An Optimal Distributed $(\Delta + 1)$-Coloring Algorithm?[*]

Yi-Jun Chang
University of Michigan

Wenzheng Li
IIIS, Tsinghua University

Seth Pettie[†]
University of Michigan

Abstract

Vertex coloring is one of the classic symmetry breaking problems studied in distributed computing. In this paper we present a new algorithm for $(\Delta + 1)$-list coloring in the randomized LOCAL model running in $O(\log^* \Delta) + O(\mathsf{Det}_d(\operatorname{poly} \log n)) = O(\mathsf{Det}_d(\operatorname{poly} \log n))$ time, where $\mathsf{Det}_d(n')$ is the deterministic complexity of $(\deg + 1)$-list coloring on n'-vertex graphs. (In this problem, each v has a palette of size $\deg(v) + 1$.) This improves upon a previous randomized algorithm of Harris, Schneider, and Su [STOC'16, JACM'18] with complexity $O(\sqrt{\log \Delta} + \log \log n + \mathsf{Det}_d(\operatorname{poly} \log n))$, and, for some range of Δ, is much faster than the best known deterministic algorithm of Fraigniaud, Heinrich, and Kosowski [FOCS'16] and Barenboim, Elkin, and Goldenberg [PODC'18], with complexity $O(\sqrt{\Delta \log \Delta} \log^* \Delta + \log^* n)$.

Our algorithm *appears to be* optimal, in view of the $\Omega(\mathsf{Det}(\operatorname{poly} \log n))$ randomized lower bound due to Chang, Kopelowitz, and Pettie [FOCS'16], where Det is the deterministic complexity of $(\Delta + 1)$-list coloring. At present, the best upper bounds on $\mathsf{Det}_d(n')$ and $\mathsf{Det}(n')$ are both $2^{O(\sqrt{\log n'})}$ and use a black box application of network decompositions (Panconesi and Srinivasan [Journal of Algorithms'96]). It is quite possible that the true complexities of both problems are the same, asymptotically, which would imply the randomized optimality of our $(\Delta + 1)$-list coloring algorithm.

1   Introduction

Much of what we know about the LOCAL model has emerged from studying the complexity of four canonical symmetry breaking problems and their variants: maximal independent set (MIS), $(\Delta + 1)$-vertex coloring, maximal matching, and $(2\Delta - 1)$-edge coloring. The palette sizes "$\Delta + 1$" and "$2\Delta - 1$" are minimal to still admit a greedy sequential solution; here Δ is the maximum degree of any vertex.

Early work [27, 30, 4, 31, 28, 1] showed that all the problems are reducible to MIS, all four problems require $\Omega(\log^* n)$ time, even with randomization; all can be solved in $O(\text{poly}(\Delta) + \log^* n)$ time (optimal when Δ is constant), or in $2^{O(\sqrt{\log n})}$ time for any Δ. Until recently, it was actually consistent with known results that all four problems had the same complexity.

Kuhn, Moscibroda, and Wattenhofer (KWM) [25] proved that the "independent set" problems (MIS and maximal matching) require $\Omega\left(\min\left\{\frac{\log \Delta}{\log\log \Delta}, \sqrt{\frac{\log n}{\log\log n}}\right\}\right)$ time, with or without randomization, via a reduction from $O(1)$-approximate minimum vertex cover. This lower bound provably separated MIS/maximal matching from simpler symmetry-breaking problems like $O(\Delta^2)$-coloring, which can be solved in $O(\log^* n)$ time [27].

We now know the KMW lower bounds cannot be extended to the canonical coloring problems, nor to variants of MIS like $(2,t)$-ruling sets, for $t \geq 2$ [9, 8, 19]. Elkin, Pettie, and Su [15] proved that $(2\Delta - 1)$-list edge coloring can be solved by a randomized algorithm in $O(\log\log n + \mathsf{Det}(\text{poly}\log n))$ time, which shows that neither the $\Omega\left(\frac{\log \Delta}{\log\log \Delta}\right)$ nor $\Omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)$ KMW lower bound applied to this problem. Here $\mathsf{Det}(n')$ represents the *deterministic* complexity of the problem in question on n'-vertex graphs. Improving on [8, 35], Harris, Schneider, and Su [22] proved a similar separation for $(\Delta + 1)$-vertex coloring. Their randomized algorithm solves the problem in $O(\sqrt{\log \Delta} + \log\log n + \mathsf{Det}_d(\text{poly}\log n))$ time, where Det_d is the complexity of $(\deg + 1)$-list coloring.

The "$\mathsf{Det}(\text{poly}\log n)$" terms in the running times of [15, 22] are a consequence of the *graph shattering* technique applied to distributed symmetry breaking. Barenboim, Elkin, Pettie, and Schneider [8] showed that all the classic symmetry breaking problems could be reduced in $O(\log \Delta)$ or $O(\log^2 \Delta)$ time, w.h.p., to a situation where we have independent subproblems of size $\text{poly}\log(n)$, which can then be solved with the best available deterministic algorithm.[1] Later, Chang, Kopelowitz, and Pettie (CKP) [10] gave a simple proof illustrating *why* graph shattering is inherent to the LOCAL model: the randomized complexity of any locally checkable problem[2] is at least its deterministic complexity on $\sqrt{\log n}$-size instances.

The CKP lower bound explains why the state-of-the-art randomized symmetry breaking algorithms have such strange stated running times: they all depend on a randomized graph shattering routine (Rand.) and a deterministic (Det.) algorithm.

- $O(\log \Delta + 2^{O(\sqrt{\log\log n})})$ for MIS (Rand. due to [19] and Det. to [31]),

- $O(\sqrt{\log \Delta} + 2^{O(\sqrt{\log\log n})})$ for $(\Delta + 1)$-vertex coloring (Rand. due to [22] and Det. to [31]),

- $O(\log \Delta + (\log\log n)^3)$ for maximal matching (Rand. due to [8] and Det. to [16]),

- $O((\log\log n)^8)$ for $(2\Delta - 1)$-edge coloring (Rand. due to [15] and Det. to [17]).

In each, the term that depends on n is the complexity of the best deterministic algorithm, scaled down to $\text{poly}\log(n)$-size instances. In general, improvements in the deterministic complexities of these problems imply improvements to their randomized complexities, but only if the running times are improved in terms of "n" rather than "Δ." For example, a recent line of research has improved the complexity of $(\Delta + 1)$-coloring in terms of Δ, from $O(\Delta + \log^* n)$ [7], to $\tilde{O}(\Delta^{3/4}) + O(\log^* n)$ [5], to the state-of-the-art bound of $O(\sqrt{\Delta \log \Delta} \log^* \Delta + \log^* n)$ due to Fraigniaud, Heinrich, and Kosowski [18], as improved by Barenboim, Elkin, and Goldenberg [6]. These improvements do not have consequences for randomized coloring algorithms using graph shattering [8, 22] since we can only assume $\Delta = (\log n)^{\Omega(1)}$ in the shattered instances. See Table 1 for a summary of lower and upper bounds for distributed $(\Delta + 1)$-list coloring in the LOCAL model.

---

[1] In the case of MIS, the subproblems actually have size $\text{poly}(\Delta)\log n$, but satisfy the additional property that they contain distance-5 dominating sets of size $O(\log n)$, which is often just as good as having $\text{poly}\log(n)$ size. See [8, §3] or [19, §4] for more discussion of this.

[2] See Naor and Stockmeyer [30] or Chang and Pettie [12] for a formal definition of the class of locally checkable labeling (LCL) problems.

	Randomized		Deterministic	
Upper Bounds	$O(\mathsf{Det}_d(\text{poly}\log n))$	**new**	$O(\sqrt{\Delta\log\Delta}\log^*\Delta+\log^* n)$	[18, 6]
	$O(\sqrt{\log\Delta}+\log\log n+\mathsf{Det}_d(\text{poly}\log n))$	[22]	$O(\sqrt{\Delta}\log^{5/2}\Delta+\log^* n)$	[18]
	$O(\log\Delta+\mathsf{Det}_d(\text{poly}\log n))$	[8]	$O(\Delta^{3/4}\log\Delta+\log^* n)$	[5]
	$O(\log\Delta+\sqrt{\log n})$	[35]	$O(\Delta+\log^* n)$	[7]
	$O(\Delta\log\log n)$	[26]	$O(\Delta\log\Delta+\log^* n)$	[26]
	$O(\log n)$	[28, 1, 24]	$O(\Delta\log n)$	[4]
			$O(\Delta^2+\log^* n)$	[20, 27]
			$O(\Delta^{O(\Delta)}+\log^* n)$	[21]
			$2^{O(\sqrt{\log n})}$	[31]
			$2^{O(\sqrt{\log n\log\log n})}$	[4]
Lower Bounds	$\Omega(\log^* n)$	[30]	$\Omega(\log^* n)$	[27]
	$\Omega(\mathsf{Det}(\sqrt{\log n}))$	[10]		

Table 1: Development of lower and upper bounds for distributed $(\Delta+1)$-list coloring in the LOCAL model. The terms $\mathsf{Det}(n')$ and $\mathsf{Det}_d(n')$ are the deterministic complexities of $(\Delta+1)$-list coloring and $(\deg+1)$-list coloring on n'-vertex graphs. All algorithms listed, except for [22] and ours, also solve the $(\deg+1)$-list coloring problem.

2 Technical Overview

In the distributed LOCAL model, the undirected input graph $G = (V, E)$ and communications network are identical. Each $v \in V$ hosts a processor that initially knows $\deg(v)$, a unique $\Theta(\log n)$-bit $\mathrm{ID}(v)$, and global graph parameters $n = |V|$ and $\Delta = \max_{v\in V}\deg(v)$. In the $(\Delta+1)$-list coloring problem each vertex v also has a palette $\Psi(v)$ of allowable colors, with $|\Psi(v)| \geq \Delta + 1$. As vertices progressively commit to their final color, we also use $\Psi(v)$ to denote v's available palette, excluding colors taken by its neighbors in $N(v)$. Each processor is allowed unbounded computation and has access to a private stream of unbiased random bits. *Time* is partitioned into synchronized rounds of communication, in which each processor sends one unbounded message to each neighbor. At the end of the algorithm, each v declares its output label, which in our case is a color from $\Psi(v)$ that is distinct from colors declared by all neighbors in $N(v)$. Refer to [27, 34] for more on the LOCAL model and variants.

In this paper we prove that $(\Delta+1)$-list coloring can be solved in $O(\mathsf{Det}_d(\text{poly}\log n))$ time w.h.p. Our algorithm's performance is best contrasted with the $\Omega(\mathsf{Det}(\text{poly}\log n))$ randomized lower bound of [10], where Det is the deterministic complexity of $(\Delta+1)$-list coloring. Despite the syntactic similarity between the $(\deg+1)$- and $(\Delta+1)$-list coloring problems, there is no hard evidence showing their complexities are the same, asymptotically. On the other hand, every deterministic algorithmic technique developed for $(\Delta+1)$-list coloring applies equally well to $(\deg+1)$-list coloring [18, 6, 5, 31, 4]. In particular, there is only *one* technique that yields upper bounds in terms of n (independent of Δ), and that is network decompositions [4, 31].

Intellectually, our algorithm builds on a succession of breakthroughs by Schneider and Wattenhofer [35], Barenboim, Elkin, Pettie, and Schneider [8], Elkin, Pettie, and Su, [15], and Harris, Schneider, and Su [22], which we shall now review.

2.1 Fast Coloring using Excess Colors

Schneider and Wattenhofer [35] gave the first evidence that the canonical coloring problems may not be subject to the KMW lower bounds. They showed that for any constants $\epsilon > 0$ and $\gamma > 0$, when $\Delta \geq \log^{1+\gamma} n$ and the palette size is $(1+\epsilon)\Delta$, the vertex coloring can be solved w.h.p., in just $O(\log^* n)$ time [35, Corollary 14]. We informally sketch the idea of [35]. Consider the case where the palette size of each vertex is at least $k\Delta$, where $k \geq 2$ is a sufficiently large constant. Suppose each vertex v selects $k/2$ colors at random from its palette. A vertex v can successfully color itself if one of its selected colors is not selected by any

neighbor in $N(v)$. The total number of colors selected by vertices in $N(v)$ is at most $k\Delta/2$. Therefore, the probability that a color selected by v is also selected by someone in $N(v)$ is at most $1/2$, so v successfully colors itself with probability at least $1 - 2^{-k/2}$. In expectation, the degree of any vertex after this coloring procedure is reduced by a $2^{k/2}$ factor. In contrast, the number of *excess colors*, i.e., the current available palette size minus the number of uncolored neighbors, is non-decreasing over time. This creates a gap of roughly $k2^{k/2}$ between the degree and palette size. Intuitively, repeating the above procedure for $O(\log^* n)$ rounds suffices to color all vertices.

Similar ideas have also been applied in other papers [35, 15, 10]. However, for technical reasons, we cannot directly apply the results in these papers. The main technical difficulty of our setting is that we need to deal with oriented graphs with varying out-degrees; the guaranteed number of excess colors at a vertex depends on its out-degree, not the global parameter Δ.

In this paper, we prove Lemma 1, whose proof is in Section 5. Recall that $\Psi(v)$ denotes the palette of v. Intuitively, $|\Psi(v)| - \deg(v)$ is the number of excess colors of v.

Lemma 1. *Consider a directed acyclic graph, where vertex v is associated with a parameter $p_v \leq |\Psi(v)| - \deg(v)$ We write $p^\star = \min_{v \in V} p_v$. Suppose that there is a number $C = \Omega(1)$ such that all vertices v satisfy $\sum_{u \in N_{\mathrm{out}}(v)} 1/p_u \leq 1/C$. Let $d^\star$ be the maximum out-degree of the graph. There is an algorithm that takes $O\left(\max\{1, \log^*(p^\star) - \log^*(C)\}\right)$ time and achieves the following. Each vertex v remains uncolored with probability at most $\exp(-\Omega(\sqrt{p^\star})) + d^\star \exp(-\Omega(p^\star))$. This is true even if the random bits generated outside a constant radius around v are determined adversarially.*

We briefly explain the intuition underlying Lemma 1. Consider the following coloring procedure. Each vertex selects $C/2$ colors from its available colors randomly. Vertex v successfully colors itself if at least one of its selected colors is not in conflict with any color selected by vertices in $N_{\mathrm{out}}(v)$. For each color c selected by v, the probability that c is also selected by some vertex in $N_{\mathrm{out}}(v)$ is $(C/2)\sum_{u \in N_{\mathrm{out}}(v)} 1/p_u \leq 1/2$. Therefore, the probability that v still remains uncolored after this procedure is $\exp(-\Omega(C))$, improving the gap between the number of excess colors and the out-degree (i.e., the parameter C) exponentially. We are done after repeating this procedure for $O(\log^*(p^\star) - \log^*(C))$ rounds. Lemma 2 is a more user-friendly version of Lemma 1 for simpler situations.

Lemma 2. *Suppose $|\Psi(v)| \geq (1 + k)\Delta$ for each vertex v, and $k = \Omega(1)$. There is an algorithm that takes $O\left(\max\{1, \log^*(\Delta) - \log^*(k)\}\right)$ time and achieves the following. Each vertex v remains uncolored with probability at most $\exp(-\Omega(\sqrt{k\Delta}))$. This is true even if the random bits generated outside a constant radius around v are determined adversarially.*

Proof. We apply Lemma 1. Orient the graph arbitrarily, and then set $p_v = k\Delta$ for each v. Use the parameters $C = k$, $p^\star = k\Delta$, and $d^\star = \Delta$. The time complexity is $O\left(\max\{1, \log^*(p^\star) - \log^*(C)\}\right) = O\left(\max\{1, \log^*(\Delta) - \log^*(k)\}\right)$. The failure probability is $\exp(-\Omega(\sqrt{p^\star})) + d^\star \exp(-\Omega(p^\star)) = \exp(-\Omega(\sqrt{k\Delta}))$. $\square$

2.2   Gaining Excess Colors

Schneider and Wattenhofer [35] illustrated that vertex coloring can be performed very quickly, given enough excess colors. However, in the $(\Delta + 1)$-list coloring problem there *is just one excess color* initially, so the problem is how to *create* them. Elkin, Pettie, and Su [15] observed that if the graph induced by $N(v)$ is not too dense, then v can obtain a significant number of excess colors after *one* iteration of the following simple random coloring routine. Each vertex v, with probability $1/5$, selects a color c from its palette $\Psi(v)$ uniformly at random; then vertex v successfully colors itself by c if c is not chosen by any vertex in $N(v)$. Intuitively, if $N(v)$ is not too close to a clique, then a significant number of *pairs* of vertices in the neighborhood $N(v)$ get assigned the same color. Each such pair effectively reduces v's palette size by 1 but its degree by 2, thereby increasing the number of excess colors at v by 1.

There are many *global* measures of sparsity, such as arboricity and degeneracy. We are aware of two locality-sensitive ways to measure it: the $(1-\epsilon)$-*local sparsity* of [2, 15, 29, 36], and the ϵ-*friends* from [22, 22], defined formally as follows.

Definition 1 ([15]). *A vertex v is $(1-\epsilon)$-locally sparse if the subgraph induced by $N(v)$ has at most $(1-\epsilon)\binom{\Delta}{2}$ edges; otherwise v is $(1 - \epsilon)$-locally dense.*

3

Definition 2 ([22]). *An edge $e = \{u, v\}$ is an ϵ-friend edge if $|N(u) \cap N(v)| \geq (1 - \epsilon)\Delta$. We call u an ϵ-friend of v if $\{u, v\}$ is an ϵ-friend edge. A vertex v is ϵ-dense if v has at least $(1 - \epsilon)\Delta$ ϵ-friends, otherwise it is ϵ-sparse.*

Throughout this paper, we only use Definition 2. Lemma 3 shows that in $O(1)$ time we can create excess colors at all locally sparse vertices.

Lemma 3. *Consider the $(\Delta + 1)$-list coloring problem. There is an $O(1)$-time algorithm that colors a subset of vertices such that the following is true for each $v \in V$ with $\deg(v) \geq 0.9\Delta$.*

- *With probability $1 - \exp(-\Omega(\Delta))$, the number of uncolored neighbors of v is at least $\Delta/2$.*

- *With probability $1 - \exp(-\Omega(\epsilon^2\Delta))$, v has at least $\Omega(\epsilon^2\Delta)$ excess colors, where ϵ is the highest value such that v is ϵ-sparse.*

We briefly explain the intuition underlying Lemma 3. The algorithm underlying Lemma 3 is the random coloring routine described above. If a vertex v is ϵ-sparse, then there must be $\Omega(\epsilon^2\Delta^2)$ pairs of vertices $\{u, w\} \subseteq N(v)$ such that $\{u, w\}$ is <u>not</u> an edge. If $|\Psi(u) \cap \Psi(w)| = \Omega(\Delta)$,[3] then the probability that both u and v are colored by the same color is $\Omega(1/\Delta)$, so the expected number of excess colors created at v is at least $\Omega\left(\frac{\epsilon^2\Delta^2}{\Delta}\right) = \Omega(\epsilon^2\Delta)$.

Similar but slightly weaker lemmas were proved in [15, 22]. The corresponding lemma from [15] does not apply to *list* coloring, and the corresponding lemma from [22] obtains a high probability bound only if $\epsilon^4\Delta = \Omega(\log n)$. Optimizing this requirement is of importance, since this is the threshold about how locally sparse a vertex needs to be in order to obtain excess colors. Since this is not the main contribution of this work, the proof of Lemma 3 is left to Appendix B.

The notion of local sparsity is especially useful for addressing the $(2\Delta - 1)$-edge coloring problem [15], since it can be phrased as $(\Delta' + 1)$-vertex coloring the *line graph* $(\Delta' = 2\Delta - 2)$, which is everywhere $(\frac{1}{2} + o(1))$-locally sparse and is also everywhere $(\frac{1}{2} - o(1))$-sparse.

2.3 Coloring Locally Dense Vertices

In the vertex coloring problem we cannot count on any kind of local sparsity, so the next challenge is to make local *density* also work to our advantage. Harris, Schneider, and Su [22] developed a remarkable new graph decomposition that can be computed in $O(1)$ rounds of communication. The decomposition takes a parameter ϵ, and partitions the vertices into an ϵ-sparse set, and several vertex-disjoint ϵ-*dense* components induced by the ϵ-friend edges, each with weak diameter at most 2.

Based on this decomposition, they designed a $(\Delta + 1)$-list coloring algorithm that takes $O(\sqrt{\log \Delta} + \log \log n + \mathsf{Det}_d(\text{poly} \log n)) = O(\sqrt{\log \Delta}) + 2^{O(\sqrt{\log \log n})}$ time. We briefly overview their algorithm, as follows.

Coloring ϵ-Sparse Vertices. By utilizing the excess colors, Harris et al. [22] showed that the ϵ-sparse set can be colored in $O(\log \epsilon^{-1} + \log \log n + \mathsf{Det}_d(\text{poly} \log n))$ time using techniques in [15] and [8]. More specifically, they applied the algorithm of [15, Corollary 4.1] using the $\epsilon'\Delta = \Omega(\epsilon^2\Delta)$ excess colors, i.e., $\epsilon' = \Theta(\epsilon^2)$. This takes $O\left(\log(\epsilon^{-1})\right) + T\left(n, O\left(\frac{\log^2 n}{\epsilon'}\right)\right)$ time, where $T(n', \Delta') = O(\log \Delta' + \log \log n' + \mathsf{Det}_d(\text{poly} \log n'))$ is the time complexity of the $(\deg +1)$-list coloring algorithm of [8, Theorem 5.1] on n'-vertex graphs of maximum degree Δ'.

Coloring ϵ-Dense Vertices. For ϵ-dense vertices, Harris et al. [22] proved that by coordinating the coloring decisions within each dense component, it takes only $O(\log_{1/\epsilon}\Delta + \log \log n + \mathsf{Det}_d(\text{poly} \log n))$ time to color the dense sets, i.e., the bound *improves* as $\epsilon \to 0$. The time for the overall algorithm is minimized by choosing $\epsilon = \exp(-\Theta(\sqrt{\log \Delta}))$.

---

[3]If the condition is not met, then we have $|(\Psi(u) \cup \Psi(w)) \setminus \Psi(v)| = \Omega(\Delta)$, and so with constant probability one of u and v successfully colors itself with a color not in $\Psi(v)$, and this also increases the number of excess colors at v by 1.

The algorithm for coloring ϵ-dense vertices first applies $O(\log_{1/\epsilon} \Delta)$ iterations of *dense coloring steps* to reduce the maximum degree to $\Delta' = O(\log n) \cdot 2^{O(\log_{1/\epsilon} \Delta)}$, and then apply the (deg $+1$)-list coloring algorithm of [8, Theorem 5.1] to color the remaining vertices in $O(\log \Delta' + \log \log n + \mathsf{Det}_d(\mathrm{poly} \log n)) = O(\log_{1/\epsilon} \Delta + \log \log n + \mathsf{Det}_d(\mathrm{poly} \log n))$ time.

In what follows, we informally sketch the idea behind the dense coloring steps. To finish in $O(\log_{1/\epsilon} \Delta)$ iterations, it suffices that the maximum degree is reduced by a factor of $\epsilon^{-\Omega(1)}$ in each iteration. Consider an ϵ-dense vertex v in a component S induced by the ϵ-friend edges. Harris et al. [22, 22] proved that the number of ϵ-dense neighbors of v that are not in S is at most $\epsilon \Delta$. Intuitively, if we let each dense component output a random coloring that has no conflict within the component, then the probability that the color choice of vertex $v \in S$ is in conflict with an *external neighbor* is $O(\epsilon)$. Harris et al. [22, 22] showed that this intuition can be nearly realized, and they developed a coloring procedure that is able to reduce the maximum degree by a factor of $O(\sqrt{\epsilon^{-1}})$ in each iteration.

2.4 New Results

In this paper we give a fast randomized algorithm for $(\Delta + 1)$-vertex coloring. It is based on a hierarchical version of the Harris-Schneider-Su decomposition with $\Theta(\log \log \Delta)$ levels determined by an increasing sequence of sparsity thresholds $(\epsilon_1, \ldots, \epsilon_\ell)$, with $\epsilon_i = \sqrt{\epsilon_{i+1}}$. Following [22], we begin with a single iteration of the *initial coloring step* (Lemma 3), in which a constant fraction of the vertices are colored. The guarantee of this procedure is that any vertex v at the ith level (which is ϵ_i-dense but ϵ_{i-1}-sparse), has $\Omega(\epsilon_{i-1}^2 \Delta)$ *pairs* of vertices in its neighborhood $N(v)$ assigned the same color, thereby creating that many excess colors in the palette of v.

At this point, the most natural way to proceed is to apply a Harris-Schneider-Su style *dense coloring step* to each level, with the hope that each will take roughly constant time. Recall that (i) any vertex v at the ith level already has $\Omega(\epsilon_{i-1}^2 \Delta)$ excess colors, and (ii) the dense coloring step reduces the maximum degree by a factor of $\epsilon^{-\Omega(1)}$ in each iteration. Thus, in $O\left(\log_{1/\epsilon_i} \frac{\Delta}{\epsilon_{i-1}^{2.5} \Delta}\right) = O(1)$ time we should be able to create a situation where any uncolored vertices have $O(\epsilon_{i-1}^{2.5} \Delta)$ uncolored neighbors but $\Omega(\epsilon_{i-1}^2 \Delta)$ excess colors in their palette. With such a large gap, a Schneider-Wattenhofer style coloring algorithm (Lemma 2) should complete in very few additional steps.

In order to color ϵ_i-dense components efficiently, we need to maintain relatively large *lower bounds* on the available palette and relatively small *upper bounds* on the number of external neighbors (i.e., the neighbors outside the ϵ_i-dense component). Thus, it is important that when we first consider a vertex, we have not already colored too many of its neighbors. Roughly speaking, our algorithm classifies the dense components at level i into *large* and *small* blocks based on the component size, and partitions the set of all blocks of all levels into $O(1)$ *groups*. We apply the dense coloring steps *in parallel* for all blocks in the same group. Whenever we process a block B, we need to make sure that all its vertices have a large enough palette. For large blocks, the palette size guarantee comes from the lower bound on the block size. For small blocks, the palette size guarantee comes from the ordering of the blocks being processed; we will show that whenever a small block B is considered, each vertex $v \in B$ has a sufficiently large number of neighbors that have yet to be colored.

All of the above coloring steps together finish in $O(\log^* \Delta)$ time. The bottleneck procedure is the algorithm of Lemma 2, and the rest takes only $O(1)$ time. Each of these coloring steps may not color all vertices it considers. The vertices left uncolored are put in $O(1)$ classes, each of which induces $O(\mathrm{poly} \log n)$-size components, which can be colored in $\mathsf{Det}_d(\mathrm{poly} \log n)$ time deterministically. In view of Linial's lower bound [27] we have $\mathsf{Det}_d(\mathrm{poly} \log n) = \Omega(\log^* n)$ and the running time of our $(\Delta + 1)$-list coloring algorithm is

$$O(\log^* \Delta) + O(\mathsf{Det}_d(\mathrm{poly} \log n)) = O(\mathsf{Det}_d(\mathrm{poly} \log n)).$$

Organization. In Section 3 we define a hierarchical decomposition based on [22]. Section 4 gives a high-level description of the algorithm, which uses a variety of coloring routines whose guarantees are specified by the following lemmas.

- Lemma 1 analyzes the procedure ColorBidding, which is a generalization of the Schneider-Wattenhofer coloring routing; it is proved in Section 5.

- Lemma 3 shows that the procedure OneShotColoring creates many excess colors; it is proved in Appendix B.

- Lemmas 8–?? analyze two versions of an algorithm DenseColoringStep, which is a generalization of the Harris-Schneider-Su routine [22] for coloring locally dense vertices; they are proved in Section 6.

Appendix A reviews some standard concentration inequalities.

3 Hierarchical Decomposition

In this section, we extend the work of Harris, Schneider, and Su [22] to define a hierarchical decomposition of the vertices based on local sparsity. Let $G = (V, E)$ be the input graph, Δ be the maximum degree, and $\epsilon \in (0, 1)$ be a parameter. An edge $e = \{u, v\}$ is an ϵ-*friend edge* if $|N(u) \cap N(v)| \geq (1 - \epsilon)\Delta$. We call u an ϵ-*friend* of v if $\{u, v\}$ is an ϵ-*friend edge*. A vertex v is called ϵ-*dense* if v has at least $(1 - \epsilon)\Delta$ ϵ-friends, otherwise it is ϵ-*sparse*. Observe that it takes one round of communication to tell whether each edge is an ϵ-friend, and hence one round to classify all vertices as ϵ-sparse or dense.

We write V_ϵ^s (and V_ϵ^d) to be the set of ϵ-sparse (and ϵ-dense) vertices. Let v be a vertex in a set $S \subseteq V$ and $V' \subseteq V$. Define $\bar{d}_{S,V'}(v) = |(N(v) \cap V') \setminus S|$ to be the *external degree* of v with respect to S and V', and $a_S(v) = |S \setminus (N(v) \cup \{v\})|$ to be the *anti-degree* of v with respect to S. A connected component C of the subgraph formed by the ϵ-dense vertices and the ϵ-friend edges is called an ϵ-*almost clique*. This term makes sense in the context of Lemma 4 from [22], which summarizes key properties of almost cliques.

Lemma 4 ([22]). *Fix any $\epsilon < 1/5$. The following conditions are met for each ϵ-almost clique C, and each vertex $v \in C$.*

(i) $\bar{d}_{C,V_\epsilon^d}(v) \leq \epsilon\Delta$. *(Small external degree.)*

(ii) $a_C(v) \leq 3\epsilon\Delta$. *(Small anti-degree.)*

(iii) $|C| \leq (1 + 3\epsilon)\Delta$. *(Small size, a consequence of (ii).)*

(iv) $\mathrm{dist}_G(u, v) \leq 2$ for each $u, v \in C$. *(Small weak diameter.)*

3.1 A Hierarchy of Almost Cliques

Throughout this section, we fix some increasing sequence of sparsity parameters $(\epsilon_1, \ldots, \epsilon_\ell)$ and a subset of vertices $V^\star \subseteq V$, whose meaning will be explained shortly. The sequence $(\epsilon_1, \ldots, \epsilon_\ell)$ always adheres to Definition 3.

Definition 3. *A sequence $(\epsilon_1, \ldots, \epsilon_\ell)$ is a valid sparsity sequence if the following conditions are met: (i) $\epsilon_i = \sqrt{\epsilon_{i-1}} = \epsilon_1^{2^{-(i-1)}}$, and (ii) $\frac{1}{\epsilon_\ell} \geq K$ for some large enough constant K.*

Layers. Define $V_1 = V^\star \cap V_{\epsilon_1}^d$, and $V_i = V^\star \cap (V_{\epsilon_i}^d \setminus V_{\epsilon_{i-1}}^d)$, for $i > 1$. Define $V_{\mathsf{sp}} = V^\star \cap V_{\epsilon_\ell}^s = V^\star \setminus (V_1 \cup \cdots \cup V_\ell)$. It is clear that $(V_1, \ldots, V_\ell, V_{\mathsf{sp}})$ is a partition of $V^\star$. We call V_i the *layer-i* vertices, and call V_{sp} the *sparse vertices*. In other words, V_i is the subset of $V^\star$ that are ϵ_i-dense but ϵ_{i-1}-sparse. Remember that the definition of sparsity is with respect to the entire graph $G = (V, E)$ not the subgraph induced by $V^\star$.

Blocks. The layer-i vertices V_i are partitioned into *blocks* as follows. Let $\{C_1, C_2, \ldots\}$ be the set of ϵ_i-almost cliques, and let $B_j = C_j \cap V_i$. Then $(B_1, B_2, \ldots)$ is a partition of V_i. For each $B_j \neq \emptyset$, we call it a *layer-i block*. See Figure 1 for an illustration, where the shaded region indicates a layer-i block B.

A layer-i block B is a *descendant* of a layer-i' block B', $i < i'$, if B and B' are both subsets of the same $\epsilon_{i'}$-almost clique. Therefore, the set of all blocks in all layers naturally forms a rooted tree $\mathcal{T}$, where the root represents V_{sp}, and every other node represents a block in some layer. For example, in Figure 1, the blocks contained in $C_1, \ldots, C_k$ are at layers $1, \ldots, i-1$, and are all descendants of B.
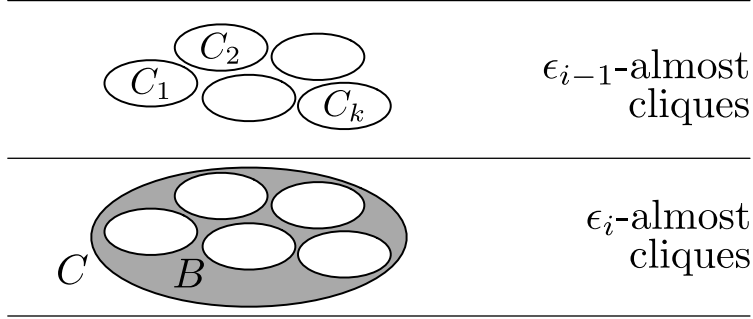
Figure 1: Almost-cliques and blocks.

3.2 Block Sizes and Excess Colors

We classify the blocks into three types: *small*, *medium*, and *large*. A block B at layer i is called *large-eligible* if

$$|B| \geq \frac{\Delta}{\log(1/\epsilon_i)}.$$

Large blocks. The set of large blocks is a maximal set of unrelated, large-eligible blocks, which prioritizes blocks by size, breaking ties by layer. More formally, a large-eligible layer i block B is large if and only if, for every large-eligible B' at layer j that is an ancestor or descendant of B, either $|B'| < |B|$ or $|B'| = |B|$ and $j < i$.

Medium blocks. Every large-eligible block that is not large is a medium block.

Small blocks. All other blocks are small.

Define V_i^{S}, V_i^{M}, and V_i^{L}, to be, respectively, the sets of all vertices in layer-i small blocks, layer-i medium blocks, and layer-i large blocks. For each $X \in \{\mathsf{S}, \mathsf{M}, \mathsf{L}\}$, we write $V_{2+}^X = \bigcup_{i=2}^{\ell} V_i^X$.

Overview of Our Algorithm. The decomposition and $\mathcal{T}$ are trivially computed in $O(1)$ rounds of communication. Let us briefly explain how our algorithm uses this hierarchical decomposition. The first step is to execute an $O(1)$-round coloring procedure (OneShotColoring) which colors a small constant fraction of the vertices in G. Let $V^{\star}$ be the remaining uncolored vertices. The set $V^{\star}$ is partitioned into subsets

$$(V_1^{\mathsf{S}}, \ldots, V_\ell^{\mathsf{S}}, V_1^{\mathsf{M}}, \ldots, V_\ell^{\mathsf{M}}, V_1^{\mathsf{L}}, \ldots, V_\ell^{\mathsf{L}}, V_{\mathsf{sp}})$$

based on the hierarchical decomposition with respect to a particular sparsity sequence $(\epsilon_1, \ldots, \epsilon_\ell)$.[4] We color the vertices of $V^{\star} \setminus V_{\mathsf{sp}}$ in six stages according to the ordering

$$(V_{2+}^{\mathsf{S}}, V_1^{\mathsf{S}}, V_{2+}^{\mathsf{M}}, V_1^{\mathsf{M}}, V_{2+}^{\mathsf{L}}, V_1^{\mathsf{L}}).$$

As we argue below, coloring vertices in the order small, medium, large ensures that when a vertex is considered, it has sufficiently many remaining colors in its palette. The reason for dealing with layer-1 vertices separately stems from the fact that a vertex at layer $i > 1$ is known to be ϵ_i-dense but ϵ_{i-1}-sparse, but layer-1 vertices are not known to have any non-trivial sparsity. At the end of this process a small portion of vertices $U \subseteq V^{\star} \setminus V_{\mathsf{sp}}$ may remain uncolored. However, they all have sufficiently large palettes such that $U \cup V_{\mathsf{sp}}$ can be colored efficiently in $O(\log^* n)$ time.

The purpose of processing the blocks in this ordering is to ensure that the vertices in small and medium blocks still have an *adequate* number of colors in their palettes when they are considered. Lemma 5 specifies exactly what an *adequate* number of colors is.

---

[4]Note that the classification of vertices into small, medium, and large blocks can only be done *after* OneShotColoring is complete. Recall that if C is an ϵ_i-almost clique, $B = C \cap V_i$ is the subset of C that is both ϵ_{i-1}-sparse <u>and</u> uncolored by OneShotColoring. Thus, whether the layer-i block in C is large-eligible depends on how many vertices are successfully colored.

Lemma 5. *For each layer $i \in [1, \ell]$, the following is true.*

- *For each $v \in V_i^{\mathsf{S}}$ with $|N(v) \cap V^{\star}| \geq \Delta/3$, we have $|N(v) \cap (V_{2+}^{\mathsf{M}} \cup V_1^{\mathsf{M}} \cup V_{2+}^{\mathsf{L}} \cup V_1^{\mathsf{L}} \cup V_{\mathsf{sp}})| \geq \frac{\Delta}{4}$.*

- *For each $v \in V_i^{\mathsf{M}}$, we have $|N(v) \cap (V_{2+}^{\mathsf{L}} \cup V_1^{\mathsf{L}} \cup V_{\mathsf{sp}})| \geq \frac{\Delta}{2\log(1/\epsilon_i)}$.*

In other words, regardless of how we proceed to *partially* color the vertices in small blocks, each $v \in V_i^{\mathsf{S}}$ always has at least $\frac{\Delta}{4}$ available colors in its palette. Similarly, regardless of how we partially color the vertices in small and medium blocks, each $v \in V_i^{\mathsf{M}}$ always has at least $\frac{\Delta}{2\log(1/\epsilon_i)}$ available colors in its palette.

Before proving Lemma 5 we first establish a useful property that constrains the structure of the block hierarchy $\mathcal{T}$. Intuitively, Lemma 6 shows that a node (block) in $\mathcal{T}$ can have exactly one child of essentially any size, but if it has two or more children then the union of all strict descendants must be small.

Lemma 6. *Let C be an ϵ_i-almost clique and $C_1, \ldots, C_l$ be the ϵ_{i-1}-almost cliques contained in C. Either $l = 1$ or $\sum_{j=1}^{l} |C_j| \leq 2(3\epsilon_i + \epsilon_{i-1})\Delta$. In particular, if B is the layer-i block contained in C, either B has one child in $\mathcal{T}$ or the number of vertices in all descendants of B is at most $2(3\epsilon_i + \epsilon_{i-1})\Delta < 7\epsilon_i\Delta$.*

Proof. Suppose, for the purpose of obtaining a contradiction, that $l \geq 2$ and $\sum_{j=1}^{l} |C_j| > 2(3\epsilon_i + \epsilon_{i-1})\Delta$. Without loss of generality, suppose C_1 is the smallest, so $\sum_{j=2}^{l} |C_j| > (3\epsilon_i + \epsilon_{i-1})\Delta$. Any $v \in C_1$ is ϵ_{i-1}-dense and therefore has at least $(1 - \epsilon_{i-1})\Delta$ neighbors that are ϵ_{i-1}-friends. By the anti-degree property of Lemma 4, v is adjacent to all but at most $3\epsilon_i\Delta$ vertices in C. Thus, by the pigeonhole principle v is joined by edges to more than $\epsilon_{i-1}\Delta$ members of $C_2 \cup \cdots \cup C_l$. By the pigeonhole principle again, at least one of these edges is one of the ϵ_{i-1}-friend edges incident to v. Any ϵ_{i-1}-friend edge is also an ϵ_i-friend edge, which means that C_1 is *not* a connected component in the graph formed by ϵ_i-dense vertices and ϵ_i-friend edges. $\square$

Proof of Lemma 5. The case of $v \in V_i^{\mathsf{M}}$ is easy. Let B be the layer-i medium block containing v. Every medium block is large-eligible but not large, meaning it must have a large ancestor or descendant B' with at least as many vertices. If B' is a layer-j block, then

$$|B'| \geq |B| \geq \frac{\Delta}{\log(1/\epsilon_k)}, \quad \text{where } k = \max\{i, j\}.$$

Let C be the layer-k almost clique containing both B and B'. By Lemma 4, v has at most $3\epsilon_k\Delta$ non-neighbors in C, which, since $B' \subseteq C$, means that the number of neighbors of v in B' is at least

$$
\begin{aligned}
|B'| - 3\epsilon_k\Delta &\geq \frac{\Delta}{\log(1/\epsilon_k)} - 3\epsilon_k\Delta \\
&\geq \frac{\Delta}{2\log(1/\epsilon_k)} && \{\epsilon_k \leq \epsilon_\ell \text{ sufficiently small}\} \\
&\geq \frac{\Delta}{2\log(1/\epsilon_i)} && \{\log(1/\epsilon_k) \leq \log(1/\epsilon_i)\}
\end{aligned}
$$

Therefore, $|N(v) \cap (V_{2+}^{\mathsf{L}} \cup V_1^{\mathsf{L}} \cup V_{\mathsf{sp}})| \geq \frac{\Delta}{2\log(1/\epsilon_i)}$.

Now consider any vertex $v \in V_i^{\mathsf{S}}$ with $|N(v) \cap V^{\star}| \geq \Delta/3$. Let B be the layer-i small block containing v. We partition the set $N(v) \cap V^{\star}$ into three groups $A_1 \cup A_2 \cup A_3$.

$\quad A_1 = N(v) \cap (V_{2+}^{\mathsf{M}} \cup V_1^{\mathsf{M}} \cup V_{2+}^{\mathsf{L}} \cup V_1^{\mathsf{L}} \cup V_{\mathsf{sp}})$.

$\quad A_2 = $ the neighbors in all ancestor and descendant blocks of B, including B.

$\quad A_3 = $ the remaining neighbors.

To prove the lemma, it suffices to show that $|A_1| \geq \frac{\Delta}{4}$. Since $|A_1 \cup A_2 \cup A_3| \geq \frac{\Delta}{3}$, we need to prove $|A_2 \cup A_3| \leq \frac{\Delta}{12}$. According to Lemma 4,

$$|A_3| \leq \sum_{j=i}^{\ell} \epsilon_j\Delta \leq 2\epsilon_\ell\Delta,$$

8

We now turn to A_2. Define $i^\star \in [1, i-1]$ to be the largest index such that B has at least two descendants at layer $i^\star$, or let $i^\star = 0$ if no such index exists. Let $A_{2,\text{low}}$ be the set of vertices in A_2 residing in blocks at layers $1, \ldots, i^\star$, and let $A_{2,\text{high}} = A_2 \setminus A_{2,\text{low}}$. By the definition of small blocks,

$$|A_{2,\text{high}}| < \sum_{j=i^\star+1}^{\ell} \frac{\Delta}{\log(1/\epsilon_j)}$$
$$< \frac{2\Delta}{\log(1/\epsilon_\ell)}. \qquad\qquad \{\text{geometric sum}\}$$

If $i^\star = 0$ then $A_{2,\text{low}} = \emptyset$. Otherwise, by Lemma 6, the number of vertices in $A_{2,\text{low}}$ is at most $7\epsilon_{i^\star+1}\Delta \leq 7\epsilon_i\Delta \leq 7\epsilon_\ell\Delta$. Since ϵ_ℓ is a sufficiently small constant,

$$|A_2 \cup A_3| < 2\epsilon_\ell\Delta + \frac{2\Delta}{\log(1/\epsilon_\ell)} + 7\epsilon_\ell\Delta < \Delta/12,$$

which completes the proof. □

Discussion. In the preliminary version of this paper [11], the algorithm for coloring locally dense vertices consists of $O(\log^* \Delta)$ stages. In this paper we improve the number of stages to $O(1)$. This improvement does not affect the asymptotic runtime, but it simplifies the algorithm.

The reader might wonder why the definition of medium blocks is needed, as all layer-i medium blocks already have the block size lower bound $\frac{\Delta}{\log(1/\epsilon_j)}$, which guarantees a sufficiently large palette size lower bound for the vertices therein. It might be possible to consider all the medium blocks as large blocks, but this will destroy the property that for any two blocks B and B' in different layers, if B is a descendant of B', then B and B' cannot both be large; without this property, the coloring algorithm for large blocks will likely be more complicated.

4 Main Algorithm

Our algorithm follows the *graph shattering* framework [8]. In each step of the algorithm, we specify an invariant that all vertices must satisfy in order to continue to participate. Those *bad vertices* that violate the invariant are removed from consideration; they form connected components of size $O(\text{poly}\log n)$ w.h.p., so we can color them later in $\text{Det}_d(\text{poly}\log n)$ time. More precisely, the emergence of the small components is due to the following lemma [8, 16].

Lemma 7 (The Shattering Lemma). *Consider a randomized procedure that generates a subset of vertices $B \subseteq V$. Suppose that for each $v \in V$, we have $\Pr[v \in B] \leq \Delta^{-3c}$, and this holds even if the random bits not in $N^c(v)$ are determined adversarially. With probability at least $1 - n^{-\Omega(c')}$, each connected component in the graph induced by B has size at most $(c'/c)\Delta^{2c}\log_\Delta n$.*

As we will see, some parts of our randomized algorithm consist of $t = O(\log^* \Delta)$ steps, and so whether a vertex v is bad actually depends on the random bits in its radius-t neighborhood. Nonetheless, we are still able to apply Lemma 7. The reason is that we are able to show that, for any specified constant k, each vertex v becomes bad in one particular step with probability at most Δ^{-k}, and this is true regardless of the outcomes in all previous steps and the choices of random bits outside of a constant radius of v.

Sparsity Sequence. The sparsity sequence for our algorithm is defined by $\epsilon_1 = \Delta^{-1/10}$, $\epsilon_i = \sqrt{\epsilon_{i-1}}$ for $i > 1$, and $\ell = \Theta(\log\log \Delta)$ is the largest index such that $\frac{1}{\epsilon_\ell} \geq K$ for some sufficiently large constant K.

4.1 Initial Coloring Step

At any point in time, the number of *excess colors* at v is the size of v's remaining palette minus the number of v's uncolored neighbors. This quantity is obviously non-decreasing over time. In the first step of our coloring algorithm, we execute the algorithm of Lemma 3, which in $O(1)$ time colors a portion of the vertices. This algorithm has the property that each remaining uncolored vertex gains a certain number of excess colors, which depends on its local sparsity. In order to proceed a vertex must satisfy both of the following conditions:

- If v is ϵ_ℓ-dense, the number of uncolored neighbors of v is at least $\Delta/2$.

- if v is ϵ_i-dense but ϵ_{i-1}-sparse, v must have $\Omega(\epsilon_{i-1}^2 \Delta)$ excess colors.

If either fails to hold, v is put in the set V_{bad}. We invoke the conditions of Lemma 3 only with $\epsilon \geq \epsilon_1 = \Delta^{-1/10}$. Thus, if $\Delta = \Omega(\log^2 n)$, then with high probability (i.e., $1 - 1/\mathrm{poly}(n)$), $V_{\mathsf{bad}} = \emptyset$. Otherwise, each component of V_{bad} must, by Lemma 7, have size $O(\mathrm{poly}(\Delta) \cdot \log n) = O(\mathrm{poly} \log n)$, w.h.p. We do not invoke a deterministic algorithm to color V_{bad} just yet. In subsequent steps of the algorithm, we will continue to add bad vertices to V_{bad}. These vertices will be colored at the end of the algorithm.

4.2 Coloring Vertices by Layer

Define $V^\star$ to be the set of uncolored vertices that are not in V_{bad}. We compute the partition $V^\star = V_{2+}^{\mathsf{S}} \cup V_1^{\mathsf{S}} \cup V_{2+}^{\mathsf{M}} \cup V_1^{\mathsf{M}} \cup V_{2+}^{\mathsf{L}} \cup V_1^{\mathsf{L}} \cup V_{\mathsf{sp}}$.

In this section, we show how we can color most of the vertices in $V_{2+}^{\mathsf{S}} \cup V_1^{\mathsf{S}} \cup V_{2+}^{\mathsf{M}} \cup V_1^{\mathsf{M}} \cup V_{2+}^{\mathsf{L}} \cup V_1^{\mathsf{L}}$, leaving a small portion of uncolored vertices. We color the vertices in this order: $(V_{2+}^{\mathsf{S}}, V_1^{\mathsf{S}}, V_{2+}^{\mathsf{M}}, V_1^{\mathsf{M}}, V_{2+}^{\mathsf{L}}, V_1^{\mathsf{L}})$.

Consider the moment we begin to color V_{2+}^{S}. We claim that each layer-i vertex $v \in V_{2+}^{\mathsf{S}}$ must have at least $\Delta/6 > \frac{\Delta}{2\log(1/\epsilon_i)}$ excess colors w.r.t. V_{2+}^{S}. That is, its palette size minus the number of its neighbors in V_{2+}^{S} is large. There are two relevant cases to consider.

- If the condition $|N(v) \cap V^\star| \geq \Delta/3$ in Lemma 5 is already met, then v has at least $\Delta/4 > \Delta/6$ excess colors w.r.t. V_{2+}^{S}.

- If the condition $|N(v) \cap V^\star| \geq \Delta/3$ in Lemma 5 is not met, then at least $(\Delta/2 - \Delta/3) = \Delta/6$ neighbors of v were included in V_{bad} after the initial coloring step, and so v automatically has at least $\Delta/6$ excess colors w.r.t. V_{2+}^{S}.

Similarly, for the sets V_1^{S}, V_{2+}^{M}, and V_1^{M}, we have the same excess colors guarantee $\frac{\Delta}{2\log(1/\epsilon_i)}$ for each layer-i vertex therein.

We apply the following lemmas to color the locally dense vertices $V^\star \setminus V_{\mathsf{sp}}$; refer to Section 6 for their proofs. For small and medium blocks, we use Lemma 8 to color V_{2+}^{S} and V_{2+}^{M}, and use Lemma 9 to color V_1^{S} and V_1^{M}.

The reason that the layer-1 blocks need to be treated differently is that layer-1 vertices do not obtain excess colors from the initial coloring step (Lemma 3). For comparison, for $i > 1$, each layer-i vertex v is ϵ_{i-1}-sparse, and so v must have $\Omega(\epsilon_{i-1}^2 \Delta) = \Omega(\epsilon_i^4 \Delta)$ excess colors. If we reduce the degree of v to $\epsilon_i^5 \Delta$, then we obtain a sufficiently big gap between the excess colors and degree at v.

Lemma 8 (Small and medium blocks; layers other than 1). *Let $S = V_{2+}^{\mathsf{S}}$ or $S = V_{2+}^{\mathsf{M}}$. Suppose that each layer-i vertex $v \in S$ has at least $\frac{\Delta}{2\log(1/\epsilon_i)}$ excess colors w.r.t. S. There is an $O(1)$-time algorithm that colors a subset of S meeting the following condition. For each vertex $v \in V^\star$, and for each $i \in [2, \ell]$, with probability at least $1 - \exp(-\Omega(\mathrm{poly}(\Delta)))$, the number of uncolored layer-i neighbors of v in S is at most $\epsilon_i^5 \Delta$. Vertices that violate this property join the set V_{bad}.*

Lemma 9 (Small and medium blocks; layer 1). *Let $S = V_1^{\mathsf{S}}$ or $S = V_1^{\mathsf{M}}$. Suppose that each vertex $v \in S$ has at least $\frac{\Delta}{2\log(1/\epsilon_1)}$ excess colors w.r.t. S. There is an $O(1)$-time algorithm that colors a subset of S meeting the following condition. Each $v \in S$ is colored with probability at least $1 - \exp(-\Omega(\mathrm{poly}(\Delta)))$, and all uncolored vertices in V_1^{S} join V_{bad}.*

The following lemmas consider large blocks. Lemma 10 colors V_{2+}^{L}. Lemma 11 colors V_1^{L}. Lemma **??** offers a different algorithm for coloring V_1^{L}, which is suitable for the case of $(\Delta + \log^k n)$-list coloring.

Lemma 10 (Large blocks; layer other than 1). *There is an $O(1)$-time algorithm that colors a subset of $S = V_{2+}^{\mathsf{L}}$ meeting the following condition. For each $v \in V^\star$ and each layer number $i \in [2, \ell]$, with probability at least $1 - \exp(-\Omega(\mathrm{poly}(\Delta)))$, the number of uncolored layer-i neighbors of v in S is at most $\epsilon_i^5 \Delta$. Vertices that violate this property join the set V_{bad}.*

Remember that our goal is to show that the bad vertices V_{bad} induce connected components of size $O(\text{poly}\log n)$. However, if in a randomized procedure each vertex is added to V_{bad} with probability $1 - 1/\text{poly}(\Delta)$, then the shattering lemma only guarantees that the size of each connected component of V_{bad} is $O(\text{poly}(\Delta, \log n))$, which is not enough for our purpose when $\Delta \gg \text{poly}\log n$. This explains why we need to switch to a different algorithm and analysis for the case Δ is large in Lemma 11. The algorithm for the small degree case in Lemma 11 does not rely on the condition $\Delta \leq \log^{100c} n$; but if we run this algorithm on a graph with $\Delta \gg \text{poly}\log n$, then we are unable to bound the component size of V_{bad} by $O(\text{poly}\log n)$.

Lemma 11 (Large blocks; layer 1). *Let c be any given sufficient large constant, i.e., $c > c_0$ for some universal constant c_0. Then there is an $O(1)$-time algorithm that colors a subset of $S = V_1^{\mathsf{L}}$ meeting the following requirement.*

- *If $\Delta \leq \log^{100c} n$, then the algorithm puts the remaining uncolored vertices in one of R or V_{bad}. Each $v \in V_1^{\mathsf{L}}$ is added to V_{bad} with probability at most $\Delta^{-\Omega(c)}$.*

- *If $\Delta > \log^{100c} n$, then the algorithm puts the remaining uncolored vertices in one of R or X, where each connected component of X has size $\log^{O(c)} n$, w.h.p.*

The subgraph induced by R has maximum degree $O(c^2)$. The runtime of the algorithm is independent of the choice of c.

In our $(\Delta+1)$-list coloring algorithm, we apply Lemmas 8, 9, 10, and 11 to color the vertices in $V^{\star} \setminus V_{\text{sp}}$, and they are processed in this order: $(V_{2+}^{\mathsf{S}}, V_1^{\mathsf{S}}, V_{2+}^{\mathsf{M}}, V_1^{\mathsf{M}}, V_{2+}^{\mathsf{L}}, V_1^{\mathsf{L}})$.

Coloring the Leftover Vertices X and R. Notice that the algorithm for Lemma 11 generates a leftover uncolored subset R which induces a constant-degree subgraph, and (in case $\Delta > \log^{100c} n$) a leftover uncolored subset X where each connected component has size at most $O(\text{poly}\log n)$. Remember that the vertices in R and X do not join V_{bad}. It takes $\mathsf{Det}_a(\text{poly}\log n)$ time to color all vertices in X deterministically; and the vertices in R can be colored deterministically in $O(\text{poly}(\Delta') + \log^* n) = O(\log^* n)$ time [27, 18, 6], with $\Delta' = O(1)$.

The Remaining Vertices. Any vertex in $V^{\star}$ that violates at least one condition specified in the lemmas is added to the set V_{bad}. All remaining uncolored vertices join the set U. In other words, U is the set of all vertices in $V^{\star} \setminus (V_{\text{bad}} \cup R \cup X)$ that remain uncolored after applying the lemmas.

4.3 Coloring the Remaining Vertices

At this point all uncolored vertices are in $U \cup V_{\text{sp}} \cup V_{\text{bad}}$. We show that $U \cup V_{\text{sp}}$ can be colored efficiently in $O(\log^* \Delta)$ time using Lemma 1.

Coloring the Vertices in U. We first consider the set U. Let G' be the directed acyclic graph induced by U, where all edges are oriented from the sparser to the denser endpoint. In particular, an edge $e = \{u, u'\}$ is oriented as (u, u') if u is at layer i, u' at layer i', and $i > i'$, or if $i = i'$ and $\text{ID}(u) > \text{ID}(u')$. We write $N_{\text{out}}(v)$ to denote the set of out-neighbors of v in G'.

For each layer-i vertex v in G', and each layer-j, the number of layer-j neighbors of v in G' is at most $O(\epsilon_j^5 \Delta)$, due to Lemmas 8 and 10. The out-degree of v is therefore at most $\sum_{j=1}^i \epsilon_j^5 \Delta = O(\epsilon_i^5 \Delta) = O(\epsilon_{i-1}^5 \Delta)$.

We write $\Psi(v)$ to denote the set of available colors of v. The number of excess colors at v is $|\Psi(v)| - \deg(v) = \Omega(\epsilon_{i-1}^2 \Delta)$. Thus, there is an $\Omega(1/\sqrt{\epsilon_{i-1}})$-factor gap between the palette size of v and the out-degree of v.

Lemma 1 is applied to color nearly all vertices in U in $O(\log^* \Delta)$ time, with any remaining uncolored vertices added to V_{bad}. We use the following parameters of Lemma 1. In view of the above, there exists a constant $\eta > 0$ such that, for each $i \in [2, \ell]$ and each layer-i vertex v in G', we set $p_v = \eta \epsilon_{i-1}^2 \Delta \leq |\Psi(v)| - \deg(v)$. There is a constant $C > 0$ such that for each $i \in [2, \ell]$ and each layer-i vertex $v \in U$ satisfies

$$\sum_{u \in N_{\text{out}}(v)} 1/p_u \leq \sum_{j=2}^i O\left(\frac{\epsilon_{j-1}^{2.5} \Delta}{\epsilon_{j-1}^2 \Delta}\right) = \sum_{j=2}^i O(\epsilon_{j-1}^{0.5}) < 1/C,$$

11

and we have:

$$p^\star = \eta\epsilon_1^2\Delta = \Omega(\Delta^{8/10}), \;\; d^\star = \Delta, \;\; C = \Omega(1).$$

Thus, by Lemma 1 the probability that a vertex still remains uncolored (and is added to V_{bad}) after the algorithm is

$$\exp(-\Omega(\sqrt{p^\star})) + d^\star \exp(-\Omega(p^\star)) = \exp(-\Omega(\Delta^{2/5})).$$

Coloring the Vertices in V_{sp}. The set V_{sp} can be colored in a similar way using Lemma 1. We let G'' be any acyclic orientation of the graph induced by V_{sp}, e.g., orienting each edge $\{u, v\}$ to the vertex v such that $\mathrm{ID}(v) > \mathrm{ID}(u)$. The number of available colors of each $v \in V_{\mathsf{sp}}$ minus its out-degree is at least $\Omega(\epsilon_\ell^2\Delta)$, which is at least $\gamma\Delta$, for some constant $\gamma > 0$, according to the way we select the sparsity sequence. We define $p_v = \gamma\Delta < |\Psi(v)| - \deg(v)$. We have $\sum_{u \in N_{\mathrm{out}}(v)}(1/p_u) \le \mathrm{outdeg}(v)/(\gamma\Delta) \le 1/\gamma$. Thus, we can apply Lemma 1 with $C = \gamma$. Notice that both $p^\star$ and $d^\star$ are $\Theta(\Delta)$, and so the probability that a vertex still remains uncolored after the algorithm (and is added to V_{bad}) is $\exp(-\Omega(\sqrt{\Delta}))$.

Coloring the Vertices in V_{bad}. At this point, all remaining uncolored vertices are in V_{bad}. If $\Delta \gg \mathrm{poly}\log n$, then $V_{\mathsf{bad}} = \emptyset$, w.h.p., in view of the failure probabilities $\exp(-\Omega(\mathrm{poly}(\Delta)))$ specified in the lemmas used in the previous coloring steps. Otherwise, $\Delta = O(\mathrm{poly}\log n)$, and by Lemma 7, each connected component of V_{bad} has size at most $\mathrm{poly}(\Delta, \log n) = O(\mathrm{poly}\log n)$. In any case, it takes $\mathsf{Det}_d(\mathrm{poly}\log n)$ to color all vertices in V_{bad} deterministically.

4.4 Time Complexity

The time for the initial coloring step is $O(1)$. The time for processing each of $(V_{2+}^{\mathsf{S}}, V_1^{\mathsf{S}}, V_{2+}^{\mathsf{M}}, V_1^{\mathsf{M}}, V_{2+}^{\mathsf{L}}, V_1^{\mathsf{L}})$ is $O(1)$. The time to color the vertices of $U \cup V_{\mathsf{sp}}$ not marked *bad* is $O(\log^* \Delta)$. In addition, we invoke $O(1)$ times

(i) an $O(\log^* n)$-time deterministic algorithm for coloring a bounded degree graph, namely the set R in Lemma 11, and

(ii) a $\mathsf{Det}_d(\mathrm{poly}\log n)$-time algorithm for coloring components of size $O(\mathrm{poly}\log n)$, for the set V_{bad} and the set X in Lemma 11.

Thus, the total time complexity is $O(\log^* n + \mathsf{Det}_d(\mathrm{poly}\log n)) = O(\mathsf{Det}_d(\mathrm{poly}\log n))$, in view of the $\mathsf{Det}_d(\mathrm{poly}\log n) = \Omega(\log^* n)$ lower bound of [27].

Theorem 1. *There is an algorithm that computes a $(\Delta + 1)$-list coloring, w.h.p., in $O(\mathsf{Det}_d(\mathrm{poly}\log n))$ time.*

Next, we argue that if we are given a sufficiently large amount of $O(\mathrm{poly}\log n)$ extra colors at the beginning, then we can improve the time complexity to just $O(\log^* \Delta)$.

Theorem 2. *There is a universal constant $\gamma > 0$ such that there is a randomized algorithm that, w.h.p., computes a $(\Delta + \log^\gamma n)$-list coloring in $O(\log^* \Delta)$ time.*

Proof. For all parts of our $(\Delta + 1)$-list coloring algorithm, except the case of small Δ in Lemma 11, the probability that a vertex v joins V_{bad} is $\exp(-\Omega(\mathrm{poly}(\Delta)))$. Let c be the constant in Lemma 11, and we select $k_1 = 100c$. We assume c is chosen to be sufficiently large so that if $\Delta > \log^{k_1} n$, then the probability that a vertex v joins V_{bad} in our $(\Delta+1)$-list coloring algorithm is $\exp(-\Omega(\mathrm{poly}(\Delta))) = 1/\mathrm{poly}(n)$. Note that when $\Delta > \log^{k_1} n$, no vertex is added to V_{bad} in Lemma 11.

Let $R' = R \cup X$ be the left-over vertices in Lemma 11 for the case $\Delta > \log^{k_1} n$. There exists a constant $k_2 > 0$ such that the subgraph induced by R' has maximum degree $\log^{k_2} n$. We set $\gamma = \max\{k_1, k_2\} + 1$. Now we show how to solve the $(\Delta + \log^\gamma n)$-list coloring problem in $O(\log^* \Delta)$ time.

If $\Delta \le \log^{\gamma-1} n$, then we apply the algorithm of Lemma 2 with $k = \frac{\log^\gamma n}{\Delta} - 1 = \Omega(\log n)$. The algorithm takes $O(\max\{1, \log^*(\Delta) - \log^*(k)\}) = O(1)$ time, and the probability that a vertex v is not colored is $\exp(-\Omega(\sqrt{k\Delta})) = \exp(-\Omega(\log^{\gamma/2} n)) \ll 1/\mathrm{poly}(n)$.

12

If $\Delta > \log^{\gamma-1} n$, then we apply our $(\Delta + 1)$-list coloring algorithm. Due to the lower bound on Δ, we have $V_{\mathsf{bad}} = \emptyset$, w.h.p. This algorithm takes $O(\log^* \Delta)$ time, and it produces an uncolored subgraph $R' = X \cup U$ that has maximum degree $\Delta' \leq \log^{k_2} n$. Again, we apply the algorithm of Lemma 2 to color R' in $O\left(\max\{1, \log^*(\Delta') - \log^*(k)\}\right) = O(1)$ time, where $k = \frac{\log^\gamma n}{\Delta'} - 1 = \Omega(\log n)$. $\square$

If every vertex is ϵ-sparse, with $\epsilon^2 \Delta$ sufficiently large, then the algorithm of Lemma 3 gives *every* vertex $\Omega(\epsilon^2 \Delta)$ excess colors, w.h.p. Combining this observation with Theorem 2, we have the following result, which shows that the $(\Delta + 1)$-list coloring problem can be solved very efficiently when all vertices are sufficiently locally sparse.

Theorem 3. *There is a universal constant $\gamma > 0$ such that the following holds. Suppose each vertex is ϵ-sparse, where the sparsity parameter ϵ is chosen such that $\epsilon^2 \Delta = \log^\gamma n$. There is a randomized algorithm that, w.h.p., computes a $(\Delta + 1)$-list coloring in $O(\log^* \Delta)$ time.*

Theorem 3 insists on every vertex being ϵ-sparse, as defined by Definition 2. It is straightforward to show connections between this definition of sparsity and others standard measures from the literature. For example, such a graph is $(1 - \epsilon')$-*locally sparse*, where $\epsilon' = \Omega(\epsilon^2)$, according to Definition 1. Similarly, any $(1-\epsilon')$-locally sparse graph is $\Omega(\epsilon')$-sparse. Graphs of *degeneracy* $d \leq (1 - \epsilon')\Delta$ or *arboricity* $\lambda \leq (1/2 - \epsilon')\Delta$ are trivially $(1 - \Omega(\epsilon'))$-locally sparse [8].

5 Fast Coloring using Excess Colors

In this section, we prove Lemma 1. Consider a directed acyclic graph $G = (V, E)$, where each vertex v has a palette $\Psi(v)$. Each vertex v is associated with a parameter $p_v \leq |\Psi(v)| - \deg(v)$, i.e., p_v is a lower bound on the number of excess colors at v. We write $p^\star = \min_{v \in V} p_v$ and $d^\star = \max_{v \in V} \mathrm{outdeg}(v)$. There is a number $C = \Omega(1)$ such that all vertices v satisfy $\sum_{u \in N_{\mathrm{out}}(v)} 1/p_u \leq 1/C$. Intuitively, the term $\sum_{u \in N_{\mathrm{out}}(v)} 1/p_u$ measures the amount of "contention" at a vertex v; this makes sense since each vertex u selects each color $c \in \Psi(u)$ with probability $\frac{C}{2|\Psi(v)|} < \frac{C}{2p_u}$ (which is proportional to $1/p_u$) in ColorBidding. All vertices agree on the value of C.

Procedure ColorBidding.

1. Each color $c \in \Psi(v)$ is added to S_v with probability $\frac{C}{2|\Psi(v)|}$ independently.

2. If there exists a color $c^\star \in S_v$ that is not selected by any vertex in $N_{\mathrm{out}}(v)$, v colors itself $c^\star$.

In Lemma 12 we present an analysis of ColorBidding. We show that after an iteration of ColorBidding, the amount of "contention" at a vertex v decreases by (roughly) an $\exp(C/6)$-factor, with very high probability.

Lemma 12. *Consider an execution of ColorBidding. Let v be any vertex. Let d be the summation of $1/p_u$ over all vertices u in $N_{\mathrm{out}}(v)$ that remain uncolored after ColorBidding. Then the following holds.*

$$\Pr[\ v \text{ remains uncolored }] \leq \exp(-C/6) + \exp(-\Omega(p^\star)).$$
$$\Pr[d \geq (1 + \lambda)\exp(-C/6)/C] \leq \exp\left(-2\lambda^2 p^\star \exp(-C/3)/C\right) + d^\star \exp(-\Omega(p^\star)).$$

Proof. For each vertex u, we define the following two events.

E_u^{good} : u selects a color that is not selected by any vertex in $N_{\mathrm{out}}(u)$.

E_u^{bad} : number of colors in $\Psi(u)$ that are selected by some vertices in $N_{\mathrm{out}}(u)$ is at least $\frac{2}{3} \cdot |\Psi(u)|$.

Notice that E_u^{good} is the event where u successfully colors itself. We show that $\Pr[E_u^{\mathrm{bad}}] = \exp(-\Omega(p^\star))$. Fix a color $c \in \Psi(u)$. The probability that c is selected by some vertex in $N_{\mathrm{out}}(u)$ is

$$1 - \prod_{w \in N_{\mathrm{out}}(u)} \left(1 - \frac{C}{2|\Psi(w)|}\right) \leq 1 - \prod_{w \in N_{\mathrm{out}}(u)} \left(1 - \frac{C}{2p_w}\right) \leq \sum_{w \in N_{\mathrm{out}}(u)} \frac{C}{2p_w} \leq \frac{1}{2}.$$

Thus, $\Pr[E_u^{\mathrm{bad}}] \leq \Pr[\mathrm{Binomial}(n', p') \geq \frac{2n'}{3}]$ with $n' = |\Psi(u)| \geq p_u$ and $p' = \frac{1}{2}$. By a Chernoff bound, we have:

$$\Pr[E_u^{\mathrm{bad}}] \leq \exp(-\Omega(n'p')) = \exp(-\Omega(p^\star)).$$

Conditioned on $\overline{E_u^{\mathrm{bad}}}$, u will color itself unless it fails to choose *any* of $|\Psi(u)|/3$ specific colors from its palette. Thus,

$$\Pr[\overline{E_u^{\mathrm{good}}} \mid \overline{E_u^{\mathrm{bad}}}] \leq \left(1 - \frac{C}{2p_u}\right)^{|\Psi(u)|/3} \leq \left(1 - \frac{C}{2p_u}\right)^{\frac{p_u}{3}} \leq \exp\left(\frac{-C}{6}\right).$$

We are now in a position to prove the first inequality. The probability that v remains uncolored is at most $\Pr[E_u^{\mathrm{bad}}] + \Pr[\overline{E_u^{\mathrm{good}}} \mid \overline{E_u^{\mathrm{bad}}}]$, which is at most $\exp(\frac{-C}{6}) + \exp(-\Omega(p^\star))$.

Next, we prove the second inequality. Let $N_{\mathrm{out}}(v) = (u_1, \ldots, u_k)$. Let E_i^{bad} and E_i^{good} be short for $E_{u_i}^{\mathrm{bad}}$ and $E_{u_i}^{\mathrm{good}}$. By a union bound,

$$\Pr\left[\bigcup_{i=1}^{k} E_i^{\mathrm{bad}}\right] \leq \mathrm{outdeg}(v) \cdot \exp(-\Omega(p^\star)) \leq d^\star \cdot \exp(-\Omega(p^\star)).$$

Let $X_i = 1/p_{u_i}$ if *either* $\overline{E_i^{\mathrm{good}}}$ or E_i^{bad} occurs, and $X_i = 0$ otherwise. Let $X = \sum_{i=1}^{k} X_i$. Notice that if E_i^{bad} does not occur, for all $i \in [1, k]$, we have $X = d$.

Notice that $\mu \stackrel{\mathrm{def}}{=} \mathrm{E}[X] \leq \exp(-C/6)/C$, since $\Pr[\overline{E_i^{\mathrm{good}}} \mid \overline{E_i^{\mathrm{bad}}}] \leq \exp(\frac{-C}{6})$. Each variable X_i is within the range $[a_i, b_i]$, where $a_i = 0$ and $b_i = 1/p_{u_i}$. We have $\sum_{i=1}^{k}(b_i - a_i)^2 \leq \sum_{u \in N_{\mathrm{out}}(v)} 1/(p_u \cdot p^\star) \leq 1/(Cp^\star)$. By Hoeffding's inequality,[5] we have

$$\Pr[X \geq (1+\lambda)\exp(-C/6)/C] \leq \Pr[X \geq (1+\lambda)\mu]$$
$$\leq \exp\left(\frac{-2(\lambda\mu)^2}{\sum_{i=1}^{k}(b_i - a_i)^2}\right)$$
$$\leq \exp\left(-2(\lambda\exp(-C/6)/C)^2(p^\star C)\right)$$
$$= \exp\left(-2\lambda^2 p^\star \exp(-C/3)/C\right).$$

Thus,

$$\Pr[d \geq (1+\lambda)\exp(-C/6)/C] \leq \exp\left(-2\lambda^2 p^\star \exp(-C/3)/C\right) + d^\star \exp(-\Omega(p^\star)). \qquad \square$$

Proof of Lemma 1. In what follows, we show how Lemma 12 can be used to derive Lemma 1. Our plan is to apply ColorBidding for $O\left(\max\{1, \log^*(p^\star) - \log^*(C)\}\right)$ iterations. For the ith iteration we use the parameter C_i, which is defined as follows: $C_1 = \min\{\sqrt{p^\star}, C\}$, and $C_k = \min\left\{\sqrt{p^\star}, \frac{C_{k-1}}{(1+\lambda)\exp(\frac{-C_{k-1}}{6})}\right\}$ for $k > 1$. Here $\lambda > 0$ must be selected to be a sufficiently small constant in such a way that $(1 + \lambda)\exp(-C_{k-1}/6) < 1$, so the sequence (C_k) increases. For example, if $C \geq 6$ initially, we can fix $\lambda = 1$ throughout.

In each iteration of ColorBidding each vertex v uses the same parameter p_v, since the number of excess colors never decreases. The last iteration $k^\star = O\left(\max\{1, \log^*(p^\star) - \log^*(C)\}\right)$ is the minimum index k such that $C_k = \sqrt{p^\star}$.

At the end of the kth iteration ($1 \leq k \leq k^\star$), we have the following invariant $\mathcal{H}_k$ that we expect all vertices to satisfy:

- If $1 \leq k < k^\star$, for each uncolored vertex v after the kth iteration, we require the summation of $1/p_u$ over all uncolored vertices u in $N_{\mathrm{out}}(v)$ to be less than $1/C_{k+1}$.

- If $k = k^\star$, all vertices are colored at the end of the kth iteration.

---

[5] The variables $\{X_1, \ldots, X_k\}$ are not independent, but we are still able to apply Hoeffding's inequality. The reason is as follows. Assume that $N_{\mathrm{out}}(v) = (u_1, \ldots, u_k)$ is sorted in reverse topological order, and so for each $1 \leq j \leq k$, we have $N_{\mathrm{out}}(u_j) \cap \{u_j, \ldots, u_k\} = \emptyset$. Thus, conditioning on (i) $\overline{E_i^{\mathrm{bad}}}$ and (ii) *any* colors selected by vertices in $\bigcup_{1 \leq j < i} N_{\mathrm{out}}(u_j) \cup \{u_j\}$, the probability that $\overline{E_i^{\mathrm{good}}}$ occurs is still at most $\exp(\frac{-C}{6})$.

The purpose of the invariant $\mathcal{H}_k$ ($1 \leq k < k^\star$) is to guarantee that the parameter C_{k+1} is a valid parameter for the $(k+1)$th iteration. For each $1 \leq k \leq k^\star$, at the end of the kth iteration, we remove from consideration all vertices v violating $\mathcal{H}_k$, and add them to the set V_{bad}. Thus, by definition of $\mathcal{H}_{k^\star}$, after the last iteration, all vertices other than the ones in V_{bad} have been colored.

To prove the lemma, it suffices to show that with probability at most $\exp(-\Omega(\sqrt{p^\star})) + d^\star \exp(-\Omega(p^\star))$, a vertex v is removed, and this is true even if the randomness outside constant distance to v is determined adversarially. By Lemma 12 the probability that a vertex v is removed at the end of the kth iteration, where $1 \leq k < k^\star$, is at most

$$\exp(\Omega(p^\star/C_{k+1})) + d^\star \exp(-\Omega(p^\star)) \leq \exp(-\Omega(\sqrt{p^\star})) + d^\star \exp(-\Omega(p^\star)),$$

and the probability that a vertex v is removed at the end of the $k^\star$th iteration is at most $\exp(-C_{k^\star}/6) + \exp(-\Omega(p^\star)) \leq \exp(-\Omega(\sqrt{p^\star}))$. By a union bound over all $O(\max\{1, \log^*(p^\star) - \log^*(C)\})$ iterations, the probability that v is removed is $\exp(-\Omega(\sqrt{p^\star})) + d^\star \exp(-\Omega(p^\star))$. $\qquad\square$

6 Coloring Locally Dense Vertices

Throughout this section, we consider the following setting. We are given a graph $G = (V, E)$, where a subset of vertices are already colored, and we are also given a *subset S of the uncolored vertices*. The vertex set S is partitioned into g disjoint *clusters* $S = S_1 \cup S_2 \cup \cdots \cup S_g$, each with weak diameter 2. Our goal is to color a large fraction of the vertices in S in only constant time.

We assume that the edges within S are oriented from the sparser to the denser endpoint, breaking ties by comparing IDs. In particular, an edge $e = \{u, u'\}$ is oriented as (u, u') if u is at layer i, u' at layer i', and $i > i'$, or if u and u' are at the same layer i and $\mathrm{ID}(u) > \mathrm{ID}(u')$. We write $N_{\mathrm{out}}(v)$ to denote the set of out-neighbors of v. Notice that this orientation is *acyclic*.

In Section 6.1 we describe a procedure DenseColoringStep (version 1) that is efficient when each vertex has many excess colors w.r.t. S. It is analyzed in Lemma 13, which is then used to prove Lemmas 8, 9. In Section 6.2 we describe a procedure DenseColoringStep (version 2), which is a generalization of the procedure introduced by Harris, Schneider, and Su [22]. It is analyzed in Lemma 14, which is then used to prove Lemmas 10, 11.

6.1 Version 1 of DenseColoringStep — Many Excess Colors are Available

In this section we focus on the case where each vertex $v \in S$ has many excess colors w.r.t. S. We make the following assumptions about the vertex set S.

Excess colors. Each $v \in S$ is associated with a parameter Z_v, which indicates a lower bound on the number of excess colors of v w.r.t. S. That is, the palette size of v minus the $|N(v) \cap S|$ is at least Z_v.

External degree. For each cluster S_j, each vertex $v \in S_j$ is associated with a parameter D_v such that $|N_{\mathrm{out}}(v) \cap (S \setminus S_j)| \leq D_v$. We define $\delta_v = D_v/Z_v$.

We briefly explain how we choose the clustering $S = S_1 \cup S_2 \cup \cdots \cup S_g$ and set these parameters in the setting of Lemma 8 and Lemma 9. For Lemma 9, each cluster is the intersection of S and a layer-1 block. For Lemma 8, each cluster is the intersection of S and an ϵ_ℓ-almost clique. For each layer-i vertex v, we use $Z_v = \frac{\Delta}{2 \log(1/\epsilon_i)}$ and $D_v = \epsilon_i \Delta$. The choices of these parameters are valid in view of the excess colors implied by Lemma 5 and the external degree upper bound of Lemma 4.

Procedure DenseColoringStep (version 1).

1. Let $\pi : \{1, \ldots, |S_j|\} \to S_j$ be the permutation that lists S_j in increasing order by layer number, breaking ties by ID. For q from 1 to $|S_j|$, the vertex $\pi(q)$ selects a color $c(\pi(q))$ uniformly at random from $\Psi(\pi(q)) \setminus \{c(\pi(q')) \mid q' < q \text{ and } \{\pi(q), \pi(q')\} \in E(G)\}$.

2. Each $v \in S_j$ permanently colors itself $c(v)$ if $c(v)$ is not selected by any vertices in $N_{\mathrm{out}}(v)$.

Notice that π is a reverse topological ordering of S_j, i.e., $q' < q$ implies $\pi(q) \notin N_{\text{out}}(\pi(q'))$. Because each S_j has weak diameter 2, Step 1 of DenseColoringStep takes only $O(1)$ rounds of communication. Intuitively, the probability that a vertex $v \in S$ remains uncolored after DenseColoringStep (version 1) is at most δ_v. The following lemma gives us the probabilistic guarantee of the DenseColoringStep (version 1).

Lemma 13. *Consider an execution of* DenseColoringStep *(version 1). Let T be any subset of S, and let $\delta = \max_{v \in T} \delta_v$. For any $t \geq 1$, the number of uncolored vertices in T is at least t with probability at most* $\Pr[\text{Binomial}(|T|, \delta) \geq t]$.

Proof. Let $T = \{v_1, \ldots, v_{|T|}\}$ be listed in increasing order by layer number, breaking ties by vertex ID. Remember that vertices in T can be spread across multiple clusters in S. Imagine exposing the color choices of all vertices in S, one by one, in this order $v_1, \ldots, v_{|T|}$. The vertex v_k in cluster S_j will successfully color itself if it chooses any color not already selected by a vertex in $N_{\text{out}}(v_k) \cap (S \setminus S_j)$. Since $|N_{\text{out}}(v_k) \cap (S \setminus S_j)| \leq D_{v_k}$ and v_k has at least Z_{v_k} colors to choose from, the probability that it fails to be colored is at most $D_{v_k}/Z_{v_k} = \delta_{v_k} \leq \delta$, *independent of the choices made by higher priority vertices* $v_1, \ldots, v_{k-1}$. Thus, for any t, the number of uncolored vertices in T is stochastically dominated by the binomial variable $\text{Binomial}(|T|, \delta)$. $\square$

Proof of Lemma 8. We execute DenseColoringStep (version 1) for 6 iterations using the same parameters Z_v and D_v for all iterations. Recall that for each layer-i vertex v, we use $Z_v = \frac{\Delta}{2\log(1/\epsilon_i)}$ and $D_v = \epsilon_i \Delta$. Consider any vertex $v \in V^\star$, and any layer number $i \in [2, \ell]$. Let T be the set of layer-i neighbors of v in S. To prove Lemma 13, it suffices to show that after 6 iterations of DenseColoringStep (version 1), with probability $1 - \exp(-\Omega(\text{poly}(\Delta)))$, the number of uncolored vertices in T is at most $\epsilon_i^5 \Delta$.

We define the following parameters.

$$\delta = \max_{u \in T}\{\delta_u\} = 2\epsilon_i \log(1/\epsilon_i),$$
$$t_0 = |T|,$$
$$\text{and } t_k = \max\{(2\delta)t_{k-1}, \epsilon_i^5 \Delta\}.$$

Since $(2\delta)^6 |T| \leq \epsilon_i^5 \Delta$, we have $t_6 = \epsilon_i^5 \Delta$.

Assume that at the beginning of the kth iteration, the number of uncolored vertices in T is at most t_{k-1}. Indeed for $k = 1$, we initially have $t_0 = |T|$. By Lemma 13, after the kth iteration, the expected number of uncolored vertices in T is at most $\delta t_{k-1} \leq \frac{t_k}{2}$. By a Chernoff bound, with probability at most $\exp(-\Omega(t_k)) \leq \exp(-\Omega(\epsilon_i^5 \Delta)) = \exp(-\Omega(\text{poly}(\Delta)))$, the number of uncolored vertices in T is more than t_k.

Therefore, after 6 iterations of DenseColoringStep (version 1), with probability $1 - \exp(-\Omega(\text{poly}(\Delta)))$, the number of uncolored vertices in T is at most $t_6 = \epsilon_i^5 \Delta$, as required. $\square$

Proof of Lemma 9. In the setting of Lemma 9 we only consider layer-1 vertices. Since $\epsilon_1 = \Delta^{-1/10}$, we have $Z_v = \frac{\Delta}{2\log(1/\epsilon_1)}$, $D_v = \epsilon_1 \Delta$, and $\delta_v = D_v/Z_v = 2\epsilon_1 \log(1/\epsilon_1) \ll \Delta^{-1/20}$, for all vertices $v \in S$.

We do one iteration of DenseColoringStep (version 1). By Lemma 13 and a Chernoff bound, for each $v \in S$, the number of uncolored vertices of $N(v) \cap S$ is at most $2\delta_v \Delta < \Delta^{19/20}$ with probability $1 - \exp(-\Omega(\text{poly}(\Delta)))$. At this moment, we put all uncolored vertices $v \in S$ such that the number of uncolored vertices in $N(v) \cap S$ is greater than $\Delta^{19/20}$ to the set V_{bad}, and they are not considered in the subsequent steps of our algorithm.

Consider the graph G' induced by the remaining uncolored vertices in S. The maximum degree of G' is at most $\Delta' = \Delta^{19/20}$. Each vertex v in G' satisfies $|\Psi(v)| \geq Z_v = \frac{\Delta}{2\log(1/\epsilon_1)} > (1 + k)\Delta'$, where we define $k = \Delta^{1/21} \ll \frac{Z_v}{\Delta'} - 1$. We run the algorithm of Lemma 2 on G', and then put all vertices that still remain uncolored to the set V_{bad}. By Lemma 2, the runtime is $O(\log^*(\Delta) - \log^*(k)) = O(1)$, and the probability that a vertex v is added to V_{bad} is at most $\exp(-\Omega(\sqrt{k\Delta})) = \exp(-\Omega(\text{poly}(\Delta)))$. $\square$

6.2 Version 2 of DenseColoringStep — No Excess Colors are Available

In this section we focus on the case where there is no guarantee on the number of excess colors. The palette size lower bound of each vertex $v \in S_j$ comes from the assumption that $|S_j|$ is large, and v is adjacent to all but a very small portion of vertices in S_j. For the case $S = V_{2+}^{\mathsf{L}}$, each S_j is a layer-i large block with $i \in [2, \ell]$. For the case $S = V_1^{\mathsf{L}}$, each S_j is a layer-1 large block. For each $v \in S$, we define $N^\star(v)$ as the set of all vertices $u \in N(v) \cap S$ such that the layer number of u is smaller than or equal to the layer number of

v. We always have $N_{\mathrm{out}}(v) \subseteq N^\star(v)$. For the case of $S = V_1^{\mathsf{L}}$, all clusters $S_1, \dots, S_g$ are layer-1 blocks, and so $N^\star(v) = N(v) \cap S$. We make the following assumptions.

Identifiers. Each of the clusters $S_1, \dots, S_g$ is a large block, and it has a unique identifier such that for each $i > j$, each layer-i block gets a higher identifier than each layer-j block. We assume that the clusters $S_1, \dots, S_g$ are ordered in increasing order by their identifiers, i.e., $\mathrm{ID}(S_1) < \dots < \mathrm{ID}(S_g)$. We also assume that for each $1 \leq i < j \leq g$, and for each $u \in S_i$ and $v \in S_j$, we have $\mathrm{ID}(u) < \mathrm{ID}(v)$. It is straightforward to generate the cluster identifiers and vertex identifiers meeting these conditions. We do not require each cluster S_j to know the index j.

Degree upper bounds. Each cluster S_j is associated with a parameter D_j such that all $v \in S_j$ satisfy the conditions: (i) $|S_j \setminus (N(v) \cup \{v\})| = |S_j \setminus (N^\star(v) \cup \{v\})| \leq D_j$ (anti-degree upper bound) and (ii) $|N^\star(v) \setminus S_j| \leq D_j$ (external degree upper bound).

Shrinking rate. Each cluster S_j is associated with a parameter δ_j such that $\delta_j \geq \frac{D_j \log(|S_j|/D_j)}{|S_j|}$ and $\delta_j \leq 1/K$ for some large enough constant K.

The procedure DenseColoringStep (version 2) aims to successfully color a large fraction of the vertices in each cluster S_j. The performance of the algorithm at a cluster S_j depends on the parameter δ_j.

Procedure DenseColoringStep (version 2).

1. Each cluster S_j selects $(1 - \delta_j)|S_j|$ vertices u.a.r. and generates a permutation π of those vertices u.a.r. The vertex $\pi(q)$ selects a color $c(\pi(q))$ u.a.r. from

$$\Psi(\pi(q)) - \{c(\pi(q')) \mid q' < q \text{ and } \{\pi(q), \pi(q')\} \in E(G)\}.$$

2. Each $v \in S_j$ that has selected a color $c(v)$ permanently colors itself $c(v)$ if $c(v)$ is not selected by any vertices $u \in N_{\mathrm{out}}(v)$.

The number of uncolored vertices in S_j after an execution of DenseColoringStep (version 2) is always at least $\delta_j |S_j|$. Lemma 14 gives a probabilistic guarantee for this algorithm. The proof for this lemma is deferred to the end of this section.

Lemma 14. *Consider an execution of* DenseColoringStep *(version 2). Let T be any subset of S, and let $\delta = \max_{j : S_j \cap T \neq \emptyset} \delta_j$. For any number t, the probability that the number of uncolored vertices in T is at least t is at most $\binom{|T|}{t} \cdot (O(\delta))^t$.*

Our assumption about the identifiers of clusters and vertices guarantees that for each $v \in S_j$, we have $N_{\mathrm{out}}(v) \subseteq \bigcup_{i=1}^{j} S_i$. Therefore, in the proof of Lemma 14, we expose the random bits of the clusters in the order $(S_1, \dots, S_g)$. As long as the random bits used in $S_1, \dots, S_j$ are fixed, we know whether $v \in S_j$ successfully colors itself.

Our proofs of Lemmas 10, 11 are based on multiple iterations of DenseColoringStep (version 2). The number of iterations is to be determined. In each iteration, the parameters D_j and δ_j might be different. In subsequent discussion, the term *anti-degree* of $v \in S_j$ refers to the number of uncolored vertices in $S_j \setminus (N(v) \cup \{v\})$, and the term *external degree* of $v \in S_j$ refers to the number of uncolored vertices in $N^\star(v) \setminus S_j$. Suppose S_j is a layer-i block. The parameters for S_j in each iteration are as follows. Let $\beta > 0$ be a sufficiently large constant to be determined.

Degree upper bounds. By Lemma 4, $D_j^{(1)} = 3\epsilon_i \Delta$. For $k > 1$, the parameter $D_j^{(k)}$ is chosen such that $D_j^{(k)} \geq \beta \delta_j^{(k-1)} \cdot D_j^{(k-1)}$. We write $\mathcal{D}_j^{(k)}$ to denote the invariant that at the beginning of the kth iteration, $D_j^{(k)}$ is an upper bound for anti-degree and external degree of all uncolored vertices in S_j.

Cluster size upper bounds. By Lemma 4, $U_j^{(1)} = (1 + 3\epsilon_i)\Delta$. For $k > 1$, the parameter $U_j^{(k)}$ is chosen such that $U_j^{(k)} \leq \beta \delta_j^{(k-1)} \cdot U_j^{(k-1)}$. We write $\mathcal{U}_j^{(k)}$ to denote the invariant that at the beginning of the kth iteration, the number of uncolored vertices in S_j is at most $U_j^{(k)}$.

Cluster size lower bounds. $L_j^{(1)} = \frac{\Delta}{\log(1/\epsilon_i)}$. For $k > 1$, the parameter $L_j^{(k)}$ is chosen such that $L_j^{(k)} \geq \delta_j^{(k-1)} \cdot L_j^{(k-1)}$. We write $\mathcal{L}_j^{(k)}$ to denote the invariant that at the beginning of the kth iteration, the number of uncolored vertices in S_j is at least $L_j^{(k)}$.

Shrinking rates. For each k, the shrinking rate $\delta_j^{(k)}$ of cluster S_j for the kth iteration is chosen such that $1/K \geq \delta_j^{(k)} \geq \frac{D_j^{(k)} \log (L_j^{(k)}/D_j^{(k)})}{L_j^{(k)}}$, and we additionally require that $\delta_x^{(k)} \leq \delta_y^{(k)}$ if the layer number of S_x is smaller than the layer number of S_y, and $\delta_x^{(k)} = \delta_y^{(k)}$ if the layer number of S_x is the same as the layer number of S_y. In particular, this implies $\delta_1^{(k)} \leq \delta_2^{(k)} \leq \cdots \leq \delta_g^{(k)}$.

Other than these three parameters $D_j^{(1)} = 3\epsilon_i\Delta$, $U_j^{(1)} = (1 + 3\epsilon_i)\Delta$, and $L_j^{(1)} = \frac{\Delta}{\log(1/\epsilon_i)}$, the actual choice of the remaining parameters are to be determined (but they have to meet the above requirements). To maximize the progress of the coloring algorithm, we will apply the following definitions for most of the cases:

$$D_j^{(k)} = \beta\delta_j^{(l-1)} \cdot D_j^{(k-1)}, \qquad\qquad U_j^{(k)} = \beta\delta_j^{(k-1)} \cdot U_j^{(k-1)},$$

$$L_j^{(k)} = \delta_j^{(k-1)} \cdot L_j^{(k-1)}, \qquad\qquad \delta_j^{(k)} = \frac{D_j^{(k)} \log\left(L_j^{(k)}/D_j^{(k)}\right)}{L_j^{(k)}}.$$

Unless otherwise stated, in the proofs of Lemmas 10, 11, we use the above *default* definition of the parameters.

Validity of Parameters. We verify the validity of the parameter choices. Consider the first iteration. We show that the invariants $\mathcal{D}_j^{(1)}$, $\mathcal{U}_j^{(1)}$, and $\mathcal{L}_j^{(1)}$ are met initially, for each cluster S_j. The validity of the parameter $D_j^{(1)}$ is due to Lemma 4's bounds on the external degree (at most $\epsilon_i\Delta$) and anti-degree (at most $3\epsilon_i\Delta$) of a vertex in an ϵ_i-almost clique. Since S_j is a layer-i large block, we have $|S_j| \geq \frac{\Delta}{\log(1/\epsilon_i)} = L_j^{(1)}$. We also have $|S_j| \leq (1 + 3\epsilon_i)\Delta = U_j^{(1)}$ in view of the upper bound on the size of an ϵ_i-almost clique.

For $k > 1$, the invariants $\mathcal{D}_j^{(k)}$ and $\mathcal{U}_j^{(k)}$ might not hold. To resolve this issue, we simply remove all vertices in each cluster S_j such that at least one of $\mathcal{D}_j^{(k)}$ and $\mathcal{U}_j^{(k)}$ is not met. All the removed vertices are added to the set V_{bad}. Notice that by inspection of the procedure DenseColoringStep (version 2), the invariant $\mathcal{L}_j^{(k)}$ always holds.

Maintenance of Invariants. We calculate the probability for the invariants $\mathcal{D}_j^{(k+1)}$ and $\mathcal{U}_j^{(k+1)}$ to be met at a cluster S_j. In what follows, we analyze the kth iteration of the algorithm, and assume that $\mathcal{D}_j^{(k)}$ and $\mathcal{U}_j^{(k)}$ are met initially. Let $T \subseteq S$ be a set of vertices that are uncolored at the beginning of the kth iteration, and suppose $\delta_j^{(k)} = \max_{j':S_{j'} \cap T \neq \emptyset} \delta_{j'}^{(k)}$. By Lemma 14, after the kth iteration, the number of uncolored vertices in T is at least t is at most $\binom{|T|}{t} \cdot \left(O(\delta_j^{(l)})\right)^t$. Using this result, we derive the following bounds:

$$\Pr[\mathcal{U}_j^{(k+1)}] \geq 1 - \exp\left(-\Omega(U_j^{(k+1)})\right),$$
$$\Pr[\mathcal{D}_j^{(k+1)}] \geq 1 - O\left(U_j^{(k)}\right)\exp\left(-\Omega(D_j^{(k+1)})\right).$$

We first consider $\Pr[\mathcal{U}_j^{(k+1)}]$. We choose T as the set of uncolored vertices in S_j at the beginning of the kth iteration, and set $t = U_j^{(k+1)}$. We have $t = U_j^{(k+1)} = \beta\delta_j^{(k)} \cdot U_j^{(k)} \geq \beta\delta_j^{(k)}|T|$, and this implies $\delta_j^{(k)}|T|/t \leq 1/\beta$. If we select β to be a large enough constant, then

$$1 - \Pr[\mathcal{U}_j^{(k+1)}] \leq \binom{|T|}{t} \cdot \left(O(\delta_j^{(k)})\right)^t \leq \left(O(\delta_j^{(k)}) \cdot e|T|/t\right)^t \leq (O(1/\beta))^t = \exp(-\Omega(U_j^{(k+1)})).$$

Next, consider $\Pr[\mathcal{D}_j^{(k+1)}]$. For each vertex $v \in S_j$ that is uncolored at the beginning of the kth iteration, define $\mathcal{E}_v^a$ (resp., $\mathcal{E}_v^e$) as the event that the anti-degree (resp., external degree) of v at the end of the kth iteration is higher than D_j^{k+1}. If we can show that both $\Pr[\mathcal{E}_v^a]$ and $\Pr[\mathcal{E}_v^e]$ are at most $\exp\left(-\Omega(D_j^{(k+1)})\right)$, then we conclude $\Pr[\mathcal{D}_j^{(k+1)}] \geq 1 - O\left(U_j^{(k)}\right) \exp\left(-\Omega(D_j^{(k+1)})\right)$ by a union bound on at most $U_j^{(k)}$ vertices $v \in S_j$ that are uncolored at the beginning of the kth iteration.

We show that $\Pr[\mathcal{E}_v^e] \leq \exp\left(-\Omega(D_j^{(k+1)})\right)$. We choose T as the set of uncolored vertices in $N^\star(v) \setminus S_j$ at the beginning of the kth iteration, and set $t = D_j^{(k+1)}$. Since the layer number of each vertex in $N^\star(v) \setminus S_j$ is smaller than or equal to the layer number of S_j, our requirement about the shrinking rates implies that $\delta_j^{(k)} \geq \max_{j': S_{j'} \cap T \neq \emptyset} \delta_{j'}^{(k)}$.

We have $t = D_j^{(k+1)} = \beta \delta_j^{(k)} \cdot D_j^{(k)} \geq \beta |T|$, and this implies $\delta_j^{(k)} |T| / t \leq 1/\beta$. If we select β to be a large enough constant, then

$$\Pr[\mathcal{E}_v^e] \leq \binom{|T|}{t} \cdot \left(O(\delta_j^{(k)})\right)^t \leq \left(O(\delta_j^{(k)}) \cdot e|T|/t\right)^t \leq (O(1/\beta))^t = \exp(-\Omega(D_j^{(k+1)})).$$

The proof of $\Pr[\mathcal{E}_v^a] \leq \exp\left(-\Omega(D_j^{(k+1)})\right)$ is analogous. Based on the above probability calculation, in what follows, we prove Lemmas 10, 11.

Proof of Lemma 10. We perform 6 iterations of DenseColoringStep (version 2) using the default parameters. Recall that the shrinking rate for the kth iteration is $\delta_j^{(k)} = \frac{D_j^{(k)} \log(L_j^{(k)}/D_j^{(k)})}{L_j^{(k)}}$ for each cluster S_j. If S_j is a layer-i block, we have $\delta_j^{(k)} = O(\epsilon_i \log(1/\epsilon_i))$ for each $1 \leq k \leq 6$.

Consider any vertex $v \in V^\star$, and a layer number i, with $i \in [2, \ell]$. Let T be the set of layer-i neighbors of v in S. To prove Lemma 10, it suffices to show that after 6 iterations of DenseColoringStep (version 2), with probability $1 - \exp(-\Omega(\text{poly}(\Delta)))$, the number of uncolored vertices in T is at most $\epsilon_i^5 \Delta$.

Define $t_0 = |T|$, and $t_k = \max\{\beta \delta_j^{(k)} t_{k-1}, \epsilon_i^5 \Delta\}$. Here $\delta_j^{(k)}$ is the shrinking rate of any cluster S_j that is a layer-i block for the kth iteration. We have $t_6 = \epsilon_i^5 \Delta$.

Assume that at the beginning of the kth iteration, the number of uncolored vertices in T is at most t_{k-1}, and the invariants $\mathcal{D}_j^{(k)}$, $\mathcal{L}_j^{(k)}$, and $\mathcal{U}_j^{(k)}$ are met for each cluster S_j such that $S_j \cap T \neq \emptyset$. By Lemma 14, after the kth iteration, the number of uncolored vertices in T is more than t_k with probability at most

$$\binom{t_{k-1}}{t_k} \cdot \left(O(\delta_j^{(k)})\right)^{t_k} \leq \left(O(\delta_j^{(k)}) \cdot e t_{k-1}/t_k\right)^{t_k} \leq (O(1/\beta))^{t_k} = \exp(-\Omega(t_k)).$$

Notice that $\exp(-\Omega(t_k)) \leq \exp(-\Omega(\epsilon_i^5 \Delta)) = \exp(-\Omega(\text{poly}(\Delta)))$. For the maintenance of the invariants, $\mathcal{L}_j^{(k+1)}$ holds with probability 1; the probability that the invariants $\mathcal{D}_j^{(k+1)}$ and $\mathcal{U}_j^{(k+1)}$ are met for all clusters S_j such that $S_j \cap T \neq \emptyset$ is at least $1 - O(|T|) \exp(-\Omega(\text{poly}(\Delta))) = 1 - \exp(-\Omega(\text{poly}(\Delta)))$. By a union bound over all six iterations, with probability $1 - \exp(-\Omega(\text{poly}(\Delta)))$, the number of uncolored layer-i neighbors of v in S is at most $\epsilon_i^5 \Delta$. □

Proof of Lemma 11. In the setting of Lemma 11, we deal with only layer-1 large blocks, and so $D_1^{(k)} = \cdots = D_g^{(k)}$, $U_1^{(k)} = \cdots = U_g^{(k)}$, $L_1^{(k)} = \cdots = L_g^{(k)}$, $\delta_1^{(k)} = \cdots = \delta_g^{(k)}$, for each iteration k. For this reason we drop the subscripts. Our algorithm consists of three phases, as follows. Recall that c is a parameter related to failure probability specified in the statement of Lemma 11.

The Small Degree Case. We first focus on the case $\Delta \leq \log^{100c} n$, and we aim at designing an $O(1)$-time algorithm that colors a subset of $S = V_1^\mathsf{L}$ such that each $v \in V_1^\mathsf{L}$ is added to V_{bad} with probability at most $\Delta^{-\Omega(c)}$, and the subgraph induced by R has maximum degree $O(c^2)$. The correctness of the proof of this case does not rely the property that $\Delta \leq \log^{100c} n$.

Phase 1. The first phase consists of 9 iterations of DenseColoringStep (version 2), with the default parameters for each $1 \leq k \leq 9$. Due to the fact that $\epsilon_1 = \Delta^{-1/10}$, we have $\delta^{(k)} = O(\Delta^{-1/10} \log \Delta)$ for each

$1 \le k \le 9$. Therefore, at the end of the 9th iteration, we have the parameters $D^{(10)} = \Theta(\log^9 \Delta)$, $L^{(10)} = \Theta(\Delta^{1/10} \log^8 \Delta)$, and $U^{(10)} = \Theta(\Delta^{1/10} \log^9 \Delta)$. In view of the calculation in previous discussion, for a cluster S_j, the probability that all invariants (for $1 \le k \le 10$) are met is at least $1 - \exp(-\Omega(\log^9 \Delta))$. If a cluster S_j does not meet a invariant for some k, then all vertices in S_j are removed and added to V_{bad} before the kth iteration.

Phase 2. For the 10th iteration, we use a non-default shrinking rate $\delta^{(10)} = \Delta^{-1/20}$. We still define $U^{(11)} = \beta\delta^{(10)} \cdot U^{(10)} = \Theta(\Delta^{1/20} \log^9 \Delta)$ and $L^{(11)} = \delta^{(10)} \cdot L^{(10)} = \Theta(\Delta^{1/20} \log^8 \Delta)$ according to their default definition. Since $\beta\delta^{(10)} \cdot D^{(10)} = o(1)$, we should not adapt the default definition of $D^{(11)}$. Instead, we use $D^{(11)} = c$. Using the probability calculation in previous discussion, for each cluster S_j the invariant $\mathcal{U}^{(11)}$ holds with probability at least $1 - \exp(-\Omega(\Delta^{1/20}\text{poly}\log \Delta))$, and the invariant $\mathcal{L}^{(11)}$ holds automatically. We will show that for a given cluster S_j, the probability that $D^{(11)}$ is a valid degree bound (i.e., $\mathcal{D}^{(11)}$ holds) is at least $1 - \Delta^{-\Omega(c)}$. If a cluster S_j does not meet at least one of $\mathcal{U}^{(11)}$, $\mathcal{L}^{(11)}$, or $\mathcal{D}^{(11)}$, then all vertices in S_j are removed and added to V_{bad} before the 11th iteration.

Phase 3. For the 11th iteration, we use the default $\delta^{(11)} = \frac{D^{(11)} \log(L^{(11)}/D^{(11)})}{L^{(11)}} = \Theta\left(\frac{1}{\Delta^{1/20} \log^8 \Delta}\right)$. We will show that after the 11th iteration, for each cluster S_j, with probability at least $1 - \Delta^{-\Omega(c)}$, there are at most c^2 uncolored vertices $v \in S_j$ such that there is at least one uncolored vertex in $N_{\mathrm{out}}(v) \setminus S_j$. If S_j does not satisfy this property, we put all remaining uncolored vertices in S_j to V_{bad}. For each cluster S_j satisfying this property, in one additional round we can color all vertices in S_j but c^2 of them. At this point, the remaining uncolored vertices induce a subgraph R of maximum degree at most $c^2 + D^{(10)} = c^2 + c = O(c^2)$.

The choice of parameters are summarized as follows. Note that we use the default shrinking rate $\delta^{(i)} = \frac{D^{(i)} \log(L^{(i)}/D^{(i)})}{L^{(i)}}$ for all i except $i = 10$.

	$D^{(i)}$	$L^{(i)}$	$U^{(i)}$	$\delta^{(i)}$
$i \in [9]$	$\Theta\left(\Delta^{\frac{10-i}{10}} \log^{i-1} \Delta\right)$	$\Theta\left(\Delta^{\frac{11-i}{10}} \log^{i-2} \Delta\right)$	$\Theta\left(\Delta^{\frac{11-i}{10}} \log^{i-1} \Delta\right)$	$\Theta\left(\Delta^{\frac{-1}{10}} \log \Delta\right)$
$i = 10$	$\Theta\left(\log^9 \Delta\right)$	$\Theta\left(\Delta^{\frac{1}{10}} \log^8 \Delta\right)$	$\Theta\left(\Delta^{\frac{1}{10}} \log^9 \Delta\right)$	$\Delta^{\frac{-1}{20}}$
$i = 11$	c	$\Theta\left(\Delta^{\frac{1}{20}} \log^8 \Delta\right)$	$\Theta\left(\Delta^{\frac{1}{20}} \log^9 \Delta\right)$	$\Theta\left(\Delta^{\frac{-1}{20}} \log^{-8} \Delta\right)$

Analysis of Phase 2. Recall $\delta^{(10)} = \Delta^{1/20}$ and $D^{(10)} = \Theta(\log^9 \Delta)$. By Lemma 14, the probability that the external degree or anti-degree of $v \in S_j$ is at most c is:

$$1 - \binom{D^{(10)}}{c}\left(O(\delta^{(10)})\right)^c \ge 1 - \binom{O(\log^9 \Delta)}{c}\left(O(\Delta^{-1/20})\right)^c \ge 1 - \Delta^{-\Omega(c)}.$$

By a union bound over at most $U^{(10)} = \Theta(\Delta^{1/10} \log^9 \Delta)$ vertices $v \in S_j$ that are uncolored at the beginning of the 10th iteration, the parameter setting $D^{(11)} = c$ is a valid upper bound of external degree and anti-degree for S_j after the 10th iteration with probability at least $1 - \Delta^{-\Omega(c)}$.

Analysis of Phase 3. Consider a vertex $v \in S_j$ that is uncolored at the beginning of the 11th iteration. Define the event $\mathcal{E}_v$ as follows. The event $\mathcal{E}_v$ occurs if, after the 11th iteration, v is still uncolored, and there is at least one uncolored vertex in $N_{\mathrm{out}}(v) \setminus S_j$. Our goal is to show that the number of vertices $v \in S_j$ such that $\mathcal{E}_v$ occurs is at most c^2 with probability at least $1 - \Delta^{-\Omega(c)}$.

Consider any size-c^2 subset Y of S_j. We claim that (using Lemma 14) the probability that $\mathcal{E}_v$ occurs for all $v \in Y$ is at most

$$(D^{(11)})^{c^2} \cdot \left(O(\delta^{(11)})\right)^{c^2\left(1 + \frac{1}{D^{(11)}}\right)}.$$

The reason is as follows. Let $v \in Y$. If $\mathcal{E}_v$ occurs, then there exists a vertex $v' \in N_{\mathrm{out}}(v) \setminus S_j$ that is uncolored. The number of uncolored vertices in $N_{\mathrm{out}}(v) \setminus S_j$ at the beginning of the 11th iteration is at most $D^{(11)}$, so there are at most $(D^{(11)})^{c^2}$ ways of mapping each $v \in Y$ to a vertex $v' \in N_{\mathrm{out}}(v) \setminus S_j$ of

v. Let $T = \bigcup_{v \in Y}\{v, v'\}$. A vertex outside of S_j can be adjacent to at most $D^{(11)}$ vertices in S_j, and so $|T| \geq c^2\left(1 + \frac{1}{D^{(11)}}\right)$. The probability that all vertices in T are uncolored is $\left(O(\delta^{(11)})\right)^{c^2\left(1+\frac{1}{D^{(11)}}\right)}$ by Lemma 14. By a union bound over at most $(D^{(11)})^{c^2}$ choices of T, we obtain the desired probabilistic bound.

Recall that $U^{(11)} = \Theta(\Delta^{1/20}\log^9\Delta) = L^{(11)} \cdot \Theta(\log\Delta)$ and $L^{(11)} = \Theta(\Delta^{1/20}\log^8\Delta)$ are the cluster size upper bound and lower bound at the beginning of the 11th iteration. By a union bound over at most $(U^{(11)})^{c^2}$ choices of a size-c^2 subset of S_j, the probability f that there exists c^2 vertices $v \in S_j$ such that $\mathcal{E}_v$ occurs is

$$f = (U^{(11)})^{c^2} \cdot (D^{(11)})^{c^2} \cdot \left(O(\delta^{(11)})\right)^{c^2\left(1+\frac{1}{D^{(11)}}\right)}.$$

Recall that $D^{(11)} = c$ and $c > 0$ is a constant. We have

$$(U^{(11)})^{c^2} = \left(O(L^{(11)}\log\Delta)\right)^{c^2},$$
$$(D^{(11)})^{c^2} = O(1),$$
$$\left(O(\delta^{(11)})\right)^{c^2\left(1+\frac{1}{D^{(11)}}\right)} = \left(O(\tfrac{\log(L^{(11)})}{L^{(11)}})\right)^{c^2+c}.$$

Since $L^{(11)} = \Theta(\Delta^{1/20}\log^8\Delta)$, we have

$$f = (U^{(11)})^{c^2} \cdot (D^{(11)})^{c^2} \cdot \left(O(\delta^{(11)})\right)^{c^2\left(1+\frac{1}{D^{(11)}}\right)} = O(\text{poly}\log\Delta) \cdot \Delta^{-\Omega(c)} = \Delta^{-\Omega(c)},$$

as required.

The Large Degree Case. Now we turn to the case $\Delta > \log^{100c} n$. In this case, no vertex joins V_{bad}, but we are allow to have a subgraph X where each connected component has $\log^{O(c)} n$ vertices. Intuitively, the proof follows the same structure as the low degree case, but in Phase 1 we have to first reduce the maximum degree to $\Delta' = \log^{O(c)} n$, so that when we bound the size the components of the bad vertices by shattering lemma (Lemma 7), we get $O(\text{poly}(\Delta', \log n)) = \log^{O(c)} n$. The choice of parameters for the large degree case is as follows.

	$D^{(i)}$	$L^{(i)}$	$U^{(i)}$	$\delta^{(i)}$
$i \in [9]$	$\Theta\left(\Delta^{\frac{10-i}{10}}\log^{i-1}\Delta\right)$	$\Theta\left(\Delta^{\frac{11-i}{10}}\log^{i-2}\Delta\right)$	$\Theta\left(\Delta^{\frac{11-i}{10}}\log^{i-1}\Delta\right)$	$\Theta\left(\Delta^{\frac{-1}{10}}\log\Delta\right)$
$i = 10$	$\Theta\left(\max\{\log^9\Delta, \log n\}\right)$	$\Theta\left(\Delta^{\frac{1}{10}}\log^8\Delta\right)$	$\Theta\left(\Delta^{\frac{1}{10}}\log^9\Delta\right)$	$\Delta^{\frac{-1}{20}}\log^9\Delta$
$i = 11$	$\Theta\left(\log n\right)$	$\Theta\left(\Delta^{\frac{1}{20}}/\log\Delta\right)$	$\Theta\left(\Delta^{\frac{1}{20}}\right)$	$\Delta^{\frac{-1}{20}}\log^{5c} n$
$i = 12$	$\Theta\left(\log n\right)$	$\Theta\left(\log^{5c} n/\log\Delta\right)$	$\Theta\left(\log^{5c} n\right)$	$\log^{-2c} n$
$i = 13$	c	$\Theta\left(\log^{3c} n/\log\Delta\right)$	$\Theta\left(\log^{3c} n\right)$	$\Theta\left(\frac{\log\Delta}{\log^{3c} n}\right)$

We use the default shrinking rate $\delta^{(i)} = \frac{D^{(i)}\log(L^{(i)}/D^{(i)})}{L^{(i)}}$ for all i except $i \in \{9, 10, 11\}$. Phase 1 consists of all iterations $i \in [11]$; Phase 2 consists of iteration $i = 12$; Phase 3 consists of iteration $i = 13$. The algorithm and the analysis are similar to the small degree case, so in subsequent discussion we only point out the difference. In order to have all $\delta^{(i)} \ll 1$, we need to have $\Delta^{1/20} \gg \log^{5c} n$. Since we require $\Delta > \log^{100c} n$, this condition is guaranteed to be met.

Phase 1. In view of the calculation in previous discussion, for a cluster S_j, the probability that all invariants $\mathcal{U}^{(i)}$, $\mathcal{L}^{(i)}$, and $\mathcal{D}^{(i)}$ (for $1 \leq i \leq 12$) are met w.h.p., i.e., with probability least $1 - \exp(-\Omega(\log n)) = 1 - 1/\text{poly}(n)$, since all parameters $D^{(i)}$, $L^{(i)}$, and $U^{(i)}$ (for $1 \leq i \leq 12$) are chosen to be $\Omega(\log n)$. Therefore, no cluster S_j is removed due to a violation of an invariant.

Phase 2. Consider iteration $i = 12$. Similarly, it is straightforward that the invariants $\mathcal{U}^{(13)}$ and $\mathcal{L}^{(13)}$ hold w.h.p., since $L^{(13)} = \Omega(\log n)$ and $U^{(13)} = \Omega(\log n)$. Now we consider the invariant $\mathcal{D}^{(13)}$. By Lemma 14, the probability that the external degree or anti-degree of $v \in S_j$ is at most c is:

$$1 - \binom{D^{(12)}}{c} \left(O(\delta^{(12)})\right)^c \geq 1 - \binom{O(\log n)}{c} \left(O(\log^{-2c} n)\right)^c \geq 1 - (\log n)^{-\Omega(c^2)}.$$

This failure probability is not small enough to guarantee that $\mathcal{D}^{(13)}$ holds everywhere w.h.p. In the high degree case, no vertex is added to V_{bad}. If a vertex v belongs to a cluster S_j such that $\mathcal{D}^{(13)}$ does not hold, we add all vertices to the set X.

Phase 3. Similarly, we will show that after the 13th iteration, for each cluster S_j, with probability at least $1 - (\log n)^{-\Omega(c^2)}$, there are at most c^2 uncolored vertices $v \in S_j$ such that there is at least one uncolored vertex in $N_{\mathrm{out}}(v) \setminus S_j$. If S_j does not satisfy this property, we put all remaining uncolored vertices in S_j to X. For each cluster S_j satisfying this property, in one additional round we can color all vertices in S_j but c^2 of them. At this point, the remaining uncolored vertices induce a subgraph R of maximum degree at most $c^2 + D^{(13)} = c^2 + c = O(c^2)$. Following the analysis in the small degree case, the probability that a vertex v is added to X in the 13th iteration is

$$\begin{aligned}
f &= (U^{(13)})^{c^2} \cdot (D^{(13)})^{c^2} \cdot \left(O(\delta^{(13)})\right)^{c^2\left(1 + \frac{1}{D^{(13)}}\right)} \\
&= O\left(\log^{3c} n\right)^{c^2} \cdot O(1) \cdot O\left(\log^{3c} n / \log \Delta\right)^{-(c^2+c)} \\
&= O\left(\log^{-3c^2} n \cdot \log^{c^2+c} \Delta\right) \\
&= (\log n)^{-\Omega(c^2)}.
\end{aligned}$$

Size of Components in X. To bound the size of each connected component of X, we use the shattering lemma (Lemma 7). Define $G' = (V', E')$ as follows. The vertex set V' consists of all vertices in S that remains uncolored at the beginning of the iteration 12. Two vertices u and v are linked by an edge in E' if (i) u and v belong to the same cluster, or (ii) u and v are adjacent in the original graph G. It is clear that the maximum degree Δ' of G' is $U^{(12)} + D^{(12)} = O(\log^{5c} n)$. In view of the above analysis, the probability of $v \in X$ is $1 - (\log n)^{-\Omega(c^2)} = 1 - (\Delta')^{-\Omega(c)}$, and this is true even if the random bits outside of a constant-radius neighborhood of v in G' are determined adversarially. Applying Lemma 7 to the graph G', the size of each connected component of X is $O(\mathrm{poly}(\Delta', \log n)) = \log^{O(c)} n$, w.h.p., both in G' and in the original graph G, since G' is the result of adding some additional edges to the subgraph of G induced by V'. $\qquad \square$

For the rest of this section, we prove Lemma 14. Recall that in DenseColoringStep (version 2), a vertex in S_j can remain uncolored for two different reasons:

- it never selects a color, because it is not among the $(1 - \delta_j)|S_j|$ selected vertices, or

- it selects a color, but is later decolored because of a conflict with some vertex in $S_{j'}$ with $j' < j$.

Lemmas 15–17 analyze different properties of DenseColoringStep (version 2), which are then applied to prove Lemma 14. Throughout we assume $\delta_j < 1/K$ for some sufficiently large K.

Lemma 15. *Let $T = \{v_1, \ldots, v_k\}$ be any subset of S_j and $c_1, \ldots, c_k$ be any sequence of colors. The probability that v_i selects c_i in DenseColoringStep (version 2), for all $i \in [1, k]$, is $\left(O\left(\frac{\log(|S_j|/D_j)}{|S_j|}\right)\right)^{|T|}$.*

Proof. Let $p^\star$ be the probability that, for all $i \in [1, k]$, v_i selects c_i. Let $L = (1 - \delta_j)|S_j|$ be the number of selected vertices. Notice that if v_i is not among the selected vertices, then v_i will not select any color, and thus cannot select c_i. Since we are upper bounding $p^\star$, it is harmless to condition on the event that v_i is selected. We write p_i to denote the rank of $v_i \in T$ in the random permutation of S_j.

Suppose that the ranks $p_1, \ldots, p_k$ were fixed. Recall that each vertex $v_i \in S_j$ is adjacent to all but at most D_j vertices in S_j. Thus, at the time v_i is considered it must have at least

$$L - p_i + \delta_j |S_j| - D_j \geq L - p_i + D_j \log(|S_j|/D_j) - D_j = (L - p_i) + D_j(\log(|S_j|/D_j) - 1)$$

available colors to choose from, at most one of which is c_i. Thus,

$$p^\star \leq \operatorname*{E}_{p_1, \ldots, p_k} \left[\prod_{i=1}^{k} \frac{1}{(L - p_i) + D_j(\log(|S_j|/D_j) - 1)} \right].$$

We divide the analysis into two cases: (i) $k \geq L/2$ and (ii) $k < L/2$. For the case $k \geq L/2$, regardless of the choices of $p_1, \ldots, p_k$, we always have

$$\prod_{i=1}^{k} \frac{1}{(L - p_i) + D_j(\log(|S_j|/D_j) - 1)} \leq \frac{1}{k!} = (O(1/k))^k \leq (O(1/|S_j|))^{|T|}.$$

We now turn to the case $k < L/2$. Regardless of the values of $p_1, \ldots, p_{i-1}$, we always have

$$
\begin{aligned}
\operatorname{E}\left[\frac{1}{((L - p_i) + D_j(\log(|S_j|/D_j) - 1)} \,\middle|\, p_1, \ldots, p_{i-1} \right] &\leq \frac{\sum_{x=0}^{L-i} \frac{1}{x + D_j(\log(|S_j|/D_j) - 1)}}{L - (i-1)} \\
&\leq \frac{\sum_{x=0}^{L/2} \frac{1}{x + D_j(\log(|S_j|/D_j) - 1)}}{L/2} \\
&= O\left(\frac{\log L - \log(D_j(\log(|S_j|/D_j) - 1))}{L} \right) \\
&= O\left(\frac{\log(|S_j|/D_j)}{|S_j|} \right).
\end{aligned}
$$

Therefore $p^\star \leq \left(O\left(\frac{\log(|S_j|/D_j)}{|S_j|} \right) \right)^{|T|}$, as claimed. $\qquad \square$

Lemma 16. *Let T be any subset of S_j. The probability that all vertices in T are decolored in* DenseColoring-Step *(version 2) is $\left(O\left(\frac{D_j \log(|S_j|/D_j)}{|S_j|} \right) \right)^{|T|}$, even allowing the colors selected in $S_1, \ldots, S_{j-1}$ to be determined adversarially.*

Proof. There are in total at most $D_j^{|T|}$ different color assignments to T that can result in decoloring all vertices in T, since each vertex $v \in T \subseteq S_j$ satisfies $|N_{\mathrm{out}}(v) \setminus S_j| \leq |N^\star(v) \setminus S_j| \leq D_j$. By Lemma 15 (and a union bound over $D_j^{|T|}$ color assignments to T) the probability that all vertices in T are decolored is $D_j^{|T|} \cdot \left(O(\frac{\log(|S_j|/D_j)}{|S_j|}) \right)^{|T|} = \left(O(\frac{D_j \log(|S_j|/D_j)}{|S_j|}) \right)^{|T|}$. Recall that for each $v \in T \subseteq S_j$, we have $N_{\mathrm{out}}(v) \setminus S_j \subseteq \bigcup_{k=1}^{j-1} S_k$, and so whether v is decolored is independent of the random bits in $S_{j+1}, \ldots, S_g$. The above analysis (which is based on Lemma 15) holds even allowing the colors selected in $S_1, \ldots, S_{j-1}$ to be determined adversarially. $\qquad \square$

Lemma 17. *Let T be any subset of S_j. The probability that all vertices in T do not select a color in* DenseColoringStep *(version 2) is $(O(\delta_j))^{|T|}$. The probability only depends on the random bits within S_j.*

Proof. The lemma follows from the fact that in DenseColoringStep (version 2) a vertex $v \in S_j$ does not select a color with probability δ_j, and the events for two vertices $u, v \in S_j$ to not select a color are negatively correlated. $\qquad \square$

Proof of Lemma 14. Recall that we assume the clusters $S = \{S_1, \ldots, S_g\}$ are ordered in such a way that for any $u \in S_j$, we have $N_{\mathrm{out}}(u) \subseteq N^\star(u) \subseteq \bigcup_{k=1}^{j} S_k$. In the proof we expose the random bits of the clusters in the order $(S_1, \ldots, S_g)$.

Consider any subset $T \subseteq S$. Let $U = U_1 \cup U_2$ be a size-t subset $U \subseteq T$. We calculate the probability that all vertices in U_1 fail to select a color, and all vertices in U_2 are decolored. Notice that there are at most 2^t ways of partitioning U into $U_1 \cup U_2$.

Suppose that U_1 has size x. We write $U_1^{(j)} = U_1 \cap S_j$. Whether a vertex $v \in U_1^{(j)}$ fails to select a color only depends the random bits in S_j. Thus, by Lemma 17, the probability that all vertices in U_1 fail to select a color is at most $\prod_{j=1}^{k} (O(\delta_j))^{|U_1^{(j)}|} \leq (O(\delta))^{|U_1|}$. Recall $\delta = \max_{j : S_j \cap T \neq \emptyset} \delta_j$.

Similarly, suppose that U_2 has size y. We write $U_2^{(j)} = U_2 \cap S_j$. Whether a vertex $v \in U_2^{(j)}$ is decolored only depends the random bits in $S_1, \ldots, S_j$. However, regardless of the random bits in $S_1, \ldots, S_{j-1}$, the probability that all vertices in $U_2^{(j)}$ are decolored is $(O(\delta_j))^{|U_1^{(j)}|}$ by Lemma 16. Recall $\delta \geq \delta_j \geq \frac{\log(|S_j|/D_j)}{|S_j|}$. Thus, the probability that all vertices in U_2 are decolored is at most $\prod_{j=1}^{k} (O(\delta_j))^{|U_2^{(j)}|} \leq (O(\delta))^{|U_2|}$.

Therefore, by a union bound over (i) at most $\binom{|T|}{t}$ choices of a size-t subset $U \subseteq T$, and (ii) at most 2^t ways of partitioning U into $U_1 \cup U_2$, the probability that the number of uncolored vertices in T is at least t is at most $2^t \cdot \binom{|T|}{t} \cdot (O(\delta))^t = \binom{|T|}{t} \cdot (O(\delta))^t$. $\qquad\square$

7 Conclusion

We have presented a randomized $(\Delta+1)$-list coloring algorithm that requires $O(\log^* \Delta + \mathsf{Det}_d(\operatorname{poly}\log n)) = O(\mathsf{Det}_d(\operatorname{poly}\log n))$ rounds of communication, which comes close to the $\Omega(\mathsf{Det}(\sqrt{\log n}))$ lower bound implied by Chang, Kopelowitz, and Pettie [10]. Recall that Det and Det_d are the deterministic complexities of $(\Delta+1)$-list coloring and $(\deg+1)$-list coloring. When Δ is unbounded (relative to n), the best known algorithms for $(\Delta+1)$- and $(\deg+1)$-list coloring are the same: they use Panconesi and Srinivasan's [31] $2^{O(\sqrt{\log n})}$-time construction of network decompositions. Even if optimal $(O(\log n), O(\log n))$-network decompositions could be computed *for free*, we still do not know how to solve $(\Delta+1)$-list coloring faster than $O(\log^2 n)$ time. Thus, reducing the $\mathsf{Det}_d(\operatorname{poly}\log n)$ term in our running time below $O((\log\log n)^2)$ will require a radically new approach to the problem.

It is an open problem to generalize our algorithm to solve the $(\deg+1)$-coloring problem. One difficulty is to extend the definition of "ϵ-friend" to account for neighbors of different degrees, while still preserving the useful properties of ϵ-dense clusters from Lemma 4. To extend our approach to solve the $(\deg+1)$-*list* coloring problem, there is a technical barrier that appears to be difficult to overcome. Suppose the subgraph induced by $N(v)$ consists of two disjoint cliques S_1 and S_2, each of size $s = \deg(v)/2$. Each vertex in S_1 has palette $[1, s+1]$, each vertex in S_1 has palette $[s+1, 2s+1]$, and the palette of v is $[1, 2s+1]$. Notice that only the color $s+1$ is shared by all vertices in S_1 and S_2, and so regardless of how we design the initial coloring step, we can only gain at most one more excess color at v, even though the local sparsity of v is about $1/2$. In contrast, in the setting of $(\Delta+1)$-list coloring, any vertex with $\Omega(1)$ local sparsity is able to obtain $\Omega(\Delta)$ excess colors (with probability $1 - \exp(-\Omega(\Delta))$) from Lemma 3.

Subsequent to the publication of the preliminary version of this work [11], there has been a line of research applying the techniques developed in this work and [22] to solve the $(\Delta+1)$-coloring problem in the *congested clique* and the *massively parallel computation* (MPC) models [3, 32, 33]. In the congested clique model, the problem can be solved in $O(\log^* \Delta)$ rounds [33]; in the MPC model with $\tilde{O}(n)$ memory per processor, the problem can be solved in $O(1)$ rounds [3]. These results also do not generalize to the $(\deg+1)$-coloring problem.

References

[1] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algor.*, 7:567–583, 1986.

[2] N. Alon, M. Krivelevich, and B. Sudakov. Coloring graphs with sparse neighborhoods. *J. Comb. Theory, Ser. B*, 77(1):73–82, 1999.

[3] S. Assadi, Y. Chen, and S. Khanna. Sublinear Algorithms for $(\Delta+1)$ Vertex Coloring. In *Proceedings 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2019.

[4] B. Awerbuch, A. V. Goldberg, M. Luby, and S. A. Plotkin. Network decomposition and locality in distributed computation. In *Proceedings 30th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 364–369, 1989.

[5] L. Barenboim. Deterministic $(\Delta + 1)$-coloring in sublinear (in Δ) time in static, dynamic and faulty networks. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 345–354, 2015.

[6] L. Barenboim, M. Elkin, and U. Goldenberg. Locally-iterative distributed $(\Delta + 1)$-coloring below Szegedy-Vishwanathan barrier, and applications to self-stabilization and to restricted-bandwidth models. In *Proceedings 38th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 437–446, 2018.

[7] L. Barenboim, M. Elkin, and F. Kuhn. Distributed $(\Delta + 1)$-coloring in linear (in Δ) time. *SIAM J. Comput.*, 43(1):72–95, 2014.

[8] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider. The locality of distributed symmetry breaking. *J. ACM*, 63(3):20:1–20:45, 2016.

[9] T. Bisht, K. Kothapalli, and S. V. Pemmaraju. Brief announcement: Super-fast t-ruling sets. In *Proceedings 33rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 379–381, 2014.

[10] Y.-J. Chang, T. Kopelowitz, and S. Pettie. An exponential separation between randomized and deterministic complexity in the LOCAL model. In *Proceedings 57th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 615–624, 2016.

[11] Y.-J. Chang, W. Li, and S. Pettie. An optimal distributed $(\Delta + 1)$-coloring algorithm? In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 445–456. ACM, 2018.

[12] Y.-J. Chang and S. Pettie. A time hierarchy theorem for the LOCAL model. In *Proceedings 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2017.

[13] D. P. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[14] D. P. Dubhashi and D. Ranjan. Balls and bins: A study in negative dependence. *J. Random Structures and Algs.*, 13(2):99–124, 1998.

[15] M. Elkin, S. Pettie, and H.-H. Su. $(2\Delta - 1)$-edge coloring is much easier than maximal matching in the distributed setting. In *Proceedings 26th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 355–370, 2015.

[16] M. Fischer and M. Ghaffari. Sublogarithmic distributed algorithms for Lovász local lemma with implications on complexity hierarchies. In *Proceedings 31st International Symposium on Distributed Computing (DISC)*, pages 18:1–18:16, 2017.

[17] M. Fischer, M. Ghaffari, and F. Kuhn. Deterministic distributed edge coloring via hypergraph maximal matching. In *Proceedings 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2017.

[18] P. Fraigniaud, M. Heinrich, and A. Kosowski. Local conflict coloring. In *Proceedings 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 625–634, 2016.

[19] M. Ghaffari. An improved distributed algorithm for maximal independent set. In *Proceedings 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 270–277, 2016.

[20] A. Goldberg, S. Plotkin, and G. Shannon. Parallel symmetry-breaking in sparse graphs. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 315–324. ACM, 1987.

[21] A. V. Goldberg and S. A. Plotkin. Parallel ($\delta +$ 1)-coloring of constant-degree graphs. *Information Processing Letters*, 25(4):241–245, 1987.

[22] D. G. Harris, J. Schneider, and H. Su. Distributed ($\Delta + 1$)-coloring in sublogarithmic rounds. *J. ACM*, 65(4):19:1–19:21, 2018.

[23] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[24] Ö. Johansson. Simple distributed $\Delta + 1$-coloring of graphs. *Info. Proc. Lett.*, 70(5):229–232, 1999.

[25] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Local computation: Lower and upper bounds. *J. ACM*, 63(2):17:1–17:44, 2016.

[26] F. Kuhn and R. Wattenhofer. On the complexity of distributed graph coloring. In *Proceedings 25th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 7–15, 2006.

[27] N. Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992.

[28] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM J. Comput.*, 15(4):1036–1053, 1986.

[29] M. Molloy and B. A. Reed. A bound on the strong chromatic index of a graph. *J. Comb. Theory, Ser. B*, 69(2):103–109, 1997.

[30] M. Naor. A lower bound on probabilistic algorithms for distributive ring coloring. *SIAM J. Discrete Mathematics*, 4(3):409–412, 1991.

[31] A. Panconesi and A. Srinivasan. On the complexity of distributed network decomposition. *J. Algor.*, 20(2):356–374, 1996.

[32] M. Parter. ($\Delta + 1$) coloring in the congested clique model. In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, 2018.

[33] M. Parter and H.-H. Su. ($\Delta + 1$) coloring in $O(\log^* \Delta)$ congested-clique rounds. In *Proceedings of the International Symposium on Distributed Computing (DISC)*, 2018.

[34] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.

[35] J. Schneider and R. Wattenhofer. A new technique for distributed symmetry breaking. In *Proceedings 29th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 257–266, 2010.

[36] V. H. Vu. A general upper bound on the list chromatic number of locally sparse graphs. *Combinatorics, Probability & Computing*, 11(1):103–111, 2002.

A   Concentration Bounds

We make use of the following standard tail bounds [13]. Let X be binomially distributed with parameters (n, p), i.e., it is the sum of n independent 0-1 variables with mean p. We have the following bound on the lower tail of X:

$$\Pr[X \leq t] \leq \exp\left(\frac{-(\mu - t)^2}{2\mu}\right), \quad \text{where } t < \mu = np.$$

Chernoff bounds also hold when X is the sum of n *negatively correlated* 0-1 random variables [14, 13] with mean p, i.e., total independent is not required. We use a bound on the upper tail of X with mean $\mu = np$.

$$\Pr[X \geq (1 + \delta)\mu] \leq \begin{cases} \exp\left(\frac{-\delta^2 \mu}{3}\right) & \text{if } \delta \in [0, 1] \\ \exp\left(\frac{-\delta \mu}{3}\right) & \text{if } \delta > 1. \end{cases}$$

Consider the scenario where $X = \sum_{i=1}^{n} X_i$, and each X_i is an independent random variable bounded by the interval $[a_i, b_i]$. Let $\mu = \mathrm{E}[X]$. Then we have the following concentration bound (Hoeffding's inequality) [23].

$$\Pr[X \geq (1+\delta)\mu] \leq \exp\left(\frac{-2(\delta\mu)^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

B Proof of Lemma 3

In this section, we prove Lemma 3. Fix a constant parameter $p \in (0, 1/4)$. The procedure OneShotColoring is a simple $O(1)$-round coloring procedure that breaks ties by ID. We orient each edge $\{u, v\}$ from v to u if $\mathrm{ID}(u) < \mathrm{ID}(v)$. That is, $N_{\mathrm{out}}(v) = \{u \in N(v) \mid \mathrm{ID}(u) < \mathrm{ID}(v)\}$. We assume that each vertex v is associated with a palette $\Psi(v)$ of size $\Delta + 1$, and this is used implicitly in the proofs of the lemmas in this section.

Procedure OneShotColoring.

1. Each uncolored vertex v decides to participate independently with probability p.

2. Each participating vertex v selects a color $c(v)$ from its palette $\Psi(v)$ uniformly at random.

3. A participating vertex v successfully colors itself if $c(v)$ is not chosen by any vertex in $N_{\mathrm{out}}(v)$.

After OneShotColoring, each vertex v removes all colors from $\Psi(v)$ that are taken by some neighbor $u \in N(v)$. The number of *excess colors* at v is the size of v's remaining palette minus the number of uncolored neighbors of v. We prove one part of Lemma 3 by showing that after a call to OneShotColoring, the number of excess colors at any ϵ-sparse v is $\Omega(\epsilon^2 \Delta)$, with probability $1 - \exp(-\Omega(\epsilon^2 \Delta))$. The rest of this section constitutes a proof of Lemma 3.

Consider an execution of OneShotColoring with any constant $p \in (0, 1/4)$. Recall that we assume $1/\epsilon \geq K$, for some large enough constant K. Let v be an ϵ-sparse vertex. Define the following two numbers.

$f_1(v)$: number of vertices $u \in N(v)$ that successfully color themselves by some $c \notin \Psi(v)$.

$f_2(v)$: number of colors $c \in \Psi(v)$ such that at least two vertices in $N(v)$ successfully color themselves c.

It is clear that $f_1(v) + f_2(v)$ is a lower bound on the number of excess colors at v after OneShotColoring. Our first goal is to show that $f_1(v) + f_2(v) = \Omega(\epsilon^2 \Delta)$ with probability at least $1 - \exp(-\Omega(\epsilon^2 \Delta))$. We divide the analysis into two cases (Lemma 20 and Lemma 21), depending on whether $f_1(v)$ or $f_2(v)$ is likely to be the dominant term. For any v, the preconditions of either Lemma 20 or Lemma 21 are satisfied. Our second goal is to show that for each vertex v of degree at least 0.9Δ, with high probability, at least $(1 - 1.1p)|N(v)| > \Delta/2$ neighbors of v remain uncolored after after OneShotColoring. This is done in Lemma 22.

Lemmas 18 and 19 establish some generally useful facts of OneShotColoring, which are used in the proofs of Lemma 20 and 21.

Lemma 18. *Let Q be any set of colors, and let S be any set of vertices with size at most 2Δ. The number of colors in Q that are selected by some vertices in S is less than $|Q|/2$ with probability at least $1 - \exp(-\Omega(|Q|))$.*

Proof. Let E_c denote the event that color c is selected by at least one vertex in S. Then $\Pr[E_c] \leq \frac{p|S|}{\Delta+1} < 2p < 1/2$, since $p < 1/4$ and $|S| \leq 2\Delta$. Moreover, the collection of events $\{E_c\}$ are negatively correlated [14].

Let X denote the number of colors in Q that are selected by some vertices in S. By linearity of expectation, $\mu = \mathrm{E}[X] < 2p \cdot |Q|$. We apply a Chernoff bound with $\delta = \frac{(1/2)-2p}{2p}$. Recall that $0 < p < 1/4$, and so $\delta > 0$. For the case of $\delta \in [0, 1]$, we have:

$$\Pr[X \geq (1+\delta)\mu = |Q|/2] \leq \exp(-\delta^2\mu/3) = \exp(-\Omega(|Q|)).$$

Similarly, if $\delta > 1$, we still have:

$$\Pr[X \geq (1+\delta)\mu = |Q|/2] \leq \exp(-\delta\mu/3) = \exp(-\Omega(|Q|)). \qquad \square$$

Lemma 19. *Fix a sufficiently small $\epsilon > 0$. Consider a set of vertices $S = \{u_1, \ldots, u_k\}$ with cardinality $\epsilon\Delta/2$. Let Q be a set of colors such that each $u_i \in S$ satisfies $|\Psi(u_i) \cap Q| \geq (1 - \epsilon/2)(\Delta + 1)$. Moreover, each $u_i \in S$ is associated with a vertex set R_i such that (i) $S \cap R_i = \emptyset$, and (ii) $|R_i| \leq 2\Delta$. Then, with probability at least $1 - \exp(-\Omega(\epsilon^2\Delta))$, there are at least $p\epsilon\Delta/8$ vertices $u_i \in S$ such that the color c selected by u_i satisfies (i) $c \in Q$, and (ii) c is not selected by any vertex in $R_i \cup S \setminus \{u_i\}$.*

Proof. Define $Q_i = \Psi(u_i) \cap Q$. We call a vertex u_i *happy* if u_i selects some color $c \in Q$ and c is not selected by any vertex in $R_i \cup S \setminus \{u_i\}$. Define the following events.

E_i^{good}: u_i selects a color $c \in Q_i$ such that c is not selected by any vertices in R_i.

E_i^{bad}: the number of colors in Q_i that are selected by some vertices in R_i is at least $|Q_i|/2$.

E_i^{repeat}: the color selected by u_i is also selected by some vertices in $\{u_1, \ldots, u_{i-1}\}$.

Let X_i be the indicator random variable that *either* E_i^{good} or E_i^{bad} occurs, and let $X = \sum_{i=1}^k X_i$. Let Y_i be the indicator random variable that E_i^{repeat} occurs, and let $Y = \sum_{i=1}^k Y_i$. *Assuming that E_i^{bad} does not occur* for each $i \in [1, k]$, it follows that $X - 2Y$ is a lower bound on the number of happy vertices. Notice that by Lemma 18, $\Pr[E_i^{\text{bad}}] = \exp(-\Omega(|Q_i|)) = \exp(-\Omega(\Delta))$. Thus, assuming that no E_i^{bad} occurs merely distorts our probability estimates by a negligible $\exp(-\Omega(\Delta))$. We prove concentration bounds on X and Y, which together imply the lemma.

We show that $X \geq p\epsilon\Delta/7$ with probability $1 - \exp(-\Omega(\epsilon\Delta))$. It is clear that

$$\Pr[X_i = 1] \geq \Pr[E_i^{\text{good}} \mid \overline{E_i^{\text{bad}}}] \geq \frac{p \cdot |Q_i|/2}{\Delta + 1} \geq \frac{p(1 - \epsilon/2)}{2} > \frac{p}{3}.$$

Moreover, since $\Pr[X_i = 1 \mid E_i^{\text{bad}}] = 1$, the above inequality also holds, when conditioned on *any* colors selected by vertices in R_i. Thus, $\Pr[X \leq t]$ is upper bounded by $\Pr[\text{Binomial}(n', p') \leq t]$ with $n' = |S| = \epsilon\Delta/2$ and $p' = \frac{p}{3}$. We set $t = p\epsilon\Delta/7$. Notice that $n'p' = p\epsilon\Delta/6 > t$. Thus, according to a Chernoff bound on the binomial distribution, $\Pr[X \leq t] \leq \exp(\frac{-(n'p'-t)^2}{2n'p'}) = \exp(-\Omega(\epsilon\Delta))$.

We show that $Y \leq p\epsilon^2\Delta/2$ with probability $1 - \exp(-\Omega(\epsilon^2\Delta))$. It is clear that $\Pr[Y_i = 1] \leq \frac{p(i-1)}{\Delta+1} \leq \frac{p\epsilon}{2}$, even if we condition on arbitrary colors selected by vertices in $\{u_1, \ldots, u_{i-1}\}$. We have $\mu = E[Y] \leq \frac{p\epsilon}{2} \cdot |S| = \frac{p\epsilon^2\Delta}{4}$. Thus, by a Chernoff bound (with $\delta = 1$), $\Pr[Y \geq p\epsilon^2\Delta/2] \leq \Pr[Y \geq (1 + \delta)\mu] \leq \exp(-\delta^2\mu/3) = \exp(-\Omega(\epsilon^2\Delta))$.

To summarize, with probability at least $1 - \exp(-\Omega(\epsilon^2\Delta))$, we have $X - 2Y \geq p\epsilon\Delta/7 - 2p\epsilon^2\Delta/2 > p\epsilon\Delta/8$. $\qquad\square$

Lemma 20 considers the case when a large fraction of v's neighbors are likely to color themselves with colors outside the palette of v, and therefore be counted by $f_1(v)$. This lemma holds regardless of whether v is ϵ-sparse or not.

Lemma 20. *Suppose that there is a subset $S \subseteq N(v)$ such that $|S| = \epsilon\Delta/5$, and for each $u \in S$, $|\Psi(u) \setminus \Psi(v)| \geq \epsilon(\Delta + 1)/5$. Then $f_1(v) \geq \frac{p\epsilon^2\Delta}{100}$ with probability at least $1 - \exp(-\Omega(\epsilon^2\Delta))$.*

Proof. Let $S = (u_1, \ldots, u_k)$ be sorted in increasing order by ID. Define $R_i = N_{\text{out}}(u_i)$, and $Q_i = \Psi(u_i) \setminus \Psi(v)$. Notice that $|Q_i| \geq \epsilon\Delta/5$. Define the following events.

E_i^{good}: u_i selects a color $c \in Q_i$ and c is not selected by any vertex in R_i.

E_i^{bad}: the number of colors in Q_i that are selected by vertices in R_i is more than $|Q_i|/2$.

Let X_i be the indicator random variable that either E_i^{good} or E_i^{bad} occurs, and let $X = \sum_{i=1}^k X_i$. Given that the events E_i^{bad} for all $i \in [1, k]$ do not occur, we have $X \leq f_1(v)$, since if E_i^{good} occurs, then u_i successfully colors itself by some color $c \notin \Psi(v)$. By Lemma 18, $\Pr[E_i^{\text{bad}}] = \exp(-\Omega(|Q_i|)) = \exp(-\Omega(\epsilon\Delta))$. Thus, up to this negligible error, we can assume that E_i^{bad} does not occur, for each $i \in [1, k]$.

We show that $X \geq \epsilon^2\Delta/100$ with probability $1 - \exp(-\Omega(\epsilon^2\Delta))$. It is clear that $\Pr[X_i = 1] \geq \Pr[E_i^{\text{good}} \mid \overline{E_i^{\text{bad}}}] \geq \frac{p|Q_i|/2}{\Delta+1} \geq \frac{p\epsilon}{10}$, and this inequality holds even when conditioning on *any* colors selected by vertices in R_i and $\bigcup_{1 \leq j < i} R_j \cup \{u_j\}$ (since $S = (u_1, \ldots, u_k)$ is sorted in increasing order by

ID, $u_i \notin R_j = N_{\text{out}}(u_j)$ for any $1 \le j < i$). Thus, $\Pr[X \ge t]$ is upper bounded by $\Pr[\text{Binomial}(n', p') \le t]$ with $n' = |S| = \epsilon\Delta/5$ and $p' = \frac{p\epsilon}{10}$. We set $t = \frac{n'p'}{2} = \frac{p\epsilon^2\Delta}{100}$. Thus, according to a lower tail of the binomial distribution, $\Pr[X \le t] \le \exp\left(\frac{-(n'p'-t)^2}{2n'p'}\right) = \exp(-\Omega(\epsilon^2\Delta))$. $\qquad \square$

Lemma 21 considers the case that many pairs of neighbors of v are likely to color themselves the same color, and contribute to $f_2(v)$. Note that any ϵ-sparse vertex that does not satisfy the preconditions of Lemma 20 *does* satisfy the preconditions of Lemma 21.

Lemma 21. *Let v be an ϵ-sparse vertex. Suppose that there is a subset $S \subseteq N(v)$ such that $|S| \ge (1 - \epsilon/5)\Delta$, and for each $u \in S$, $|\Psi(u) \cap \Psi(v)| \ge (1 - \epsilon/5)(\Delta + 1)$. Then $f_2(v) \ge p^3\epsilon^2\Delta/2000$ with probability $1 - \exp(-\Omega(\epsilon^2\Delta))$.*

Proof. Let $S' = \{u_1, \ldots, u_k\}$ be any subset of S such that (i) $|S'| = \frac{p\epsilon\Delta}{100}$, (ii) for each $u_i \in S'$, there exists a set $S_i \subseteq S \setminus (S' \cup N(u_i))$ of size $\frac{\epsilon\Delta}{2}$. The existence of $S', S_1, \ldots, S_k$ is guaranteed by the ϵ-sparseness of v. In particular, S must contain at least $\epsilon\Delta - \epsilon\Delta/5 > p\epsilon\Delta/100 = |S'|$ *non*-friends of v, and for each such non-friend $u_i \in S'$, we have $|S \setminus (S' \cup N(u_i))| \ge \Delta(1 - \epsilon/5 - p\epsilon/100 - (1 - \epsilon)) > \epsilon\Delta/2$.

Order the set $S' = \{u_1, \ldots, u_k\}$ in increasing order by vertex ID. Define $R_i = \{u_1, \ldots, u_{i-1}\} \cup N_{\text{out}}(u_i)$, and $Q_i = \Psi(u_i) \cap \Psi(v)$. Define Q_i^{good} as the subset of colors $c \in Q_i$ such that c is selected by some vertex $w \in S_i$, but c is not selected by any vertex in $(N_{\text{out}}(w) \cup N_{\text{out}}(u_i)) \setminus S'$. Define the following events.

E_i^{good}: u_i selects a color $c \in Q_i^{\text{good}}$.

E_i^{bad}: the number of colors in Q_i^{good} is less than $p\epsilon\Delta/8$.

E_i^{repeat}: the color selected by u_i is also selected by some vertices in $\{u_1, \ldots, u_{i-1}\}$.

Let X_i be the indicator random variable that either E_i^{good} or E_i^{bad} occurs, and let $X = \sum_{i=1}^{k} X_i$. Let Y_i be the indicator random variable that E_i^{repeat} occurs, and let $Y = \sum_{i=1}^{k} Y_i$. Suppose that E_i^{good} occurs, and the color c selected by u_i is not selected by any vertex in $S \setminus \{u_i\}$. Then there must exist a vertex $w \in S_i$ such that both u_i and w successfully color themselves c. Notice that w and u_i are not adjacent. Thus, $X - Y \le f_2(v)$, given that E_i^{bad} does not occur, for each $i \in [1, k]$. Notice that $\Pr[E_i^{\text{bad}}] = \exp(-\Omega(\epsilon^2\Delta))$ (by Lemma 19 and the definition of Q_i^{good}), and thus indeed we can assume that E_i^{bad} does not occur. In what follows, we prove concentration bounds on X and Y, which together imply the lemma.

We show that $X \ge \frac{p^3\epsilon^2\Delta}{1000}$ with probability $1 - \exp(-\Omega(\epsilon^2\Delta))$. It is clear that $\Pr[X_i = 1] \ge p \cdot \frac{p\epsilon\Delta/8}{\Delta+1} > \frac{p^2\epsilon}{8}$, regardless of the colors selected by vertices in R_i. Thus, $\Pr[X \le t]$ is upper bounded by $\Pr[\text{Binomial}(n', p') \le t]$ with $n' = |S'| = \frac{p\epsilon\Delta}{100}$ and $p' = \frac{p^2\epsilon}{8}$. We set $t = \frac{p^3\epsilon^2\Delta}{1000} < n'p'$. According to a tail bound of binomial distribution, $\Pr[X \le t] \le \exp(\frac{-(n'p'-t)^2}{2n'p'}) = \exp(-\Omega(\epsilon^2\Delta))$.

We show that $Y \le \frac{p^3\epsilon^2\Delta}{2000}$ with probability $1 - \exp(-\Omega(\epsilon^2\Delta))$. It is clear that $\Pr[Y_i = 1] \le p \cdot \frac{(i-1)}{\Delta+1} \le \frac{p^2\epsilon}{100}$ regardless of the colors selected by vertices in $\{u_1, \ldots, u_{i-1}\}$. We have $\mu = \mathrm{E}[Y] \le \frac{p^2\epsilon}{100} \cdot |S'| = \frac{p^3\epsilon^2\Delta}{10,000}$. Thus, by a Chernoff bound (with $\delta = 4$), $\Pr[Y \ge \frac{p^3\epsilon^2\Delta}{2000}] \le \Pr[Y \ge (1 + \delta)\mu] \le \exp(-\delta\mu/3) = \exp(-\Omega(\epsilon^2\Delta))$.

To summarize, with probability at least $1 - \exp(-\Omega(\epsilon^2\Delta))$, we have $X - Y \ge p^3\epsilon^2\Delta/1000 - p^3\epsilon^2\Delta/2000 > p^3\epsilon^2\Delta/2000$. $\qquad \square$

Lemma 22. *The number of vertices in $N(v)$ that remain uncolored after OneShotColoring is at least $(1 - 1.1p)|N(v)|$, with probability at least $1 - \exp(-\Omega(|N(v)|))$.*

Proof. Let X be the number of vertices in $N(v)$ participating in OneShotColoring. It suffices to show that $X \le 1.1p|N(v)|$ with probability $1 - \exp(-\Omega(|N(v)|))$. Since a vertex participates with probability p,

$$\Pr[X \ge (1 + 0.1)p|N(v)|] \le \exp\left(-\frac{(0.1)^2 p|N(v)|}{3}\right) = \exp(-\Omega(|N(v)|))$$

by Chernoff bound with $\delta = 0.1$. $\qquad \square$