell-1}{3} > 6$. We further truncate the last element $f^{\text{ext}}_m$ and obtain the following inequality

$$\Pr_C[\mathcal{B}(\mathcal{P})] \leq \Pr_C[\forall i \in [m-1], \exists c_i \in [s] \text{ s.t. } \mathcal{B}_i(c_i)] \leq \sum_{\boldsymbol{c} \in [s]^{m-1}} \Pr_C\left[\bigwedge_{i=1}^{m-1} \mathcal{B}_i(c_i)\right], \tag{29}$$

where the second inequality follows from the union bound, and $\boldsymbol{c} = (c_1, \ldots, c_{m-1}) \in [s]^{m-1}$. The truncation ensures that all elements $(j, w) \in \cup_{i=1}^{m-1} f^{\text{ext}}_i$ satisfy $j > 0$. (See Definition 30 of IPPs.)

Fix $\boldsymbol{c} \in [s]^{m-1}$, we bound the probability of the event $\bigwedge_{i=1}^{m-1} \mathcal{B}_i(c_i)$. For each $1 \leq i < m$, we define

$$S^{\text{ext}}_i := \begin{cases} f^{\text{ext}}_i & \text{if } i = 1; \\ f^{\text{ext}}_i \setminus f^{\text{ext}}_{i-1} & \text{if } i > 1. \end{cases}$$

Since $S^{\text{ext}}_i \subseteq f^{\text{ext}}_i$, we have the following bound

$$\Pr_C\left[\bigwedge_{i=1}^{m-1} \mathcal{B}_i(c_i)\right] \leq \Pr_C\left[\bigwedge_{i=1}^{m-1} \left(\forall (j, w) \in S^{\text{ext}}_i, \left(X_j(w) \neq Y_j(w)\right) \vee \left(X_j(w) = Y_j(w) = c_i\right)\right)\right].$$

By the first property in Lemma 36, all $S^{\text{ext}}_i$ are mutually disjoint. Now we list all the extended vertices $\cup_{i=1}^{m-1} S^{\text{ext}}_i$ as $(j_1, w_1), (j_2, w_2), \ldots, (j_M, w_M)$, where $0 < j_1 < j_2 < \ldots < j_M$. For each $1 \leq p \leq M$, there is a unique $i$ such that $(j_p, w_p) \in S^{\text{ext}}_i$ and we denote $\mathsf{idx}(j_p) := i$. We define a bad event $\mathcal{A}(p)$ that either $X_{j_p}(w_p) \neq Y_{j_p}(w_p)$ or $X_{j_p}(w_p) = Y_{j_p}(w_p) = c_{\mathsf{idx}(j_p)}$. Using the chain rule for the RHS of the inequality above, it holds that

$$\Pr_C\left[\bigwedge_{i=1}^{m-1} \mathcal{B}_i(c_i)\right] \leq \prod_{p=1}^{M} \Pr_C\left[\mathcal{A}(p) \mid \bigwedge_{p' < p} \mathcal{A}(p')\right].$$

Consider the probability of $\mathcal{A}(p)$ conditional on all $\mathcal{A}(p')$ for $p' < p$. To simplify the notation, let $j = j_p > 0$ and $w = w_p$. In the $j$-th update, $X_j(w)$ is sampled from the distribution $\nu_w^{X_{j-1}(V\setminus\{w\})}$ and $Y_j(w)$ is sampled from the distribution $\nu_w^{Y_{j-1}(V\setminus\{w\})}$. For any $\tau \in [s]^{V\setminus\{w\}}$, it holds that

$$\forall x \in [s], \quad \nu_w^\tau(x) = \sum_{y \in h^{-1}(x)} \mu_w^\tau(y).$$

Note that $\mu^\tau$ is actually the uniform distribution over a list colouring instance on $H$ where for each $u \neq w$, the colour list is $h^{-1}(\tau_u)$, and the colour list for $w$ is $[q]$. Hence, for each $u \neq w$, the size of colour list of $u$ is at least $\lfloor q/s \rfloor$, and the size of colour list of $w$ is $q$, where $s = \lceil \sqrt{q} \rceil$. Note that $q \geq 40\Delta^{\frac{2}{k-4}}$ and $k \geq 20$ implies $\lfloor q/s \rfloor^k \geq 2\mathrm{e}q^2k\Delta$. By Lemma 6, for all $\tau \in [s]^{V\setminus\{w\}}$, it holds that

$$\forall y \in [q], \quad \frac{1}{q}\left(1 - \frac{4}{kq}\right) \leq \frac{1}{q}\exp\left(-\frac{2}{kq}\right) \leq \mu_w^\tau(y) \leq \frac{1}{q}\exp\left(\frac{2}{kq}\right) \leq \frac{1}{q}\left(1 + \frac{4}{kq}\right).$$

Hence, for any $\tau \in [s]^{V\setminus\{w\}}$, it holds that for any $x \in [s]$,

$$\frac{\left|h^{-1}(x)\right|}{q}\left(1 - \frac{4}{kq}\right) \leq \nu_w^\tau(x) \leq \frac{\left|h^{-1}(x)\right|}{q}\left(1 + \frac{4}{kq}\right).$$

Note that all the events $\mathcal{A}(p')$ for $p' < p$ are determined by the updates from time 1 to time $j-1$. The above bounds for $\nu_w^\tau(x)$ holds for any configuration $\tau \in [s]^{V\setminus\{w\}}$. In the $j$-th update step, since $X_j(w)$ and $Y_j(w)$ are coupled by the optimal coupling and $\left|h^{-1}(x)\right| \leq \lceil q/s \rceil$, we have the probability of $X_j(w) \neq Y_j(w)$ is at most $\frac{1}{2}\sum_{x\in[s]} \frac{\left|h^{-1}(x)\right|}{q} \cdot \frac{8}{kq} = \frac{4}{kq}$, and the probability of $X_j(w) = Y_j(w) = c_i$ is at most $\frac{\lceil q/s \rceil}{q}\left(1 + \frac{4}{kq}\right)$. Hence,

$$\begin{aligned}\Pr_C\left[\mathcal{A}(p) \mid \bigwedge_{p'<p} \mathcal{A}(p')\right] &\leq \frac{4}{kq} + \frac{\lceil q/s \rceil}{q}\left(1 + \frac{4}{kq}\right) \overset{(\star)}{\leq} \frac{\lceil q/s \rceil}{q}\left(1 + \frac{5}{k}\right)\\ &\leq \frac{1.16}{\sqrt{q}}\left(1 + \frac{5}{k}\right).\end{aligned}$$

where $(\star)$ holds because $\frac{\lceil q/s \rceil}{kq} \geq \frac{4}{kq}$ if $q \geq 40$ and the last inequality is due to $\lceil q/s \rceil \leq 1.16\sqrt{q}$. This implies

$$\Pr_C\left[\bigwedge_{i=1}^{m-1} \mathcal{B}_i(c_i)\right] \leq \prod_{p=1}^{M}\left(\frac{1.16}{\sqrt{q}}\left(1 + \frac{5}{k}\right)\right) = \prod_{i=1}^{m-1}\left(\frac{1.16}{\sqrt{q}}\left(1 + \frac{5}{k}\right)\right)^{\left|S_i^{\mathrm{ext}}\right|}.$$

By the second property in Lemma 36 and the definition $S_i^{\mathrm{ext}}$, it holds that

$$\forall 1 \leq i \leq m, \quad \left|S_i^{\mathrm{ext}}\right| \geq k - 1.$$

Combining with (29), we have

$$\begin{aligned}\Pr_C\left[\mathcal{B}(\mathcal{P})\right] &\leq \sum_{c \in [s]^{m-1}} \Pr_C\left[\bigwedge_{i=1}^{m-1} \mathcal{B}_i(c_i)\right] \leq \sum_{c\in[s]^{m-1}}\left(\frac{1.16}{\sqrt{q}}\left(1 + \frac{5}{k}\right)\right)^{(m-1)(k-1)}\\ &\leq \left(s\left(\frac{1.16}{\sqrt{q}}\left(1 + \frac{5}{k}\right)\right)^{k-1}\right)^{m-1}.\end{aligned}$$

Now we claim that

$$s\left(\frac{1.16}{\sqrt{q}}\left(1+\frac{5}{k}\right)\right)^{k-1} \leq \frac{1}{10^3\Delta k^6}.$$

Using $s = \lceil\sqrt{q}\rceil \leq 1.16\sqrt{q}$, it suffices to show that

$$1.16 \times 10^3 (1.16)^{k-1}\left(1+\frac{5}{k}\right)^{k-1} \Delta k^6 \leq q^{(k-2)/2}.$$

Using $\left(1+\frac{5}{k}\right)^{\frac{2(k-1)}{k-2}} \leq 1.7$ and $k^{12/(k-2)} \leq 7.4$ for $k \geq 20$, we further simplifies the condition into

$$q \geq 7.4 \times 1.7 \times (1.16 \times 10^3)^{2/(k-2)}(1.16)^{2(k-1)/(k-2)}\Delta^{2/(k-2)},$$

which is implied by $q \geq 40\Delta^{\frac{2}{k-4}}$ and $k \geq 20$.

The claim implies that

$$\Pr_C\left[\mathcal{B}(\mathcal{P})\right] \leq \left(\frac{1}{10^3\Delta k^6}\right)^{m-1} = 10^3\Delta k^6\left(\frac{1}{10^3\Delta k^6}\right)^{m}.$$

Finally, by the third property in Lemma 36, we have

$$\Pr_C\left[\mathcal{B}(\mathcal{P})\right] \leq 10^3\Delta k^6\left(\frac{1}{10^3\Delta k^6}\right)^{R_{\mathrm{out}}+\frac{1}{3}(R_{\mathrm{self}}-b)}. \qquad \square$$

# References


[Alo91] Noga Alon. A parallel algorithmic version of the local lemma. *Random Struct. Algorithms*, 2(4):367–378, 1991.

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

[Bec91] József Beck. An algorithmic approach to the Lovász local lemma. I. *Random Struct. Algorithms*, 2(4):343–366, 1991.

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

[CS00] Artur Czumaj and Christian Scheideler. Coloring nonuniform hypergraphs: A new algorithmic approach to the general Lovász local lemma. *Random Struct. Algorithms*, 17(3-4):213–237, 2000.

[EL75] 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. 1975.

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

[FGYZ21] 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.

[FHY20] Weiming Feng, Kun He, and Yitong Yin. Sampling constraint satisfaction solutions in the local lemma regime. *arXiv*, abs/2011.03915, 2020.

[FM11] Alan M. Frieze and Páll Melsted. Randomly coloring simple hypergraphs. *Inf. Process. Lett.*, 111(17):848–853, 2011.

[FM13] Alan Frieze and Dhruv Mubayi. Coloring simple hypergraphs. *J. Combin. Theory Ser. B*, 103(6):767–794, 2013.

[GGW21] Andreas Galanis, Heng Guo, and Jiaheng Wang. Inapproximability of counting hypergraph colourings. *arXiv preprint arXiv:2107.05486*, 2021.

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

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

[GST16] Heidi Gebauer, Tibor Szabó, and Gábor Tardos. The local lemma is asymptotically tight for SAT. *J. ACM*, 63(5):43:1–43:32, 2016.

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

[HSW21] Kun He, Xiaoming Sun, and Kewen Wu. Perfect sampling for (atomic) Lovász local lemma. *arXiv*, abs/2107.03932, 2021.

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

[Hub98] Mark Huber. Exact sampling and approximate counting techniques. In *STOC*, pages 31–40. ACM, 1998.

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

[JPV20] Vishesh Jain, Huy Tuan Pham, and Thuy Duong Vuong. Towards the sampling Lovász local lemma. *arXiv*, abs/2011.12196, 2020.

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

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

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

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

[Mos09] Robin A. Moser. A constructive proof of the Lovász local lemma. In *STOC*, pages 343–350. ACM, 2009.

[MR98] Michael Molloy and Bruce A. Reed. Further algorithmic aspects of the local lemma. In *STOC*, pages 524–529. ACM, 1998.

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

[PW96] 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.

[Sri08] Aravind Srinivasan. Improved algorithmic versions of the Lovász local lemma. In *SODA*, pages 611–620. SIAM, 2008.

[Sta99] Richard P. Stanley. *Enumerative combinatorics. Vol. 2*, volume 62 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1999.

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

[WW20] Ian M Wanless and David R Wood. A general framework for hypergraph colouring. *arXiv preprint arXiv:2008.00775*, 2020.