SIAM J. COMPUT.
Vol. 49, No. 3, pp. 497–539

DISTRIBUTED $(\Delta + 1)$-COLORING VIA ULTRAFAST GRAPH SHATTERING*

YI-JUN CHANG†, WENZHENG LI‡, AND SETH PETTIE§

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 $O(\mathsf{Det}_d(\operatorname{poly}\log n)) = O(\operatorname{poly}(\log\log n))$ 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 [*J. ACM*, 65 (2018), 19] with complexity $O(\sqrt{\log\Delta}+\log\log n+\mathsf{Det}_d(\operatorname{poly}\log n)) = O(\sqrt{\log n})$. Unless Δ is small, it is also faster than the best known deterministic algorithm of Fraigniaud, Heinrich, and Kosowski [*Proceedings of the* 57*th Annual IEEE Symposium on Foundations of Computer Science* (FOCS), 2016] and Barenboim, Elkin, and Goldenberg [*Proceedings of the* 38*th Annual ACM Symposium on Principles of Distributed Computing* (PODC), 2018], with complexity $O(\sqrt{\Delta\log\Delta}\log^*\Delta + \log^* n)$. Our algorithm's running time is *syntactically* very similar to the $\Omega(\mathsf{Det}(\operatorname{poly}\log n))$ lower bound of Chang, Kopelowitz, and Pettie [*SIAM J. Comput.*, 48 (2019), pp. 122–143], where $\mathsf{Det}(n')$ is the deterministic complexity of $(\Delta+1)$-list coloring on n'-vertex graphs. Although distributed coloring has been actively investigated for 30 years, the best deterministic algorithms for $(\deg+1)$- and $(\Delta+1)$-list coloring (that depend on n' but not Δ) use a black-box application of network decompositions. The recent deterministic network decomposition algorithm of Rozhoň and Ghaffari [*Proceedings of the* 52*nd Annual ACM SIGACT Symposium on Theory of Computing* (STOC), 2020] implies that $\mathsf{Det}_d(n')$ and $\mathsf{Det}(n')$ are both $\operatorname{poly}(\log n')$. Whether they are asymptotically equal is an open problem.

Key words. distributed algorithm, local model, graph coloring

AMS subject classifications. 05C85, 68W15

DOI. 10.1137/19M1249527

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 [38, 42, 5, 44, 40, 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 $O(\operatorname{poly}(\Delta)+\log^* n)$ time (optimal for $\Delta = O(1)$), and all can be solved using *network decompositions* [5, 43]. A recent breakthrough in network decompositions by Rozhoň and Ghaffari [48] shows that all four problems can be solved in $\operatorname{poly}(\log n)$ time deterministically. Until recently, it was actually consistent with known results that these problems had exactly the same complexity.

Kuhn, Moscibroda, and Wattenhofer [36] proved that the "independent set" problems (MIS and maximal matching) require $\Omega\big(\min\big\{\frac{\log\Delta}{\log\log\Delta}, \sqrt{\frac{\log n}{\log\log n}}\big\}\big)$ time, with

*Received by the editors March 12, 2019; accepted for publication (in revised form) March 2, 2020; published electronically May 19, 2020. A preliminary version of this paper appeared in *Proceedings of the* 50*th Annual ACM Symposium on Theory of Computing* (STOC), 2018, pp. 445–456.
https://doi.org/10.1137/19M1249527
Funding: This work was funded by NSF grants CCF-1514383, CCF-1637546, and CCF-1815316.
†Institute for Theoretical Studies, ETH Zürich, Zürich, 8092, Switzerland (yi-jun.chang@eth-its.ethz.ch).
‡Computer Science Department, Stanford University, Stanford, CA 94305 (wzli@stanford.edu).
§EECS, University of Michigan, Ann Arbor, MI 48109 (pettie@umich.edu).

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 [38]. Very recently, Balliu et al. [6] proved that maximal matching and MIS require $\Omega\big(\min\big\{\Delta, \frac{\log n}{\log\log n}\big\}\big)$ time deterministically, which strictly improves on the Kuhn–Moscibroda–Wattenhofer (KMW) bounds, and that randomized algorithms for maximal matching and MIS require $\Omega\big(\min\big\{\Delta, \frac{\log\log n}{\log\log\log n}\big\}\big)$ time, which is stronger than KMW when $\Delta < \log n$ but weaker when $\Delta \gg \log n$.

The KMW lower bounds [36] cannot be extended to the canonical coloring problems, or to variants of MIS like $(2,t)$-ruling sets, for $t \geq 2$ [14, 13, 28]. Elkin, Pettie, and Su [25] proved that $(2\Delta-1)$-list edge coloring can be solved by a randomized algorithm in $O(\log\log n + \mathsf{Det}(\operatorname{poly}\log n)) = \operatorname{poly}(\log\log n)$ time, which shows that neither the $\Omega\big(\frac{\log\Delta}{\log\log\Delta}\big)$ nor the $\Omega\big(\sqrt{\frac{\log n}{\log\log n}}\big)$ KMW lower bound applies to this problem. Here $\mathsf{Det}(n')$ represents the *deterministic* complexity of the problem in question on n'-vertex graphs. Improving on [13, 49], Harris, Schneider, and Su [32] proved a similar separation for $(\Delta+1)$-vertex coloring. Their randomized algorithm solves the problem in

$$O(\sqrt{\log\Delta} + \log\log n + \mathsf{Det}_d(\operatorname{poly}\log n)) = O(\sqrt{\log n})$$

time, where Det_d is the complexity of $(\deg+1)$-list coloring.

The "$\mathsf{Det}(\operatorname{poly}\log n)$"-type terms in the running times of [25, 32] are a consequence of the *graph shattering* technique applied to distributed symmetry breaking. Barenboim et al. [13] 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 $\operatorname{poly}\log(n)$, which can then be solved with the best available deterministic algorithm.[1] Later, Chang, Kopelowitz, and Pettie [20] 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 Chang–Kopelowitz–Pettie (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.

- $O(\log\Delta + \operatorname{poly}(\log\log n))$ for MIS (Rand. due to [28] and Det. to [48]),
- $O(\sqrt{\log\Delta} + \operatorname{poly}(\log\log n))$ for $(\Delta+1)$-vertex coloring (Rand. due to [32] and Det. to [48]),
- $O(\log\Delta + (\log\log n)^3)$ for maximal matching (Rand. due to [13] and Det. to [26]),
- $O((\log\log n)^{3+o(1)})$ for $(2\Delta-1)$-edge coloring (Rand. due to [25] and Det. to [31]).

In each, the term that depends on n is the complexity of the best deterministic algorithm, scaled down to $\operatorname{poly}\log(n)$-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

[1] In the case of MIS, the subproblems actually have size $\operatorname{poly}(\Delta)\log n$, 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 $\operatorname{poly}\log(n)$ size. See [13, section 3] or [28, section 4] for more discussion of this.

[2] See [42, 22, 20] for the formal definition of the class of locally checkable labeling (LCL) problems.

"Δ." For example, a recent line of research has improved the complexity of $(\Delta+1)$-coloring in terms of Δ, from $O(\Delta + \log^* n)$ [12], to $\tilde{O}(\Delta^{3/4}) + O(\log^* n)$ [8], to the state-of-the-art bound of $O(\sqrt{\Delta \log \Delta} \log^* \Delta + \log^* n)$ due to Fraigniaud, Heinrich, and Kosowski [27] and Barenboim, Elkin, and Goldenberg [11]. A recent algorithm of Kuhn [35] solves $(\Delta+1)$-list coloring in $2^{O(\sqrt{\log \Delta})} \log n$ time; i.e., the dependence on Δ is better than [27, 11], but the dependence on n is worse. These improvements do not have consequences for randomized coloring algorithms using graph shattering [13, 32] since we can only assume $\Delta = (\log n)^{\Omega(1)}$ in the shattered instances.[3] See Table 1 for a summary of lower and upper bounds for distributed $(\Delta+1)$-list coloring in the LOCAL model.

In this paper, we prove that $(\Delta+1)$-list coloring can be solved, w.h.p., in just $O(\mathsf{Det}_d(\text{poly}\log n))$ time. Our algorithm's performance is best contrasted with the $\Omega(\mathsf{Det}(\text{poly}\log n))$ randomized lower bound of [20], 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, in the regime we care about (deterministic algorithms that depend on n but not Δ), the state-of-the-art in $(\deg+1)$- and $(\Delta+1)$-list coloring has not changed much in 30 years: the algorithms begin by (1) computing a generic network decomposition [5, 39, 43, 48], and then (2) applying it to simulate the sequential greedy coloring algorithm. So long as this is the template for the best deterministic vertex coloring algorithms, it will be nearly impossible to prove $(\Delta+1)$-coloring is *strictly* easier than $(\deg+1)$-coloring.

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 identifier $\text{ID}(v)$, and global graph parameters $n = |V|$ and $\Delta = \max_{v \in V} \deg(v)$. Refer to [38, 47] for more on the LOCAL model and variants.

We write $N(v)$ to denote the set of neighbors of the vertex v. For directed graphs, $N_{\text{out}}(v)$ is the set of out-neighbors of v. We write $N^k(v) = \{u \in V \mid \text{dist}(u, v) \le k\}$ to denote the set of vertices within distance k of v. Note that $v \in N^k(v)$ for any $k \ge 0$.

In the $(\Delta+1)$-list coloring problem, each vertex v also has a palette $\Psi(v)$ of allowable colors, with $|\Psi(v)| \ge \Delta + 1$. 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 a message of unbounded size 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 $N(v)$.

In this paper, we prove that $(\Delta+1)$-list coloring can be solved, w.h.p., in $O(\mathsf{Det}_d(\text{poly}\log n))$ time. Intellectually, our algorithm builds on a succession of breakthroughs by Schneider and Wattenhofer [49], Barenboim et al. [13], Elkin, Pettie, and Su, [25], and Harris, Schneider, and Su [32], which we shall now review.

2.1. Fast coloring using excess colors. Schneider and Wattenhofer [49] 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

[3] Strictly speaking, the algorithm of Kuhn [35] may be more desirable than that of [48] in a graph shattering–type coloring algorithm. For sufficiently small Δ, $2^{O(\sqrt{\log \Delta})} \log\log n$ is better than $\text{poly}(\log\log n)$.

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* [32] *and ours, also solve the* $(\deg +1)$*-list coloring problem.*

	Randomized		Deterministic	
Upper bounds	$O(\mathsf{Det}_d(\operatorname{poly}\log n))$	new	$O(\operatorname{poly}\log n)$	[48]
	$O(\sqrt{\log \Delta} + \log\log n + \mathsf{Det}_d(\operatorname{poly}\log n))$	[32]	$2^{O(\sqrt{\log \Delta})} \cdot \log n$	[35]
	$O(\log \Delta + \mathsf{Det}_d(\operatorname{poly}\log n))$	[13]	$O(\sqrt{\Delta \log \Delta} \log^* \Delta + \log^* n)$	[27, 11]
	$O(\log \Delta + \sqrt{\log n})$	[49]	$O(\sqrt{\Delta} \log^{5/2} \Delta + \log^* n)$	[27]
	$O(\Delta \log\log n)$	[37]	$O(\Delta^{3/4} \log \Delta + \log^* n)$	[8]
	$O(\log n)$	[40, 1, 34]	$O(\Delta + \log^* n)$	[12]
			$O(\Delta \log \Delta + \log^* n)$	[37]
			$O(\Delta \log n)$	[5]
			$O(\Delta^2 + \log^* n)$	[29, 38]
			$O(\Delta^{O(\Delta)} + \log^* n)$	[30]
			$2^{O(\sqrt{\log n})}$	[44]
			$2^{O(\sqrt{\log n \log\log n})}$	[5]
Lower bounds	$\Omega(\log^* n)$	[42]	$\Omega(\log^* n)$	[38]
	$\Omega(\mathsf{Det}(\sqrt{\log n}))$	[20]		

$\Delta \geq \log^{1+\gamma} n$ and the palette size is $(1+\epsilon)\Delta$, vertex coloring can be solved w.h.p. in just $O(\log^* n)$ time [49, Corollary 14]. The emergence of this log-star behavior in [49] is quite natural. Consider the case where the palette size of each vertex is at least $k\Delta$, where $k \geq 2$. 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 in the uncolored part of the graph after this coloring procedure is at most $\Delta' = \Delta/2^{k/2}$. In contrast, the number of *excess colors*, i.e., the size of the current available palette at v (i.e., the initial palette excluding the colors already taken by the neighbors of v) minus the number of uncolored neighbors, is nondecreasing over time. It is at least $(k-1)\Delta = (k-1)2^{k/2}\Delta'$. Intuitively, repeating the above procedure for $O(\log^* n)$ rounds suffices to color all vertices.

Similar ideas have also been applied in other papers [49, 25, 20]. However, for technical reasons, we cannot directly apply the results in these papers. The main difficulty in our setting is that we need to deal with *oriented* graphs with widely varying out-degrees, palette sizes, and excess colors; the guaranteed number of excess colors at a vertex depends on its out-degree, *not* the global parameter Δ.

Lemma 2.1 summarizes the properties of our ultrafast coloring algorithm when each vertex has many excess colors; its proof appears in section 5. Recall that $\Psi(v)$ denotes the palette of v, so $|\Psi(v)| - \deg(v)$ is the number of excess colors at v. Also recall that $N_{\text{out}}(v)$ denotes the set of out-neighbors of v in a directed graph.

LEMMA 2.1. *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* $\sum_{u \in N_{\text{out}}(v)} 1/p_u \leq 1/C$. *Let* $d^\star$ *be the maximum out-degree of the graph. There is an algorithm that takes* $O(1 + \log^* p^\star - \log^* C)$ *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 2.1. 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 $N_{\text{out}}(v)$. For each color c selected by v, the probability that c is also selected by some vertex in $N_{\text{out}}(v)$ is $(C/2)\sum_{u \in N_{\text{out}}(v)} 1/p_u \leq 1/2$. 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 $O(1 + \log^* p^\star - \log^* C)$ rounds. Lemma 2.2 is a more user-friendly version of Lemma 2.1 for simpler situations.

LEMMA 2.2. *Suppose* $|\Psi(v)| \geq (1+\rho)\Delta$ *for each vertex* v, *and let* $\rho = \Omega(1)$. *There is an algorithm that takes* $O(1 + \log^* \Delta - \log^* \rho)$ *time and achieves the following. Each vertex* v *remains uncolored with probability at most* $\exp(-\Omega(\sqrt{\rho\Delta}))$. *This is true even if the random bits generated outside a constant radius around* v *are determined adversarially.*

Proof. We apply Lemma 2.1. Orient the graph arbitrarily, and then set $p_v = \rho\Delta$ for each v. Use the parameters $C = \rho$, $p^\star = \rho\Delta$, and $d^\star = \Delta$. The time complexity is $O(1 + \log^* p^\star - \log^* C) = O(1 + \log^* \Delta - \log^* \rho\})$. The failure probability is $\exp(-\Omega(\sqrt{p^\star})) + d^\star \exp(-\Omega(p^\star)) = \exp(-\Omega(\sqrt{\rho\Delta}))$. ∎

2.2. Gaining excess colors. Schneider and Wattenhofer [49] 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 [25] 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, 25, 41, 50], and the ϵ-*friends* from [32], defined formally as follows.

DEFINITION 2.3 (see [25]). *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.*

DEFINITION 2.4 (see [32]). *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 2.4. Lemma 2.5 shows that in $O(1)$ time we can make excess colors at all locally sparse vertices by coloring a subset of V.

LEMMA 2.5. *Consider the* $(\Delta+1)$*-list coloring problem. There is an* $O(1)$*-time algorithm that colors a subset of* V *such that the following are true for each* $v \in V$ *with* $\deg(v) \geq (5/6)\Delta$*:*

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

(ii) *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.*

The algorithm behind Lemma 2.5 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 *not* an edge. If $|\Psi(u) \cap \Psi(w)| = \Omega(\Delta)$, then the probability that both u and w are colored by the same color is $\Omega(1/\Delta)$, and this increases the number of excess colors at v by 1. Otherwise, we have $|(\Psi(u) \cup \Psi(w)) \setminus \Psi(v)| = \Omega(\Delta)$, and so with probability $\Omega(1)$ one of u and w successfully colors itself with a color not in $\Psi(v)$, and this also increases the number of excess colors at v by 1. Therefore, the expected number of excess colors created at v is at least $\Omega\big(\frac{\epsilon^2\Delta^2}{\Delta}\big) = \Omega(\epsilon^2\Delta)$.

Similar but slightly weaker lemmas were proved in [25, 32]. The corresponding lemma from [25] does not apply to *list* coloring, and the corresponding lemma from [32] 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 2.5 appears in Appendix B.

The notion of local sparsity is especially useful for addressing the $(2\Delta-1)$-edge coloring problem [25], 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 [32] 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 time on the order of

$$\sqrt{\log\Delta} + \log\log n + \mathsf{Det}_d(\operatorname{poly}\log n) \;=\; \sqrt{\log\Delta} + \operatorname{poly}(\log\log n) \;=\; \sqrt{\log n}.$$

We briefly overview each stage of their algorithm.

Coloring ϵ*-sparse vertices.* Using the excess colors, Harris, Schneider, and Su [32] showed that the ϵ-sparse set can be colored in $O(\log\epsilon^{-1} + \log\log n + \mathsf{Det}_d(\operatorname{poly}\log n))$ time using techniques of [25, 13]. More specifically, they applied the algorithm of [25, Corollary 4.1] using the $\epsilon'\Delta = \Omega(\epsilon^2\Delta)$ excess colors, i.e., $\epsilon' = \Theta(\epsilon^2)$. This takes $O\left(\log(\epsilon^{-1})\right) + T\big(n, O\big(\frac{\log^2 n}{\epsilon'}\big)\big)$ time, where $T(n', \Delta') = O(\log\Delta' + \log\log n' + \mathsf{Det}_d(\operatorname{poly}\log n'))$ is the time complexity of the $(\deg+1)$-list coloring algorithm of [13, Theorem 5.1] on n'-vertex graphs of maximum degree Δ'.

Coloring ϵ*-dense vertices.* For ϵ-dense vertices, Harris, Schneider, and Su [32] proved that by coordinating the coloring decisions within each dense component, it takes only $O(\log_{1/\epsilon}\Delta + \log\log n + \mathsf{Det}_d(\operatorname{poly}\log n))$ 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 $\epsilon = \exp(-\Theta(\sqrt{\log\Delta}))$.

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 applies the $(\deg +1)$-list coloring algorithm of [13, Theorem 5.1] to color the remaining vertices in $O(\log \Delta' + \log\log n + \mathsf{Det}_d(\text{poly}\log n)) = O(\log_{1/\epsilon} \Delta + \log\log n + \mathsf{Det}_d(\text{poly}\log n))$ 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, Schneider, and Su [32] 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 a vertex $v \in S$ is in conflict with an *external neighbor* of v is $O(\epsilon)$. Harris, Schneider, and Su [32] 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 $\Omega(\sqrt{\epsilon^{-1}})$ in each iteration.

2.4. New results. In this paper, we give a fast randomized algorithm for $(\Delta+1)$-list coloring. It is based on a hierarchical version of the Harris–Schneider–Su decomposition with $\log\log \Delta - O(1)$ levels determined by an increasing sequence of sparsity thresholds $(\epsilon_1, \ldots, \epsilon_\ell)$, with $\epsilon_i = \sqrt{\epsilon_{i+1}}$. Following [32], we begin with a single iteration of the *initial coloring step* (Lemma 2.5), in which a constant fraction of the vertices are colored. The guarantee of this procedure is that any vertex v at the ith layer (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 layer, with the hope that each will take roughly constant time. Recall that (i) any vertex v at the ith layer 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 $O\big(\log_{1/\epsilon_i} \frac{\Delta}{\epsilon_{i-1}^{2.5}\Delta}\big) = O(1)$ 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 2.2) should complete in very few additional steps.

It turns out that 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 blocks at layer i into *small*, *medium*, and *large* based chiefly on the block size and partitions the set of all blocks of all layers 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 and medium blocks, the palette size guarantee comes from the ordering of the blocks being processed; we will show that whenever a small or medium 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 coloring steps outlined above finish in $O(\log^* \Delta)$ time. The bottleneck procedure is the algorithm of Lemma 2.2, 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 either induces a bounded degree graph or is composed of $O(\text{poly}\log n)$-size components, w.h.p. The former type can be colored deterministically in $O(\log^* n)$ time and the latter in $\mathsf{Det}_d(\text{poly}\log n)$ time. In view of Linial's lower bound [38], we have $\mathsf{Det}_d(\text{poly}\log n) = \Omega(\log^* n)$ and the running time of our $(\Delta+1)$-list coloring algorithm is

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

Recent developments. After the initial publication of this work [21], our algorithm was adapted to solve $(\Delta+1)$-coloring in several other models of computation, namely the *congested clique*, the massively parallel computation (*MPC*) model, and the *centralized local computation* model [4, 45, 46, 17]. Chang et al. [17], improving [45, 46], showed that $(\Delta+1)$-coloring can be solved in the congested clique in $O(1)$ rounds, w.h.p. In the MPC model, Assadi, Chen, and Khanna [4] solve $(\Delta+1)$-coloring in $O(1)$ rounds using $\tilde{O}(n)$ memory per machine, whereas Chang et al. [17] solve it in $O(\log\log\log n)$ time with just $O(n^\epsilon)$ memory per machine. In the centralized local computation model, Chang et al. [17] proved that $(\Delta+1)$-coloring queries can be answered with just *polynomial* probe complexity $\Delta^{O(1)}\log n$.

Organization. In section 3, we define a hierarchical decomposition based on [32]. 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 2.1 analyzes the procedure $\mathsf{ColorBidding}$, which is a generalization of the Schneider–Wattenhofer coloring routing; it is proved in section 5.
- Lemma 2.5 shows that the procedure $\mathsf{OneShotColoring}$ creates many excess colors; it is proved in Appendix B.
- Lemmas 4.2–4.5 analyze two versions of an algorithm $\mathsf{DenseColoringStep}$, which is a generalization of the Harris–Schneider–Su routine [32] for coloring locally dense vertices; they are proved in section 6.

Appendix A reviews all of the standard concentration inequalities that we use.

3. Hierarchical decomposition.

In this section, we extend the work of Harris, Schneider, and Su [32] 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 for each vertex to decide whether it is ϵ-sparse or ϵ-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 w.r.t. S and V' and $a_S(v) = |S \setminus (N(v) \cup \{v\})|$ to be the *antidegree* of v w.r.t. 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 3.1 from [32], which summarizes key properties of almost cliques.

LEMMA 3.1 (see [32]). *Fix any $\epsilon < 1/5$. The following conditions are met for each ϵ-almost clique C and each vertex $v \in C$:*

(i) $\bar{d}_{C,V_\epsilon^{\mathsf{d}}}(v) \leq \epsilon\Delta$ *(small external degree w.r.t. ϵ-dense vertices).*
(ii) $a_C(v) < 3\epsilon\Delta$ *(small antidegree).*

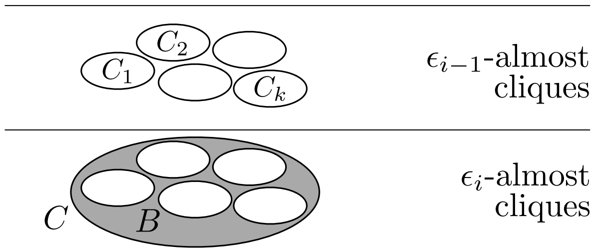

FIG. 1. *Almost cliques and blocks: the shaded region indicates a layer-i block B, and the hollow regions are those ϵ_{i-1}-almost cliques.*

(iii) $|C| \le (1 + 3\epsilon)\Delta$ *(small size, a consequence of* (ii)*).*
(iv) $\mathrm{dist}_G(u, v) \le 2$ *for each* $u, v \in C$ *(small weak diameter).*

Lemma 3.1(iv) implies that *any* sequential algorithm operating solely on C can be simulated in $O(1)$ rounds in the LOCAL model. The node in C with minimum ID can gather all the relevant information from C in two rounds of communication, compute the output of the algorithm locally, and disseminate these results in another two rounds of communication. For example, the DenseColoringStep algorithm (versions 1 and 2) presented in section 6 are nominally sequential algorithms but can be implemented in $O(1)$ distributed rounds.

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 $V^\star \subseteq V$, which, roughly speaking, are those left uncolored by the initial coloring procedure of Lemma 2.5 and also satisfy the two conclusions of Lemma 2.5(i)–(ii). The sequence $(\epsilon_1, \ldots, \epsilon_\ell)$ always adheres to Definition 3.2.

DEFINITION 3.2. *A sequence* $(\epsilon_1, \ldots, \epsilon_\ell)$ *is a valid* sparsity sequence *if the following conditions are met:*

- $\epsilon_i = \sqrt{\epsilon_{i-1}} = (\epsilon_1)^z$*, where* $z = 2^{-(i-1)}$*, and*
- $\epsilon_\ell \le 1/K$ *for some sufficiently large constant* K*.*

Layers. Define $V_1 = V^\star \cap V^{\mathsf{d}}_{\epsilon_1}$ and $V_i = V^\star \cap (V^{\mathsf{d}}_{\epsilon_i} \setminus V^{\mathsf{d}}_{\epsilon_{i-1}})$ for $i > 1$. Define $V_{\mathsf{sp}} = V^\star \cap V^{\mathsf{s}}_{\epsilon_\ell} = V^\star \setminus (V_1 \cup \cdots \cup V_\ell)$. 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 w.r.t. 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. List the ϵ_i-almost cliques arbitrarily as $(C_1, C_2, \ldots)$, and let $B_j = C_j \cap V_i$. Then $(B_1, B_2, \ldots)$ is a partition of V_i. Each $B_j \neq \emptyset$ is called a *layer-i block.* See Figure 1 for an illustration.

A layer-i block B is a *descendant* of a layer-i' block B', where $i < i'$, 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.

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

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

Large blocks. The set of large blocks is a maximal set of unrelated[4] 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 $V_{2+}^X = \bigcup_{i=2}^{\ell} V_i^X$ to be the set of all vertices of type X, excluding those in layer 1.

Overview of our algorithm. The decomposition and $\mathcal{T}$ are trivially computed in $O(1)$ rounds of communication. The first step of our algorithm is to execute an $O(1)$-round coloring procedure ($\mathsf{OneShotColoring}$) which colors a small constant fraction of the vertices in G; the relevant guarantees of this algorithm were stated in Lemma 2.5. Let $V^\star$ be the subset of uncolored vertices that, in addition, satisfy the conclusions of Lemma 2.5(i)–(ii). Once $V^\star$ is known, it can be partitioned into the following sets:

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

These are determined by the hierarchical decomposition w.r.t. a particular sparsity sequence $(\epsilon_1, \ldots, \epsilon_\ell)$.[5] We color the vertices of $V^\star \setminus V_{\mathsf{sp}}$ in six stages according to the ordering

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

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, as formalized by Lemma 3.3 below. 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 nontrivial 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.

LEMMA 3.3. *For each layer $i \in [1, \ell]$, the following are 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, due to the number of its (still uncolored) neighbors in medium and large blocks, and V_{sp}. 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.

[4]In other words, no two blocks are related by the *ancestor* relation.

[5]Note that the classification of vertices into small, medium, and large blocks can only be done *after* $\mathsf{OneShotColoring}$ 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 *and* uncolored by $\mathsf{OneShotColoring}$. Thus, whether the layer-i block in C is large-eligible depends on how many vertices are successfully colored.

Before proving Lemma 3.3, we first establish a useful property that constrains the structure of the block hierarchy $\mathcal{T}$. Intuitively, Lemma 3.4 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 descendants must be very small.

LEMMA 3.4. *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 $\sum_{j=1}^{l}|C_j|\le 2(3\epsilon_i+\epsilon_{i-1})\Delta$. 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 $l\ge 2$ 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 antidegree property of Lemma 3.1, 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. This means that C_1 *cannot* be a connected component in the graph formed by ϵ_{i-1}-dense vertices and ϵ_{i-1}-friend edges. ∎

Proof of Lemma 3.3. First consider the case of $v\in V_i^{\mathsf{M}}$. 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

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

Let C be the layer-k almost clique containing both B and B'. By Lemma 3.1, v has at most $3\epsilon_k\Delta$ nonneighbors in C, which, since $B'\subseteq C$, means that the number of neighbors of v in B' is at least

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

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

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

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

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

$A_3=$ the remaining neighbors.

To prove the lemma, it suffices to show that $|A_1|\ge\frac{\Delta}{4}$. Since $|A_1\cup A_2\cup A_3|\ge\frac{\Delta}{3}$, we need to prove $|A_2\cup A_3|\le\frac{\Delta}{12}$. We first bound $|A_3|$ and then $|A_2|$.

Note that v is ϵ_j-dense for $j\in[i,\ell]$, so, according to Lemma 3.1, v must have at least $(1-\epsilon_j)\Delta$ ϵ_j-friends. Let u be any neighbor of v not in an ancestor/descendant of B, which means that either (i) $u \in V_{\mathsf{sp}}$ or (ii) for some $j \in [i, \ell]$, v and u are in distinct ϵ_j-almost cliques. In case (i), u is counted in A_1. In case (ii), it follows that u *cannot* be an ϵ_j-friend of v. Since, by Lemma 3.1, v has at most $\epsilon_j \Delta$ ϵ_j-nonfriends,

$$|A_3| \le \sum_{j=i}^{\ell} \epsilon_j \Delta < 2\epsilon_\ell \Delta.$$

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,

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

If $i^\star = 0$, then $A_{2,\text{low}} = \emptyset$. Otherwise, by Lemma 3.4, the number of vertices in $A_{2,\text{low}}$ is at most $7\epsilon_{i^\star+1}\Delta \le 7\epsilon_i \Delta \le 7\epsilon_\ell \Delta$. 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. ∎

Remark 1. In the preliminary version of this paper [21], the algorithm for coloring locally dense vertices consisted of $O(\log^* \Delta)$ stages. In this paper, we improve the number of stages to $O(1)$. This improvement does not affect the overall asymptotic time in the LOCAL model, but it simplifies the algorithm and is critical to *adaptations* of our algorithm to models in which Linial's lower bound [38] does not apply, e.g., the congested clique [4, 17].

Remark 2. 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 $\frac{\Delta}{\log(1/\epsilon_i)}$, 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 for distributed symmetry breaking problems [13]. 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 $\mathsf{Det}_d(\text{poly}\log n)$ time.[6] More precisely, the emergence of the small components is due to the following lemma [13, 26]. A proof of this lemma can be found in [18, Lemma 1.2].

[6]A (deg +1)-list coloring algorithm applied to $n' = \text{poly}\log n$ size graphs requires $O(\log n')$-bit IDs. In $O(\mathsf{Det}_d(\text{poly}\log n))$ time, we can generate short, not necessarily distinct, IDs that are indistinguishable from distinct IDs; see [13, Remark 3.6] for the method.

LEMMA 4.1 (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 $(c'/c)\Delta^{2c} \log_\Delta n$.*

Lemma 4.1 obviously applies to randomized procedures that take c rounds. It also applies to $\omega(1)$-round procedures that are composed of a *series* of c-round experiments, where vertices that fail to satisfy some invariant are included in B immediately after the experiment. What is important is that the bound $\Pr[v \in B] \leq \Delta^{-3c}$ holds if an adversary is allowed to completely control how the series of experiments proceeds outside $N^c(v)$, so long as it cannot see the random bits generated inside $N^c(v)$.

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 $\ell = \log\log\Delta - O(1)$ 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 nondecreasing over time. In the first step of our coloring algorithm, we execute the algorithm of Lemma 2.5, 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 conditions:

- If v is ϵ_ℓ-dense, the number of uncolored neighbors of v is at least $\Delta/2$.
- If v is ϵ_i-sparse, v must have $\Omega(\epsilon_i^2\Delta)$ excess colors.

If either condition fails to hold, v is put in the set V_{bad}. We invoke the conditions of Lemma 2.5 only with $\epsilon \geq \epsilon_1 = \Delta^{-1/10}$. Thus, if $\Delta = \Omega(\log^2 n)$, then w.h.p. (i.e., $1 - 1/\mathrm{poly}(n)$), $V_{\mathsf{bad}} = \emptyset$. Otherwise, each component of V_{bad} must, by Lemma 4.1, 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. By definition, $V^\star$ is the set of all vertices that remain uncolored after the initial coloring step *and* are not put in V_{bad}. The partition $V^\star = V_{2+}^{\mathsf{S}} \cup V_1^{\mathsf{S}} \cup V_{2+}^{\mathsf{M}} \cup V_1^{\mathsf{M}} \cup V_{2+}^{\mathsf{L}} \cup V_1^{\mathsf{L}} \cup V_{\mathsf{sp}}$ is computed in $O(1)$ time. In this section, we show how we can color *most* of the vertices in $V_{2+}^{\mathsf{S}} \cup V_1^{\mathsf{S}} \cup V_{2+}^{\mathsf{M}} \cup V_1^{\mathsf{M}} \cup V_{2+}^{\mathsf{L}} \cup V_1^{\mathsf{L}}$, in that order, leaving a small portion of uncolored vertices.

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 3.3 is already met, then v has at least $\Delta/4 > \Delta/6$ excess colors w.r.t. V_{2+}^{S}.
- Suppose $|N(v) \cap V^\star| < \Delta/3$. One criterion for adding v to V_{bad} is that v is ϵ_ℓ-dense but has less than $\Delta/2$ uncolored neighbors after the initial coloring step. We know v is ϵ_ℓ-dense and not in V_{bad} (because it is in V_{2+}^{S}), so it must have had at least $\Delta/2$ uncolored neighbors after initial coloring. If $|N(v) \cap V^\star| < \Delta/3$, then at least $(\Delta/2 - \Delta/3) = \Delta/6$ of v's uncolored neighbors must have joined V_{bad}, which provide v with $\Delta/6$ 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 4.2 to color V_{2+}^{S} and V_{2+}^{M} and use Lemma 4.3 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 2.5). 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 4.2 (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 4.3 (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 $1 - \exp(-\Omega(\mathrm{poly}(\Delta)))$; all uncolored vertices in S join V_{bad}.*

The following lemmas consider large blocks. Lemma 4.4 colors V_{2+}^{L} and has guarantees similar to Lemma 4.2, whereas Lemma 4.5 colors nearly all of V_1^{L} and partitions the remaining uncolored vertices among three sets, R, X, and V_{bad}, with certain guarantees.

LEMMA 4.4 (large blocks; layer other than 1). *There is an $O(1)$-time algorithm that colors a subset of V_{2+}^{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 V_{2+}^{L} 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 $O(\mathrm{poly}\log n)$. However, if in a randomized procedure each vertex is added to V_{bad} with probability $1/\mathrm{poly}(\Delta)$, then the shattering lemma only guarantees that the size of each connected component of V_{bad} is $O(\mathrm{poly}(\Delta)\log n)$, which is not necessarily $\mathrm{poly}\log n$. This explains why Lemma 4.5 has two types of guarantees.

LEMMA 4.5 (large blocks; layer 1). *Let c be a sufficiently large constant. Then there is a constant time (independent of c) algorithm that colors a subset of V_1^{L} while satisfying one of the following cases:*

- *The uncolored vertices of V_1^{L} are partitioned among R or V_{bad}. The graph induced by R has degree $O(c^2)$; each vertex joins V_{bad} with probability $\Delta^{-\Omega(c)}$.*
- *If $\Delta > \log^{\alpha c} n$, where $\alpha > 0$ is some universal constant, then the uncolored vertices of V_1^{L} are partitioned among R and X, where the graph induced by R has degree $O(c^2)$ and the components induced by X have size $\log^{O(c)} n$, w.h.p.*

In our $(\Delta + 1)$-list coloring algorithm, we apply Lemmas 4.2, 4.3, 4.4, and 4.5 to color the vertices in $V^\star \setminus V_{\mathsf{sp}}$, 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 4.5 generates a leftover uncolored subset R which induces a constant-degree subgraph and (in case $\Delta > \log^{\Theta(c)} n$) 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}. All vertices in X are colored deterministically in $\mathsf{Det}_d(\text{poly}\log n)$ time; the vertices in R are colored deterministically in $O(\text{poly}(\Delta') + \log^* n) = O(\log^* n)$ time [38, 27, 11], with $\Delta' = O(c^2) = O(1)$.

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 $V^\star \setminus (V_{\mathsf{sp}} \cup V_{\mathsf{bad}} \cup R \cup X)$ that remain uncolored after step 3 of the algorithm described in Figure 2.

4.3. Coloring the remaining vertices. At this point, all uncolored vertices are in $U \cup V_{\mathsf{sp}} \cup V_{\mathsf{bad}}$. We show that $U \cup V_{\mathsf{sp}}$ can be colored efficiently in $O(\log^* \Delta)$ time using Lemma 2.1 and then consider V_{bad}.

Coloring the vertices in 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' is at layer i', and $i > i'$, or if $i = i'$ and $\text{ID}(u) > \text{ID}(u')$. Recall that $N_{\text{out}}(v)$ is 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 4.2 and 4.4. The out-degree of v is therefore at most $\sum_{j=1}^{i} \epsilon_j^5 \Delta = O(\epsilon_i^5 \Delta) = O(\epsilon_{i-1}^{5/2} \Delta)$.

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 2.1 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 2.1. 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 $p_v = \eta \epsilon_{i-1}^2 \Delta \le |\Psi(v)| - \deg(v)$. There is a constant $C > 0$ such that for each $i \in [2, \ell]$ and each layer-i vertex $v \in U$, we have

$$\sum_{u \in N_{\text{out}}(v)} 1/p_u \le \sum_{j=2}^{i} O\left(\frac{\epsilon_{j-1}^{5/2} \Delta}{\epsilon_{j-1}^2 \Delta}\right) = \sum_{j=2}^{i} O(\epsilon_{j-1}^{1/2}) < 1/C.$$

The remaining parameters to Lemma 2.1 are

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

Thus, by Lemma 2.1 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 2.1. We let G'' be any acyclic orientation of the graph induced by V_{sp}, e.g., orienting each edge $\{u, v\}$ towards the vertex v such that $\text{ID}(u) > \text{ID}(v)$. 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 $\sum_{u \in N_{\text{out}}(v)} (1/p_u) \le \text{outdeg}(v)/(\gamma\Delta) \le 1/\gamma$. Thus, we can apply Lemma 2.1 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 $\Delta \gg \operatorname{poly}\log n$,[7] then $V_{\mathsf{bad}} = \emptyset$, w.h.p., in view of the failure probabilities $\exp(-\Omega(\operatorname{poly}(\Delta)))$ specified in the lemmas used in the previous coloring steps. Otherwise, $\Delta = \operatorname{poly}\log n$, and by Lemma 4.1, each connected component of V_{bad} has size at most $\operatorname{poly}(\Delta)\log n = \operatorname{poly}\log n$. In any case, it takes $\mathsf{Det}_d(\operatorname{poly}\log n)$ to color all vertices in V_{bad} deterministically.

In our application of Lemma 4.1, both c and c' are set to be constants. This is possible because the calculation of the probability that a vertex v joins V_{bad} during our algorithm works even if the random bits outside of a constant radius of v are determined adversarially. This is consistent with the fact that whether a vertex v joins V_{bad} depends on vertices outside of its constant radius. This argument is used in many applications of the graph shattering technique [13].

See Figure 2 for a synopsis of every step of the $(\Delta+1)$-list coloring algorithm.

4.4. Time complexity. The time for $\mathsf{OneShotColoring}$ (Figure 2, step 2) is $O(1)$. The time for processing each of V^{S}_{2+}, V^{S}_{1}, V^{M}_{2+}, V^{M}_{1}, V^{L}_{2+}, V^{L}_{1} (steps 3(a)–(f)) is $O(1)$. Observe that each of steps 2 and 3(a)–(f) may put vertices in V_{bad}, that steps 3(a), (c), (e) leave some vertices uncolored, and that step 3(f) also puts vertices in special sets X and R. W.h.p., R induces components with constant degree, which can be colored deterministically in $O(\log^* n)$ time (step 4). The uncolored vertices (U) from steps 3(a), (c), (e) have a large gap between their palette size and degree and can be colored in $O(\log^*\Delta)$ time using the $\mathsf{ColorBidding}$ algorithm (Lemma 2.1) in step 6. The same type of palette size-degree gap exists for V_{sp} as well, so $\mathsf{ColorBidding}$ colors it in $O(\log^*\Delta)$ time; for step 7, we are applying Lemma 2.1 again but with different parameters.

Finally, steps 5 and 8 solve a $(\deg+1)$-list coloring problem on a graph whose components have size $\operatorname{poly}\log n$. Observe that V_{bad} is guaranteed to induce components with size $\operatorname{poly}(\Delta)\log n$, which happens to be $\operatorname{poly}\log n$ since no vertices are added to V_{bad}, w.h.p., if $\Delta \gg \operatorname{poly}\log n$ is sufficiently large. In contrast, in step 5, X can be nonempty even when Δ is large, but it still induces components with size $\operatorname{poly}\log n$.

Since $\log^*\Delta \le \log^* n = O(\mathsf{Det}_d(\operatorname{poly}\log n))$ [38], the bottleneck in the algorithm is solving $(\deg+1)$-list coloring in steps 5 and 8.

THEOREM 4.6. *In the* LOCAL *model, the* $(\Delta+1)$*-list coloring problem can be solved, w.h.p., in* $O(\mathsf{Det}_d(\operatorname{poly}\log n))$ *time, where* $\mathsf{Det}_d(n')$ *is the deterministic complexity of* $(\deg+1)$*-list coloring on* n'*-vertex graphs.*

Next, we argue that if the palettes have $\operatorname{poly}\log n$ extra colors initially, we can list color the graph in $O(\log^*\Delta)$ time.

THEOREM 4.7. *There is a universal constant* $\gamma > 0$ *such that the* $(\Delta+\log^\gamma n)$*-list coloring problem can be solved in the* LOCAL *model, w.h.p., in* $O(\log^*\Delta)$ *time.*

Proof. For all parts of our $(\Delta+1)$-list coloring algorithm, except the first case of Lemma 4.5, the probability that a vertex v joins V_{bad} is $\exp(-\Omega(\operatorname{poly}(\Delta)))$. Let α and c be the constants in Lemma 4.5 and $k_1 = \Theta(c) \ge \alpha c$ be such 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 $\exp(-\Omega(\operatorname{poly}(\Delta))) = 1/\operatorname{poly}(n)$. Note that when $\Delta > \log^{k_1} n$, no vertex is added

[7] Precisely, it means that $\Delta = \Omega(\log^h n)$ for some universal constant $h > 0$.

$(\Delta + 1)$-List Coloring Algorithm

1. Determine the ϵ-almost cliques for $\epsilon \in \{\epsilon_1, \ldots, \epsilon_\ell\}$ (Lemma 3.1).
2. Perform the initial coloring step using algorithm OneShotColoring (Lemma 2.5), and partition the remaining uncolored vertices into $V^\star$ and V_{bad}. Further partition $V^\star$ into a sparse set V_{sp} and a hierarchy $\mathcal{T}$ of small, medium, and large blocks. Partition $V^\star \backslash V_{\mathsf{sp}}$ into six sets: $V_{2+}^{\mathsf{S}}, V_1^{\mathsf{S}}, V_{2+}^{\mathsf{M}}, V_1^{\mathsf{M}}, V_{2+}^{\mathsf{L}}, V_1^{\mathsf{L}}$.
3. Color most of $V_{2+}^{\mathsf{S}}, V_1^{\mathsf{S}}, V_{2+}^{\mathsf{M}}, V_1^{\mathsf{M}}, V_{2+}^{\mathsf{L}}, V_1^{\mathsf{L}}$ in six steps.
   (a) Color a subset of V_{2+}^{S} using algorithm DenseColoringStep (version 1). Any vertices that violate the conclusion of Lemma 4.2 are added to V_{bad}.
   (b) Color V_1^{S} using algorithm DenseColoringStep (version 1). Any remaining uncolored vertices are added to V_{bad} (Lemma 4.3).
   (c) Color a subset of V_{2+}^{M} using algorithm DenseColoringStep (version 1). Any vertices that violate the conclusion of Lemma 4.2 are added to V_{bad}.
   (d) Color V_1^{M} using algorithm DenseColoringStep (version 1). Any remaining uncolored vertices are added to V_{bad} (Lemma 4.3).
   (e) Color a subset of V_{2+}^{L} using algorithm DenseColoringStep (version 2). Any vertices that violate the conclusion of Lemma 4.4 are added to V_{bad}.
   (f) Color V_1^{L} using algorithm DenseColoringStep (version 2). Each remaining uncolored vertex is added to one of X, R, or V_{bad}. (See Lemma 4.5.)
4. W.h.p., R induces a graph with constant maximum degree. Color R in $O(\log^* n)$ time deterministically using a standard algorithm [38, 27, 11].
5. W.h.p., X induces a graph whose components have size $\operatorname{poly}\log n$. Color X in $O(\mathsf{Det}_d(\operatorname{poly}\log n))$ time deterministically; see [48, 13].
6. Color those uncolored vertices U in $\left(V_{2+}^{\mathsf{S}} \cup V_{2+}^{\mathsf{M}} \cup V_{2+}^{\mathsf{L}}\right) \backslash V_{\mathsf{bad}}$ in $O(\log^* \Delta)$ time using algorithm ColorBidding (Lemma 2.1). Any vertices in U that remain uncolored are added to V_{bad}.
7. Color V_{sp} in $O(\log^* \Delta)$ time using algorithm ColorBidding (Lemma 2.1). Any vertices that remain uncolored are added to V_{bad}.
8. W.h.p., V_{bad} induces components of size $\operatorname{poly}\log n$. Color V_{bad} in $O(\mathsf{Det}_d(\operatorname{poly}\log n))$ time deterministically; see [48, 13].

FIG. 2. *Steps* 1, 2, *and* 3(a)–(f) *take constant time. Steps* 4, 6, *and* 7 *take* $O(\log^* n) = O(\mathsf{Det}_d(\operatorname{poly}\log n))$ *time* [38, 42]. *The bottlenecks in the algorithm are steps* 5 *and* 8, *which take* $O(\mathsf{Det}_d(\operatorname{poly}\log n))$ *time. The algorithm succeeds in the prescribed time, so long as the input to steps* 4, 5, *and* 8 *are as they should be, i.e., inducing subgraphs with constant degree, or* $\operatorname{poly}\log n$*-size components, respectively. (These are instances of* $(\deg+1)$*-list coloring.) When* $\Delta \gg \operatorname{poly}\log n$ *is sufficiently large, the set* V_{bad} *is empty, w.h.p., but* X *may be nonempty and induce components with size* $\operatorname{poly}\log n$.

to V_{bad} in Lemma 4.5.

Let $R' = R \cup X$ be the leftover vertices in Lemma 4.5 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 2.2 directly, with $\rho =$

$\frac{\log^\gamma n}{\Delta} - 1 = \Omega(\log n)$. The algorithm takes $O(1+\log^* \Delta - \log^* \rho) = O(1)$ time, and the probability that a vertex v is not colored is $\exp(-\Omega(\sqrt{\rho\Delta})) = \exp(-\Omega(\log^{\gamma/2} n)) \ll 1/\text{poly}(n)$.

If $\Delta > \log^{\gamma-1} n$, then we apply steps 1, 2, 3, 6, and 7 of our $(\Delta+1)$-list coloring algorithm. Due to the lower bound on Δ, we have $V_{\mathsf{bad}} = \emptyset$, w.h.p., which obviates the need to implement step 8.

This algorithm takes $O(\log^* \Delta)$ time and produces an uncolored subgraph $R' = R \cup X$ that has maximum degree $\Delta' \le \log^{k_2} n$. In lieu of steps 4 and 5, we apply the algorithm of Lemma 2.2 to color R' in $O(1 + \log^* \Delta' - \log^* \rho) = O(1)$ time, where $\rho = \frac{\log^\gamma n}{\Delta'} - 1 = \Omega(\log n)$. ◻

If every vertex is ϵ-sparse, with $\epsilon^2\Delta$ sufficiently large, then the algorithm of Lemma 2.5 gives *every* vertex $\Omega(\epsilon^2\Delta)$ excess colors, w.h.p. Combining this observation with Theorem 4.7, 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 4.8. *There is a universal constant $\gamma > 0$ such that the following holds. Suppose G is a graph with maximum degree Δ in which each vertex is ϵ-sparse, where $\epsilon^2\Delta > \log^\gamma n$. A $(\Delta+1)$-list coloring of G can be computed in the* LOCAL *model, w.h.p., in $O(\log^* n)$ time.*

Note that the assumption $\epsilon^2\Delta > \log^\gamma n$ in Theorem 4.8 implies that $\log^* \Delta = \Theta(\log^* n)$.

Remark 3. Theorem 4.8 insists on every vertex being *ϵ-sparse* according to Definition 2.4. It is straightforward to show connections between this definition of sparsity and other standard measures from the literature. For example, such a graph is $(1-\epsilon')$-*locally sparse*, where $\epsilon' = \Omega(\epsilon^2)$, according to Definition 2.3. Similarly, any $(1-\epsilon')$-locally sparse graph is $\Omega(\epsilon')$-sparse. Graphs of *degeneracy* $d \le (1-\epsilon')\Delta$ or *arboricity* $\lambda \le (1/2-\epsilon')\Delta$ are trivially $(1-\Omega(\epsilon'))$-locally sparse.

Remark 4. We have made no effort to minimize the constant γ in Theorems 4.7 and 4.8, and it is impractically large. It would be useful to know whether these theorems remain true when γ is small, say 1; i.e., is $(\Delta + \log n)$-coloring solvable in $O(\log^* \Delta)$ time, w.h.p.?

Remark 5. Our algorithm requires that all vertices know the parameter Δ. It is an open question to achieve the same time complexity for $(\Delta+1)$-coloring without this assumption. The exact knowledge of n is not absolutely necessary. The algorithm works as long as all vertices agree on an estimate $n' = n^{\Omega(1)}$.

5. Fast coloring using excess colors. In this section, we prove Lemma 2.1. 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 $p_v \le |\Psi(v)| - \deg(v)$; i.e., p_v is a lower bound on the number of excess colors at v. All vertices agree on values $p^\star \le \min_{v\in V} p_v$, $d^\star \ge \max_{v \in V} \text{outdeg}(v)$, and $C = \Omega(1)$, such that the following is satisfied for all v:

$$\sum_{u \in N_{\text{out}}(v)} 1/p_u \le 1/C. \tag{5.1}$$

Intuitively, the sum $\sum_{u\in N_{\text{out}}(v)} 1/p_u$ measures the amount of "contention" at a vertex v. In the ColorBidding algorithm, each vertex v selects each color $c \in \Psi(v)$ with probability $\frac{C}{2|\Psi(v)|} < \frac{C}{2p_v}$ and permanently colors itself if it selects a color not selected by any out-neighbor.

Procedure ColorBidding.

1. Each color $c \in \Psi(v)$ is added to S_v independently with probability $\frac{C}{2|\Psi(v)|}$.
2. If there exists a color $c^\star \in S_v \setminus \left(\cup_{u \in N_{\text{out}}(v)} S_u\right)$, v permanently colors itself $c^\star$.

In Lemma 5.1, 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 large probability.

LEMMA 5.1. *Consider an execution of* ColorBidding*. Let v be any vertex. Let D be the summation of $1/p_u$ over all vertices u in $N_{\text{out}}(v)$ that remain uncolored after* ColorBidding*. Then the following holds:*

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

Proof. For each vertex v, we define the following two events:
E_v^{good} : v selects a color that is not selected by any vertex in $N_{\text{out}}(v)$.
E_v^{bad} : the number of colors in $\Psi(v)$ that are selected by some vertices in $N_{\text{out}}(v)$ is at least $\frac{2}{3} \cdot |\Psi(v)|$.
Notice that E_v^{good} is the event where v successfully colors itself. We first show that $\Pr[E_v^{\text{bad}}] = \exp(-\Omega(p^\star))$. Fix any color $c \in \Psi(v)$. The probability that c is selected by some vertex in $N_{\text{out}}(v)$ is

$$1 - \prod_{u \in N_{\text{out}}(v)} \left(1 - \tfrac{C}{2|\Psi(u)|}\right) \le 1 - \prod_{u \in N_{\text{out}}(v)} \left(1 - \tfrac{C}{2p_u}\right) \le \sum_{u \in N_{\text{out}}(v)} \tfrac{C}{2p_u} \le \tfrac{1}{2},$$

where the last inequality follows from (5.1). Since these events are independent for different colors, $\Pr[E_v^{\text{bad}}] \le \Pr[\text{Binomial}(n', p') \ge \frac{2n'}{3}]$, with $n' = |\Psi(v)| \ge p_v$ and $p' = \frac{1}{2}$. By a Chernoff bound, we have

$$\Pr\left[E_u^{\text{bad}}\right] \le \exp(-\Omega(n'p')) = \exp(-\Omega(p^\star)).$$

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

$$\Pr\left[\overline{E_v^{\text{good}}} \;\middle|\; \overline{E_v^{\text{bad}}}\right] \le \left(1 - \tfrac{C}{2|\Psi(v)|}\right)^{|\Psi(v)|/3} \le \exp\left(\tfrac{-C}{6}\right). \tag{5.2}$$

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

Next, we prove the second inequality on the upper tail of the random variable D. Let $N_{\text{out}}(v) = (u_1, \ldots, u_k)$. Let E_i^{bad} and E_i^{good} be short for $E_{u_i}^{\text{bad}}$ and $E_{u_i}^{\text{good}}$, and let $\mathcal{E}$ be the event $\bigcup_i E_i^{\text{bad}}$. By a union bound,

$$\begin{aligned}\Pr[\mathcal{E}] &\le \text{outdeg}(v) \cdot \exp(-\Omega(p^\star)) \\ &\le d^\star \cdot \exp(-\Omega(p^\star)).\end{aligned}$$

Let $X = \sum_{i=1}^{k} X_i$, where $X_i = 1/p_{u_i}$ if *either* $\overline{E_i^{\text{good}}}$ or E_i^{bad} occurs, and $X_i = 0$ otherwise. Observe that if we condition on $\overline{\mathcal{E}}$, then X is exactly D, the random variable we want to bound.

By linearity of expectation,

$$\begin{aligned}\mu = \mathrm{E}[X \mid \overline{\mathcal{E}}] = \sum_i \mathrm{E}[X_i \mid \overline{\mathcal{E}}] \\ &\leq \sum_i \frac{1}{p_{u_i}} \cdot \Pr\left[\overline{E_i^{\text{good}}} \;\middle|\; \overline{E_i^{\text{bad}}}\right] \\ &\leq \sum_i \frac{1}{p_{u_i}} \cdot \exp(-C/6) && \text{(see (5.2))} \\ &\leq \frac{\exp(-C/6)}{C} && \text{(see (5.1)).}\end{aligned}$$

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 $\sum_{i=1}^{k}(b_i - a_i)^2 \leq \sum_{u \in N_{\text{out}}(v)} 1/(p_u \cdot p^\star) \leq 1/(Cp^\star)$. By Hoeffding's inequality, we have

$$\begin{aligned}\Pr\left[X \geq (1+\lambda)\cdot\frac{\exp(-C/6)}{C} \;\middle|\; \overline{\mathcal{E}}\right] &\leq \Pr[X \geq (1+\lambda)\mu \mid \overline{\mathcal{E}}] \\ &\leq \exp\left(\frac{-2(\lambda\mu)^2}{\sum_{i=1}^{k}(b_i - a_i)^2}\right) \\ &\leq \exp\left(-2\left(\frac{\lambda\exp(-C/6)}{C}\right)^2 Cp^\star\right) \\ &= \exp\left(-\frac{2\lambda^2 p^\star \exp(-C/3)}{C}\right).\end{aligned}$$

Thus,

$$\begin{aligned}&\Pr[D \geq (1+\lambda)\exp(-C/6)/C] \\ &\leq \Pr[X \geq (1+\lambda)\exp(-C/6)/C \mid \overline{\mathcal{E}}] + \Pr[\mathcal{E}] \\ &\leq \exp\left(-2\lambda^2 p^\star \exp(-C/3)/C\right) + d^\star \exp(-\Omega(p^\star)).\end{aligned}$$

Note that 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 $N_{\text{out}}(v) = (u_1, \ldots, u_k)$ is sorted in reverse topological order, and so for each $1 \leq j \leq k$, we have $N_{\text{out}}(u_j) \cap \{u_j, \ldots, u_k\} = \emptyset$. Thus, conditioning on (i) $\overline{E_i^{\text{bad}}}$ and (ii) *any* colors selected by vertices in $\bigcup_{1 \leq j < i} N_{\text{out}}(u_j) \cup \{u_j\}$, the probability that $\overline{E_i^{\text{good}}}$ occurs is still at most $\exp(\frac{-C}{6})$. ∎

Restatement of Lemma 2.1. *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 $\sum_{u \in N_{\text{out}}(v)} 1/p_u \leq 1/C$. Let $d^\star$ be the maximum out-degree of the graph. There is an algorithm that takes $O(1 + \log^* p^\star - \log^* C)$ 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.*

Proof. In what follows, we show how Lemma 5.1 can be used to derive Lemma 2.1. Our plan is to apply ColorBidding for $k^\star = \log^* p^\star - \log^* C + O(1)$ iterations. For the kth iteration, we use the parameter C_k, which is defined as follows:

$$
\begin{aligned}
C_1 &= \min\{\sqrt{p^\star},\ C\},\\
C_k &= \min\left\{\sqrt{p^\star},\ \frac{C_{k-1}}{(1+\lambda)\exp(-C_{k-1}/6)}\right\},\\
k^\star &= \min\left\{k \mid C_k = \sqrt{p^\star}\right\} \qquad \text{(the last iteration).}
\end{aligned}
$$

Here $\lambda > 0$ must be selected to be sufficiently small so that

$$(1+\lambda)\exp(-C_{k-1}/6) < 1.$$

This guarantees that the sequence (C_k) is strictly increasing. For example, if $C \geq 6$ initially, we can fix $\lambda = 1$ throughout.

We analyze each iteration of ColorBidding using the same (initial) vector of (p_v) values; i.e., we do not count on the number of excess colors at any vertex increasing over time.

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

- If $k \in [1, k^\star)$, $\mathcal{H}_k$ stipulates that for each uncolored vertex v after the kth iteration, the summation of $1/p_u$ over all uncolored $u \in N_{\text{out}}(v)$ is less than $1/C_{k+1}$.
- $\mathcal{H}_{k^\star}$ stipulates that all vertices still participating are colored at the end of the $k^\star$th iteration.

The purpose of $\mathcal{H}_k$, $k \in [1, k^\star)$, is to guarantee that C_{k+1} is a valid parameter for the $(k+1)$th iteration of ColorBidding. For each $k \in [1, k^\star]$, at the end of the kth iteration we remove all vertices violating $\mathcal{H}_k$ from further participation in the procedure and add them to the set V_{bad}. Thus, by the 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 the probability of v joining V_{bad} is at most $\exp(-\Omega(\sqrt{p^\star})) + d^\star \exp(-\Omega(p^\star))$, and this is true even if the randomness outside a constant radius around v is determined adversarially. By Lemma 5.1, the probability that a vertex is removed at the end of the kth iteration, where $k \in [1, k^\star)$, is at most

$$
\begin{aligned}
&\exp(\Omega(p^\star/C_{k+1})) + d^\star \exp(-\Omega(p^\star))\\
&\leq \exp(-\Omega(\sqrt{p^\star})) + d^\star \exp(-\Omega(p^\star)).
\end{aligned}
$$

The probability that a vertex 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 $k^\star = \log^* p^\star - \log^* C + O(1)$ iterations, the probability that a vertex joins V_{bad} is $\exp(-\Omega(\sqrt{p^\star})) + d^\star \exp(-\Omega(p^\star))$. ∎

6. Coloring locally dense vertices. Throughout this section, we consider the following setting. We are given a graph $G = (V, E)$, where some vertices are already colored. We are also given a *subset* S of the uncolored vertices, which is partitioned into g disjoint *clusters* $S = S_1 \cup S_2 \cup \cdots \cup S_g$, each with weak diameter 2. (In particular, this implies that otherwise sequential algorithms can be executed on each cluster in $O(1)$ rounds in the LOCAL model.) 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' is at layer i', and $i > i'$, or if $i = i'$ and $\mathrm{ID}(u) > \mathrm{ID}(u')$. Notice that this orientation is *acyclic*. In this section, $N_{\text{out}}(v) \subseteq S$ denotes the set of out-neighbors of v in S, as we only focus on the vertices in S.

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 6.1, which is then used to prove Lemmas 4.2 and 4.3. In section 6.2, we describe a procedure DenseColoringStep (version 2), which is a generalization of Harris, Schneider, and Su's procedure [32]. It is analyzed in Lemma 6.2, which is then used to prove Lemmas 4.4 and 4.5.

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 $|N(v) \cap S|$ 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 $|N_{\text{out}}(v) \cap (S \setminus S_j)| \leq D_v$.

The ratio of these two quantities plays an important role in the analysis. Define δ_v as

$$\delta_v = \frac{D_v}{Z_v}.$$

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 settings of Lemmas 4.2 and 4.3. For Lemma 4.3, S is either V_1^{S} or V_1^{M}, and each cluster of S is the intersection of S and an ϵ_1-almost clique (a layer-1 block). For Lemma 4.2, S is either V_{2+}^{S} or V_{2+}^{M}, and each cluster of S is the intersection of S and an ϵ_ℓ-almost clique. In all cases, clusters have weak diameter 2. All vertices in the same layer adopt the same D- and Z-values. A layer-i vertex v takes

$$\begin{aligned} Z_v &= \frac{\Delta}{2\log(1/\epsilon_i)}, \\ D_v &= \epsilon_i \Delta. \end{aligned}$$

The choices of these parameters are valid in view of the excess colors implied by Lemma 3.3 and the external degree upper bound of Lemma 3.1.

Procedure DenseColoringStep (version 1).

1. Let $\pi : \{1, \ldots, |S_j|\} \to S_j$ be the *unique* permutation that lists the vertices of S_j in increasing order by layer number, breaking ties (within the same layer) 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 $N_{\text{out}}(v)$.

Notice that π is a reverse topological ordering of S_j, i.e., if $\pi(q')$ precedes $\pi(q)$, then $\pi(q) \notin N_{\text{out}}(\pi(q'))$. Because each S_j has weak diameter 2, we can simulate step 1 of DenseColoringStep in just $O(1)$ rounds of communication. Intuitively, the probability that a vertex $v \in S$ remains uncolored after DenseColoringStep (version 1) is at most δ_v since it is *guaranteed* not to have any conflicts with neighbors in the same cluster. The following lemma gives us the probabilistic guarantee of the DenseColoringStep (version 1).

LEMMA 6.1. *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 at this moment, 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 $\text{Binomial}(|T|, \delta)$. ∎

RESTATEMENT OF LEMMA 4.2. *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 $1 - \exp(-\Omega(\text{poly}(\Delta)))$, 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}.*

Proof. We execute DenseColoringStep (version 1) for six iterations, where each participating vertex $x \in S$ uses the same (initial) values of Z_x and D_x, namely $Z_x = \frac{\Delta}{2\log(1/\epsilon_i)}$ and $D_x = \epsilon_i\Delta$ if x is at layer i.

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 4.2, it suffices to show that after six 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:

$$\begin{aligned}
\delta &= \max_{u\in T}\{\delta_u\} \ = \ 2\epsilon_i \log(1/\epsilon_i), \\
t_1 &= |T|, \\
t_k &= \max\left\{(2\delta)t_{k-1}, \epsilon_i^5\Delta\right\}.
\end{aligned}$$

Since $(2\delta)^6|T| \leq \epsilon_i^5\Delta$, we have $t_7 = \epsilon_i^5\Delta$. (Remember that T is the set of layer-i neighbors of v in S, and so $|T| \leq \Delta$.)

Assume that at the beginning of the kth iteration, the number of uncolored vertices in T is at most t_k. Indeed, for $k = 1$, we initially have $t_1 = |T|$. By Lemma 6.1, after the kth iteration, the expected number of uncolored vertices in T is at most $\delta t_k \leq t_{k+1}/2$. By a Chernoff bound, with probability at most $\exp(-\Omega(t_{k+1})) \leq \exp(-\Omega(\epsilon_i^5\Delta)) = \exp(-\Omega(\text{poly}(\Delta)))$, the number of uncolored vertices in T is more than t_{k+1}.

Therefore, after six iterations of DenseColoringStep (version 1), with probability $1-\exp(-\Omega(\text{poly}(\Delta)))$, the number of uncolored vertices in T is at most $t_7 = \epsilon_i^5 \Delta$, as required. ∎

Restatement of Lemma 4.3. *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 $1-\exp(-\Omega(\text{poly}(\Delta)))$; all uncolored vertices in S join V_{bad}.*

Proof. In the setting of Lemma 4.3, we only consider layer-1 vertices but have the higher burden of coloring *each* vertex with high enough probability. Since $\epsilon_1 = \Delta^{-1/10}$, we have $Z_v = \frac{\Delta}{2\log(1/\epsilon_1)}$, $D_v = \epsilon_1\Delta$, and $\delta_v = D_v/Z_v = 2\epsilon_1 \log(1/\epsilon_1)$ for all vertices $v \in S$.

We begin with one iteration of DenseColoringStep (version 1). By Lemma 6.1 and a Chernoff bound, for each $v \in S$, the number of uncolored vertices of $N(v) \cap S$ is at most $2\delta_v\Delta = \Delta' < O(\Delta^{9/10}\log\Delta)$ with probability $1-\exp(-\Omega(\text{poly}(\Delta)))$. Any uncolored vertex $v \in S$ that violates this property, i.e., for which $|N(v)\cap S| > \Delta'$, is added to V_{bad} and removed from further consideration.

Consider the graph G' induced by the remaining uncolored vertices in S. The maximum degree of G' is at most Δ'. Each vertex v in G' satisfies $|\Psi(v)| \geq Z_v = \frac{\Delta}{2\log(1/\epsilon_1)} = (1+\rho)\Delta'$, where ρ is $\Delta^{\Omega(1)}$. We run the algorithm of Lemma 2.2 on G' and then put all vertices that still remain uncolored in the set V_{bad}. By Lemma 2.2, the time for this procedure is $O(\log^*\Delta - \log^*\rho) = O(1)$, and the probability that a vertex v remains uncolored and is added to V_{bad} is at most $\exp(-\Omega(\sqrt{\rho\Delta})) = \exp(-\Omega(\text{poly}(\Delta)))$. ∎

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 $S = V_{2+}^{\mathsf{L}}$ (Lemma 4.4), each cluster S_j is a large block in some layer $i \in [2,\ell]$. For the case $S = V_1^{\mathsf{L}}$ (Lemma 4.5), each S_j is a layer-1 large block. For each $v \in S$, we define $N^\star(v)$ to be 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. Observe that $N_{\text{out}}(v) \subseteq N^\star(v)$ since $N_{\text{out}}(v)$ excludes some vertices at v's layer, depending on the ordering of IDs. For the case of $S = V_1^{\mathsf{L}}$, all clusters $S_1,\ldots,S_g$ are layer-1 blocks, and so $N^\star(v) = N(v)\cap S$. We make the following assumptions.

Identifiers. List the clusters $S_1,\ldots,S_g$ in nondecreasing order by layer number. We assume each cluster and each vertex within a cluster have an ID that is consistent with this order, in particular

$$\begin{aligned}&\text{ID}(S_1) < \cdots < \text{ID}(S_g),\\ &\max_{v\in S_j} \text{ID}(v) < \min_{u\in S_{j+1}} \text{ID}(u) \ \text{ for all } j \in [1,g).\end{aligned}$$

Given arbitrary IDs, it is straightforward to compute new IDs satisfying these properties in $O(1)$ time. (It is not required that each cluster S_j learns 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 following two conditions:

(i) $|S_j \setminus (N(v)\cup\{v\})| = |S_j \setminus (N^\star(v)\cup\{v\})| \leq D_j$ (antidegree upper bound).

(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

$$1/K \geq \delta_j \geq \frac{D_j \log(|S_j|/D_j)}{|S_j|}$$

for some sufficiently large constant K.

The procedure DenseColoringStep (version 2) aims to successfully color a large fraction of the vertices in each cluster S_j. In step 1, each cluster selects a $(1-\delta_j)$-fraction of its vertices uniformly at random, permutes them randomly, and marches through this permutation one vertex at a time. As in DenseColoringStep (version 1), when a vertex v is processed it picks a random color $c(v)$ from its available palette that was not selected by previously processed vertices in S_j. Step 2 is the same: if $c(v)$ has not been selected by any vertices of $N_{\text{out}}(v)$, it permanently commits to $c(v)$. There are only two reasons a vertex in S_j may be left uncolored by DenseColoringStep (version 2): it is not among the $(1-\delta_j)$-fraction of vertices participating in step 1, or it has a color conflict with an external neighbor in step 2. The first cause occurs with probability δ_j and, intuitively, the second cause occurs with probability about δ_j because vertices typically have *many* options for colors when they are processed but *few* external neighbors that can generate conflicts. Lemma 6.2 captures this formally; it is the culmination and corollary of Lemmas 6.3–6.5, which are proved later in this section. Lemma 6.2 is used to prove Lemmas 4.4 and 4.5.

Procedure DenseColoringStep (version 2).

1. Each cluster S_j selects $(1-\delta_j)|S_j|$ vertices uniformly at random and generates a permutation π of those vertices uniformly at random. The vertex $\pi(q)$ selects a color $c(\pi(q))$ uniformly at random 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 $u \in N_{\text{out}}(v)$.

LEMMA 6.2. *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 $N_{\text{out}}(v) \subseteq \bigcup_{i=1}^{j} S_i$. Therefore, in the proof of Lemma 6.2, we expose the random bits of the clusters in the order $(S_1, \dots, S_g)$. Once the random bits of $S_1, \dots, S_j$ are revealed, we can determine whether any particular $v \in S_j$ successfully colors itself.

Our proofs of Lemmas 4.4 and 4.5 are based on a constant number of iterations of DenseColoringStep (version 2). In each iteration, the parameters D_j and δ_j might be different. In subsequent discussion, the term *antidegree* 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 large 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 3.1, $D_j^{(1)} = 3\epsilon_i\Delta$ upper bounds the initial antidegree and external degree. 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 on the antidegree and external degree of all uncolored vertices in $S_j \setminus V_{\mathsf{bad}}$.

Cluster size upper bounds. By Lemma 3.1, $U_j^{(1)} = (1+3\epsilon_i)\Delta$ is an upper bound on the initial cluster size. 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 $S_j \setminus V_{\mathsf{bad}}$ 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 $S_j \setminus V_{\mathsf{bad}}$ is at least $L_j^{(k)}$. By the definition of *large* blocks, $\mathcal{L}_j^{(1)}$ holds initially.

Shrinking rates. For each k, the shrinking rate $\delta_j^{(k)}$ of cluster S_j for the kth iteration is chosen such that

$$1/K \geq \delta_j^{(k)} \geq \frac{D_j^{(k)} \log(L_j^{(k)}/D_j^{(k)})}{L_j^{(k)}}.$$

Additionally, we require that $\delta_1^{(k)} \leq \cdots \leq \delta_g^{(k)}$, with $\delta_j^{(k)} = \delta_{j+1}^{(k)}$ if S_j and S_{j+1} are in the same layer.

Although the initial values of $D_j^{(1)}, U_j^{(1)}, L_j^{(1)}$ are determined, there is considerable freedom in choosing the remaining values to satisfy the four rules above. We refer to the following equations involving $D_j^{(k)}, U_j^{(k)}, L_j^{(k)}$, and $\delta_j^{(k)}$ as the *default settings* of these parameters. Unless stated otherwise, the proofs of Lemmas 4.4 and 4.5 use the default settings:

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

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

Validity of parameters. Before the first iteration, the invariants $\mathcal{D}_j^{(1)}$, $\mathcal{U}_j^{(1)}$, and $\mathcal{L}_j^{(1)}$ are met initially for each cluster S_j. Suppose S_j is a layer-i large block. Lemma 3.1 shows that the initial value of $D_j^{(1)}$ is a valid upper bound on the external degree (at most $\epsilon_i\Delta$) and antidegree (at most $3\epsilon_i\Delta$). We also have

$$U_j^{(1)} = (1+3\epsilon_i)\Delta \geq |S_j| \geq \frac{\Delta}{\log(1/\epsilon_i)} = L_j^{(1)},$$

where the lower bound is from the definition of *large* and the upper bound is from Lemma 3.1.

For $k > 1$, the invariants $\mathcal{D}_j^{(k)}$ and $\mathcal{U}_j^{(k)}$ might not hold naturally. Before the kth iteration begins, we *forcibly* restore them by removing from consideration all vertices in the clusters that violate either invariant, putting these vertices in V_{bad}. Notice that $\mathsf{DenseColoringStep}$ (version 2) *always* satisfies invariant $\mathcal{L}_j^{(k)}$.

Maintenance of invariants. We calculate the probability for the invariants $\mathcal{D}_j^{(k+1)}$ and $\mathcal{U}_j^{(k+1)}$ to naturally hold at a cluster S_j. In what follows, we analyze the kth iteration of the algorithm and assume that $\mathcal{D}_j^{(k)}$ and $\mathcal{U}_j^{(k)}$ hold 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 6.2, after the kth iteration, the probability that the number of uncolored vertices in T is at least t is at most $\binom{|T|}{t} \cdot \left(O(\delta_j^{(k)})\right)^t$. Using this result, we derive the following bounds:

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

We first consider $\Pr\left[\mathcal{U}_j^{(k+1)}\right]$. Let T be the set of uncolored vertices in $S_j \setminus V_{\mathsf{bad}}$ at the beginning of the kth iteration, and let t be

$$t = U_j^{(k+1)} = \beta\delta_j^{(k)} \cdot U_j^{(k)} \geq \beta\delta_j^{(k)}|T|.$$

This implies that $\delta_j^{(k)}|T|/t \leq 1/\beta$. If we select β to be a large enough constant, then

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

Next, consider $\Pr\left[\mathcal{D}_j^{(k+1)}\right]$. For each vertex $v \in S_j \setminus V_{\mathsf{bad}}$ 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 antidegree (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 $\exp\left(-\Omega\left(D_j^{(k+1)}\right)\right)$, then we conclude that $\Pr[\mathcal{D}_j^{(k+1)}] \geq 1 - O\left(U_j^{(k)}\right)\exp\left(-\Omega(D_j^{(k+1)})\right)$ by a union bound over at most $U_j^{(k)}$ vertices $v \in S_j \setminus V_{\mathsf{bad}}$ that are uncolored at the beginning of the kth iteration.

We show that $\Pr[\mathcal{E}_v^e] \leq \exp\left(-\Omega\left(D_j^{(k+1)}\right)\right)$. We choose T as the set of uncolored vertices in $N^\star(v)\setminus(S_j \cup V_{\mathsf{bad}})$ at the beginning of the kth iteration and set $t = D_j^{(k+1)}$. Since the layer number of each vertex in $N^\star(v) \setminus (S_j \cup V_{\mathsf{bad}})$ 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 $t = D_j^{(k+1)} = \beta\delta_j^{(k)} \cdot D_j^{(k)} \geq \beta\delta_j^{(k)}|T|$, and this implies $\delta_j^{(k)}|T|/t \leq 1/\beta$. If we select β to be a large enough constant, then

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

The bound $\Pr[\mathcal{E}_v^a] \leq \exp\left(-\Omega\left(D_j^{(k+1)}\right)\right)$ is proved in the same way. Based on the probability calculations above, we are now prepared to prove Lemmas 4.4 and 4.5.

RESTATEMENT OF LEMMA 4.4. *There is an $O(1)$-time algorithm that colors a subset of V^{L}_{2+} meeting the following condition. For each $v \in V^\star$ and each layer number $i \in [2, \ell]$, with probability at least $1 - \exp(-\Omega(\text{poly}(\Delta)))$, the number of uncolored layer-i neighbors of v in V^{L}_{2+} is at most $\epsilon_i^5 \Delta$. Vertices that violate this property join the set V_{bad}.*

Proof. We perform six iterations of DenseColoringStep (version 2) using the default settings of all parameters. Recall that the shrinking rate for the kth iteration is $\delta_j^{(k)} = \frac{D_j^{(k)} \log\left(L_j^{(k)}/D_j^{(k)}\right)}{L_j^{(k)}}$ for each cluster S_j. If S_j is a layer-i block, we have $\delta_j^{(k)} = O\big(\epsilon_i \log^2(1/\epsilon_i)\big)$ for each $k \in [1, 6]$ since $D_j^{(\cdot)}$ and $L_j^{(\cdot)}$ decay at the same rate, asymptotically.

Consider any vertex $v \in V^\star$ and a layer number $i \in [2, \ell]$. Let T be the set of layer-i neighbors of v in S. To prove Lemma 4.4, it suffices to show that after six 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_k) as in the proof of Lemma 4.2:

$$t_1 = |T|,$$
$$t_k = \max\left\{\beta \delta_j^{(k-1)} t_{k-1},\ \epsilon_i^5 \Delta\right\}.$$

Here $\delta_j^{(k)}$ is the common shrinking rate of any layer-i cluster S_j. We have $t_7 = \epsilon_i^5 \Delta$ since $\epsilon_i \le \epsilon_\ell$ is sufficiently small.

Assume that at the beginning of the kth iteration, the number of uncolored vertices in $T \setminus V_{\mathsf{bad}}$ is at most t_k, 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 6.2, after the kth iteration, the probability that the number of uncolored vertices in $T \setminus V_{\mathsf{bad}}$ is higher than t_{k+1} is

$$\begin{aligned}\binom{t_k}{t_{k+1}} \cdot \left(O\left(\delta_j^{(k)}\right)\right)^{t_{k+1}} &\le \left(O\left(\delta_j^{(k)}\right) \cdot e t_k / t_{k+1}\right)^{t_{k+1}} \\ &\le (O(1/\beta))^{t_{k+1}} = \exp(-\Omega(t_{k+1})).\end{aligned}$$

Notice that $\exp(-\Omega(t_{k+1})) \le \exp(-\Omega(\epsilon_i^5 \Delta)) = \exp(-\Omega(\text{poly}(\Delta)))$. 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 $S \setminus V_{\mathsf{bad}}$ is at most $t_7 = \epsilon_i^5 \Delta$. ∎

RESTATEMENT OF LEMMA 4.5. *Let c be any sufficiently large constant. Then there is a constant time (independent of c) algorithm that colors a subset of V^{L}_1 while satisfying one of the following cases:*

- *The uncolored vertices of V^{L}_1 are partitioned among R or V_{bad}. The graph induced by R has degree $O(c^2)$; each vertex joins V_{bad} with probability $\Delta^{-\Omega(c)}$.*
- *If $\Delta > \log^{\alpha c} n$, where $\alpha > 0$ is some universal constant, then the uncolored vertices of V^{L}_1 are partitioned among R and X, where the graph induced by R has degree $O(c^2)$ and the components induced by X have size $\log^{O(c)} n$, w.h.p.*

Proof. In the setting of Lemma 4.5, 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 large enough constant related to the failure probability specified in the statement of Lemma 4.5.

The low degree case. The following algorithm and analysis apply to all values of Δ. The conclusion is that we can color most of V_1^{L} such that the probability that any vertex joins V_{bad} is $\Delta^{-\Omega(c)}$ and all remaining uncolored vertices (i.e., R) induce a graph with maximum degree $O(c^2)$. Since the guarantee on V_{bad} is that it induces components with size $\mathrm{poly}(\Delta)\log n$, this analysis is only appropriate when Δ is, itself, $\mathrm{poly}\log n$. We deal with larger Δ in the *high degree case* and prove that the uncolored vertices can be partitioned into R and X with the same guarantee on R, and the stronger guarantee that X induces $\mathrm{poly}\log n$-size components, regardless of Δ.

Phase 1. The first phase consists of nine iterations of DenseColoringStep (version 2), using the default settings of all parameters. Due to the fact that $\epsilon_1 = \Delta^{-1/10}$, we have $\delta^{(k)} = O(\Delta^{-1/10}\log^2\Delta)$ for each $k \in [1, 9]$. Therefore, at the end of the ninth iteration, we have the parameters

$$\begin{aligned} D^{(10)} &= \Theta(\log^{18}\Delta), \\ L^{(10)} &= \Theta(\Delta^{1/10}\log^{17}\Delta), \\ U^{(10)} &= \Theta(\Delta^{1/10}\log^{18}\Delta). \end{aligned}$$

In view of the previous calculations, the probability that all invariants hold for a specific cluster S_j and all $k \in [1, 10]$ is at least $1 - \exp(-\Omega(\log^{18}\Delta))$. If a cluster S_j does not satisfy an invariant for some k, then *all* vertices in S_j halt and join V_{bad}. They do not participate in the kth iteration or subsequent steps.

Phase 2. For the 10th iteration, we switch to a *non*default shrinking rate

$$\delta^{(10)} = \Delta^{-1/20}.$$

However, we still define

$$\begin{aligned} U^{(11)} &= \beta\delta^{(10)} \cdot U^{(10)} = \Theta(\Delta^{1/20}\log^{18}\Delta), \\ L^{(11)} &= \delta^{(10)} \cdot L^{(10)} = \Theta(\Delta^{1/20}\log^{17}\Delta) \end{aligned}$$

according to their default setting. Since $\beta\delta^{(10)} \cdot D^{(10)} = o(1)$, we should not adopt the default definition of $D^{(11)}$. Instead, we fix it to be the sufficiently large constant c:

$$D^{(11)} = c.$$

Using the previous probability calculations, for each cluster S_j the invariant $\mathcal{U}^{(11)}$ holds with probability at least $1 - \exp(-\Omega(\Delta^{1/20}\mathrm{poly}\log\Delta))$, and the invariant $\mathcal{L}^{(11)}$ holds with certainty. 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 halt and join V_{bad}.

Phase 3. For the 11th iteration, we use the default shrinking rate

$$\delta^{(11)} = \frac{D^{(11)}\log(L^{(11)}/D^{(11)})}{L^{(11)}} = \Theta\left(\frac{1}{\Delta^{1/20}\log^{16}\Delta}\right).$$

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 $N_{\text{out}}(v) \setminus S_j$. 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 $O(1)$ additional rounds we color all vertices in S_j but c^2 of them since at most c^2 have potential conflicts outside of S_j. At this point, the remaining uncolored vertices R induce a subgraph 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 $i = 10$.

	$D^{(i)}$	$L^{(i)}$	$U^{(i)}$	$\delta^{(i)}$
$i \in [9]$	$\Theta\left(\Delta^{\frac{10-i}{10}} \log^{2i-2} \Delta\right)$	$\Theta\left(\Delta^{\frac{11-i}{10}} \log^{2i-3} \Delta\right)$	$\Theta\left(\Delta^{\frac{11-i}{10}} \log^{2i-2} \Delta\right)$	$\Theta\left(\Delta^{-\frac{1}{10}} \log^{2} \Delta\right)$
$i = 10$	$\Theta(\log^{18} \Delta)$	$\Theta\left(\Delta^{\frac{1}{10}} \log^{17} \Delta\right)$	$\Theta\left(\Delta^{\frac{1}{10}} \log^{18} \Delta\right)$	$\Delta^{-\frac{1}{20}}$
$i = 11$	c	$\Theta\left(\Delta^{\frac{1}{20}} \log^{17} \Delta\right)$	$\Theta\left(\Delta^{\frac{1}{20}} \log^{18} \Delta\right)$	$\Theta\left(\Delta^{-\frac{1}{20}} \log^{-16} \Delta\right)$

Analysis of Phase 2. Recall that $\delta^{(10)} = \Delta^{-1/20}$ and $D^{(10)} = \Theta(\log^{18} \Delta)$. By Lemma 6.2, the probability that the external degree or antidegree of $v \in S_j$ is at most c is

$$1 - \binom{D^{(10)}}{c} \left(O\left(\delta^{(10)}\right)\right)^{c} \geq 1 - \binom{O\left(\log^{18} \Delta\right)}{c} \left(O\left(\Delta^{-1/20}\right)\right)^{c} \geq 1 - \Delta^{-\Omega(c)}.$$

By a union bound over at most $U^{(10)} = \Theta(\Delta^{1/10} \log^{18} \Delta)$ 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 antidegree 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 $N_{\text{out}}(v) \setminus (S_j \cup V_{\mathsf{bad}})$. 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. As a consequence of Lemma 6.2, we argue that the probability that $\mathcal{E}_v$ occurs for all $v \in Y$ is at most

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

The reason is as follows. Pick some $v \in Y$. If $\mathcal{E}_v$ occurs, then there must exist a neighbor $v' \in N_{\text{out}}(v) \setminus (S_j \cup V_{\mathsf{bad}})$ that is uncolored. The number of uncolored vertices in $N_{\text{out}}(v) \setminus (S_j \cup V_{\mathsf{bad}})$ 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 $v' \in N_{\text{out}}(v) \setminus (S_j \cup V_{\mathsf{bad}})$ 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 $|T| \geq c^2(1 + 1/D^{(11)})$. The probability that all vertices in T are uncolored is $\left(O(\delta^{(11)})\right)^{c^2\left(1+1/D^{(11)}\right)}$ by Lemma 6.2. By a union bound over at most $\left(D^{(11)}\right)^{c^2}$ choices of T, we obtain the desired probabilistic bound.

Recall that the cluster size upper and lower bounds at the beginning of the 11th iteration are

$$U^{(11)} = \Theta(\Delta^{1/20} \log^{18} \Delta) = L^{(11)} \cdot \Theta(\log \Delta),$$
$$L^{(11)} = \Theta(\Delta^{1/20} \log^{17} \Delta).$$

By a union bound over at most $\left(U^{(11)}\right)^{c^2}$ choices of a size-c^2 subset of S_j, the probability f that there exist c^2 vertices $v \in S_j$ such that $\mathcal{E}_v$ occurs is

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

Recall that $D^{(11)} = c$ is sufficiently large. We have

$$\left(U^{(11)}\right)^{c^2} = \left(O\left(L^{(11)} \log \Delta\right)\right)^{c^2}, \tag{6.1}$$

$$\left(D^{(11)}\right)^{c^2} = O(1), \tag{6.2}$$

$$\left(O\left(\delta^{(11)}\right)\right)^{c^2\left(1+1/D^{(11)}\right)} = \left(O\left(\frac{\log(L^{(11)})}{L^{(11)}}\right)\right)^{c^2+c}, \tag{6.3}$$

where $L^{(11)} = \Theta(\Delta^{1/20} \log^8 \Delta)$. Taking the product of (6.1), (6.2), and (6.3), we have

$$f = O(\log \Delta)^{O(c^2)} \cdot O\left(\Delta^{-1/20}\right)^c = \Delta^{-\Omega(c)},$$

as required.

Remark 6. The analysis of Phase 2 would proceed in the same way if we had chosen $\delta^{(10)}$ according to its default setting of $\Theta(\Delta^{-1/10} \log^2 \Delta)$. We choose a larger value of $\delta^{(10)}$ in order to keep $L^{(11)}$ artificially large ($\Delta^{\Omega(1)}$) and thereby allow Phase 3 to fail with smaller probability $\Delta^{-\Omega(c)}$.

The high degree case. The low degree case handles all Δ that are $\operatorname{poly}\log n$. We now assume Δ is sufficiently large, i.e., $\Delta > \log^{\alpha c} n$, where α is some large universal constant, and we want to design an algorithm such that no vertex joins V_{bad}, and all uncolored vertices are partitioned into R and X, with R having the same $O(c^2)$-degree guarantee as before, and the components induced by X have size $\log^{O(c)} n = \operatorname{poly}\log n$, regardless of Δ. Intuitively, the proof follows along the same lines as the low degree case, but in Phase 1 we first reduce the maximum degree to $\Delta' = \log^{O(c)} n$ and then put any bad vertices that fail to satisfy an invariant into X (rather than V_{bad}). According to the shattering lemma (Lemma 4.1), the components induced by X have size $\operatorname{poly}(\Delta') \log n = \log^{O(c)} n$. The high degree case consists of 13 iterations of $\mathsf{DenseColoringStep}$ (version 2) with the following parameter settings.

	$D^{(i)}$	$L^{(i)}$	$U^{(i)}$	$\delta^{(i)}$
$i \in [9]$	$\Theta\left(\Delta^{\frac{10-i}{10}} \log^{2i-2} \Delta\right)$	$\Theta\left(\Delta^{\frac{11-i}{10}} \log^{2i-3} \Delta\right)$	$\Theta\left(\Delta^{\frac{11-i}{10}} \log^{2i-2} \Delta\right)$	$\Theta\left(\Delta^{-\frac{1}{10}} \log^2 \Delta\right)$
$i = 10$	$\Theta\left(\max\{\log^{18} \Delta, \log n\}\right)$	$\Theta\left(\Delta^{\frac{1}{10}} \log^{17} \Delta\right)$	$\Theta\left(\Delta^{\frac{1}{10}} \log^{18} \Delta\right)$	$\Delta^{-\frac{1}{20}} \log^{-18} \Delta$
$i = 11$	$\Theta(\log n)$	$\Theta\left(\Delta^{\frac{1}{20}} / \log \Delta\right)$	$\Theta\left(\Delta^{\frac{1}{20}}\right)$	$\Delta^{-\frac{1}{20}} \log^{5c} n$
$i = 12$	$\Theta(\log n)$	$\Theta\left(\dfrac{\log^{5c} n}{\log \Delta}\right)$	$\Theta\left(\log^{5c} n\right)$	$\log^{-3c} n$
$i = 13$	c	$\Theta\left(\dfrac{\log^{2c} n}{\log \Delta}\right)$	$\Theta\left(\log^{2c} n\right)$	$\Theta\left(\dfrac{\log \Delta \log \log n}{\log^{2c} n}\right)$

We use the default shrinking rate $\delta^{(i)} = \frac{D^{(i)}\log(L^{(i)}/D^{(i)})}{L^{(i)}}$ for all i except $i \in \{10, 11, 12\}$. 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 differences. In order to have all $\delta^{(i)} \ll 1$, we need to have $\Delta^{1/20} \gg \log^{5c} n$. We proceed under the assumption that $\Delta > \log^{\alpha c} n$ (α is some large universal constant), so this condition is met.

Phase 1. In view of previous calculations, all invariants hold for a cluster S_j ($\mathcal{U}^{(i)}$, $\mathcal{L}^{(i)}$, and $\mathcal{D}^{(i)}$ for $i \in [1, 12]$) with probability at least $1 - \exp(-\Omega(\log n)) = 1 - 1/\mathrm{poly}(n)$, since all parameters $D^{(i)}$, $L^{(i)}$, and $U^{(i)}$ are chosen to be $\Omega(\log n)$. Therefore, no cluster S_j is put in V_{bad} due to an invariant violation, w.h.p.

Phase 2. Consider iteration $i = 12$. 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 6.2, the probability that the external degree or antidegree of $v \in S_j$ is at most c is

$$1 - \binom{D^{(12)}}{c}\left(O\left(\delta^{(12)}\right)\right)^c \geq 1 - \binom{O(\log n)}{c}\left(O\left(\log^{-3c} n\right)\right)^c \geq 1 - (\log n)^{-\Omega(c^2)}.$$

This failure probability is *not* small enough to guarantee that $\mathcal{D}^{(13)}$ holds everywhere, w.h.p. In the high degree case, if a vertex v belongs to a cluster S_j such that $\mathcal{D}^{(13)}$ does not hold, we add the remaining uncolored vertices in S_j (at most $U^{(12)} = O(\log^{5c} n)$ of them) to 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 $N_{\mathrm{out}}(v) \setminus (S_j \cup X)$. 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

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

Size of components in X. To bound the size of each connected component of X, we use the shattering lemma (Lemma 4.1). Define $G' = (V', E')$ as follows. The vertex set V' consists of all vertices in S that remain uncolored at the beginning of 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 4.1 to the graph G', the size of each connected component of X is $O(\text{poly}(\Delta') \log n) = \log^{O(c)} n$, 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'. ∎

The reader may recall that the proofs of Lemmas 4.4 and 4.5 were based on the veracity of Lemma 6.2. The remainder of this section is devoted to proving Lemma 6.2, which bounds the probability that a certain number of vertices remain uncolored by DenseColoringStep (version 2). By inspection of the DenseColoringStep (version 2) pseudocode, a vertex in S_j can remain uncolored for two different reasons:

- it never selects a color because it is not among the $(1 - \delta_j)|S_j|$ participating vertices in step 1, or
- it selects a color in step 1 but is later decolored in step 2 because of a conflict with some vertex in $S_{j'}$ with $j' < j$.

Lemmas 6.3–6.5 analyze different properties of DenseColoringStep (version 2), which are then applied to prove Lemma 6.2. Throughout, we make use of the property that every $\delta_j < 1/K$ for some sufficiently large K.

LEMMA 6.3. *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 $M = (1 - \delta_j)|S_j|$ be the number of participating vertices in step 1. Notice that if v_i is not among the participating 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 a participating vertex. 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

$$
\begin{aligned}
&M - p_i + \delta_j |S_j| - D_j \\
&\geq M - p_i + D_j \log(|S_j|/D_j) - D_j && \text{(constraint on } \delta_j) \\
&= (M - p_i) + D_j(\log(|S_j|/D_j) - 1)
\end{aligned}
$$

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

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

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

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

We now turn to the case $k < M/2$. We imagine choosing the rank vector $(p_1, \ldots, p_k)$ one element at a time. Regardless of the values of $(p_1, \ldots, p_{i-1})$, we always have

$$
\begin{aligned}
&\mathrm{E}\left[\frac{1}{((M-p_i)+D_j(\log(|S_j|/D_j)-1)}\,\middle|\, p_1,\dots,p_{i-1}\right]\\
&\le \frac{1}{M-(i-1)}\sum_{x=0}^{M-i}\frac{1}{x+D_j(\log(|S_j|/D_j)-1)}
\end{aligned}
$$

since there are $M-(i-1)$ choices for p_i and the worst case is when $\{p_1,\dots,p_{i-1}\} = \{1,\dots,i-1\}$. Observe that the terms in the sum are strictly decreasing, which means the average is maximized when $i=k<M/2$ is maximized. Continuing,

$$
\le \frac{1}{M/2}\sum_{x=0}^{M/2}\frac{1}{x+D_j(\log(|S_j|/D_j)-1).}
$$

The sum is the difference between two harmonic sums, and hence

$$
\begin{aligned}
&= O\left(\frac{1}{M}\cdot\big(\log M-\log(D_j(\log(|S_j|/D_j)-1))\big)\right)\\
&= O\left(\frac{\log(|S_j|/D_j)}{|S_j|}\right) \qquad\qquad \text{since } M=\Theta(|S_j|).
\end{aligned}
$$

Therefore, regardless of k, $p^\star \le \big(O\big(\frac{\log(|S_j|/D_j)}{|S_j|}\big)\big)^{|T|}$, as claimed. ∎

LEMMA 6.4. *Let T be any subset of S_j. The probability that all vertices in T are decolored in* DenseColoringStep *(version 2) is $\big(O\big(\frac{D_j\log(|S_j|/D_j)}{|S_j|}\big)\big)^{|T|}$, even allowing the colors selected in $S_1,\dots,S_{j-1}$ 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 $|N_{\mathrm{out}}(v)\setminus S_j| \le |N^\star(v)\setminus S_j| \le D_j$. By Lemma 6.3 (and a union bound over $D_j^{|T|}$ color assignments to T), the probability that all vertices in T are decolored is

$$
D_j^{|T|}\cdot\left(O\left(\frac{\log(|S_j|/D_j)}{|S_j|}\right)\right)^{|T|} = \left(O\left(\frac{D_j\log(|S_j|/D_j)}{|S_j|}\right)\right)^{|T|}.
$$

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 $S_{j+1},\dots,S_g$. The above analysis (which is based on Lemma 6.3) holds, even allowing the colors selected in $S_1,\dots,S_{j-1}$ to be determined adversarially. ∎

LEMMA 6.5. *Let T be any subset of S_j. The probability that no vertex in T selects a color in step 1 of* 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 participate in step 1 with probability δ_j, and the events for two vertices $u,v\in S_j$ to not participate in step 1 are negatively correlated. ∎

Restatement of Lemma 6.2. *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 $\binom{|T|}{t}\cdot(O(\delta))^t$.*

Proof. 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 do not participate in step 1, and all vertices in U_2 are decolored in step 2. Notice that there are at most 2^t ways of partitioning U into $U_1\cup U_2$.

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 on the random bits in S_j. Thus, by Lemma 6.5, the probability that all vertices in U_1 fail to select a color is at most $\prod_{j=1}^{k}(O(\delta_j))^{\left|U_1^{(j)}\right|}\le(O(\delta))^{|U_1|}$. Recall $\delta=\max_{j:S_j\cap T\neq\emptyset}\delta_j$.

We write $U_2^{(j)}=U_2\cap S_j$. Whether a vertex $v\in U_2^{(j)}$ is decolored only depends on 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 $(O(\delta_j))^{\left|U_2^{(j)}\right|}$ by Lemma 6.4. Recall $\delta\ge\delta_j\ge\frac{D_j\log(|S_j|/D_j)}{|S_j|}$. Thus, the probability that all vertices in U_2 are decolored is at most $\prod_{j=1}^{k}(O(\delta_j))^{\left|U_2^{(j)}\right|}\le(O(\delta))^{|U_2|}$.

Therefore, by a union bound over at most $\binom{|T|}{t}$ choices of U and 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 $2^t\cdot\binom{|T|}{t}\cdot(O(\delta))^t=\binom{|T|}{t}\cdot(O(\delta))^t$. This concludes the analysis of DenseColoringStep (version 2). ∎

7. Conclusion. We have presented a randomized $(\Delta+1)$-list coloring algorithm that requires $O(\mathsf{Det}_d(\operatorname{poly}\log n))$ rounds of communication, which is *syntactically* close to the $\Omega(\mathsf{Det}(\operatorname{poly}\log n))$ lower bound implied by Chang, Kopelowitz, and Pettie [20]. Recall that Det and Det_d are the deterministic complexities of $(\Delta+1)$-list coloring and $(\deg+1)$-list coloring. A natural question is whether $(\Delta+1)$-list coloring is *strictly* easier than $(\deg+1)$-list coloring. Answering this question in the negative would imply the randomized optimality of our algorithm. Historically, all advancements in deterministic $(\Delta+1)$-list coloring also applied to $(\deg+1)$-list coloring [5, 48, 9, 10, 12, 8, 27, 11, 35]. Furthermore, when we restrict our attention to algorithms that depend on n (but independent of Δ), only *one* coloring technique has been developed in the last 30 years for $(\Delta+1)$-/$(\deg+1)$-list coloring, namely to use *network decompositions* [5, 39, 43, 48]. So long as network decompositions are the state-of-the-art, it will be difficult to find asymptotically better upper bounds on Det than Det_d. On the lower bound side, progress on round elimination techniques [15, 16, 20, 19, 6] has yielded deterministic $\Omega(\log n)$ lower bounds on *non-greedy* coloring problems such as Δ-vertex coloring or $(2\Delta-2)$-edge coloring, even on trees [16, 20, 19]. The best round elimination lower bounds for greedy coloring problems (e.g., $(\Delta+1)$-coloring) are still $\Omega(\log^* n)$ [38, 42, 15, 7], and they may in fact be tight.

It is an open problem to generalize our algorithm to solve the $(\deg+1)$-list coloring problem, and here it may be useful to think about a problem of intermediate difficulty, at least conceptually. Define $(\deg+1)$-coloring to be the coloring problem

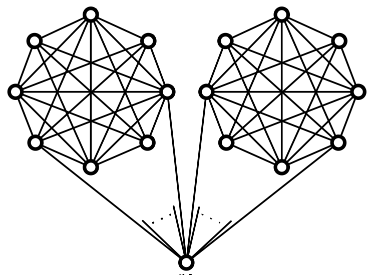

FIG. 3. *An example illustrating the difficulty of* (deg +1)*-list coloring.*

when v's palette is $\{1, \ldots, \deg(v) + 1\}$ (rather than an arbitrary set of $\deg(v) + 1$ colors).[8] Whether the problem is (deg +1)-coloring or (deg +1)-*list* coloring, the difficulty is generalizing the notion of "ϵ-friend edge" and "ϵ-sparse vertex" to graphs with irregular degrees. See Figure 3 for an extreme example illustrating the difficulty of (deg +1)-list coloring. Suppose $N(v)$ is partitioned into sets S_1, S_2 with $|S_1| = |S_2| = |N(v)|/2 = s$. The graphs induced by $S_1 \cup \{v\}$ and $S_2 \cup \{v\}$ are $(s+1)$-cliques, and there are no edges joining S_1 and S_2. The palettes of vertices in S_1 and S_2 are, respectively, $[1, s+1]$ and $[s+1, 2s+1]$.

Notice that v is ϵ-sparse according to our definition (for any $\epsilon < 1/2$) and yet regardless of how we design the initial coloring step, we cannot hope to create more than *one* excess color at v since the two palettes $[1, s+1] \cap [s+1, 2s+1] = \{s+1\}$ only intersect at one color. Thus, it must be wrong to classify v as "ϵ-sparse" since it does not satisfy key properties of ϵ-sparse vertices. On the other hand, if v is to be classified as "ϵ-dense," then it is not clear whether we can recover any of the useful properties of ϵ-dense vertices from Lemma 3.1, e.g., that they form almost cliques with $O(1)$ weak diameter and have external degrees bounded by $O(\epsilon\Delta)$. This particular issue does not arise in instances of the (deg +1)-coloring problem, which suggests that attacking this problem may be a useful conceptual stepping stone on the way to solving (deg +1)-*list* coloring.

Appendix A. Concentration bounds. We make use of some standard tail bounds [23]. 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:

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

Multiplicative Chernoff bounds give the following tail bounds of X with mean

[8] We are aware of one application [3] in distributed computing where the palettes are fixed in this way.

$\mu = np$:

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

Note that Chernoff bounds hold even when X is the summation of n *negatively correlated* 0-1 random variables [24, 23] with mean p, i.e., total independence is not required. The bounds for $\Pr[X \geq (1+\delta)\mu]$ also hold when $\mu > np$ is an overestimate of $\mathrm{E}[X]$. Similarly, the bound for $\Pr[X \leq (1+\delta)\mu]$ also holds when $\mu < np$ is an underestimate of $\mathrm{E}[X]$.

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 $\mu = \mathrm{E}[X]$. Hoeffding's inequality [33] states that

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

Appendix B. Proof of Lemma 2.5. We first recall Lemma 2.5.

Restatement of Lemma 2.5. *Consider the $(\Delta+1)$-list coloring problem. There is an $O(1)$-time algorithm that colors a subset of V such that the following are true for each $v \in V$ with $\deg(v) \geq (5/6)\Delta$:*

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

(ii) *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.*

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 $\{u, v\}$ towards the endpoint with lower ID, 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 2.5 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 2.5.

Consider an execution of OneShotColoring with any constant $p \in (0, 1/4)$. Let v be an ϵ-sparse vertex. Define the following two numbers:

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

$f_2(v)$: the 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 (Lemmas B.3 and B.4), 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 B.3 or Lemma B.4 are satisfied. Our second goal is to show that for each vertex v of degree at least $(5/6)\Delta$, with high probability, at least $(1-1.5p)|N(v)| > (1-(1.5)/4)\cdot(5/6)\Delta > \Delta/2$ neighbors of v remain uncolored after OneShotColoring. This is done in Lemma B.5.

Lemmas B.1 and B.2 establish some generally useful facts about OneShotColoring, which are used in the proofs of Lemmas B.3 and B.4.

Lemma B.1. *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 in step 2 of* OneShotColoring *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 $\Pr[E_c] \le \frac{p|S|}{\Delta+1} < 2p < 1/2$ since $p < 1/4$ and $|S| \le 2\Delta$. Moreover, the collection of events $\{E_c\}$ are negatively correlated [24].

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

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

Lemma B.2. *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 $|\Psi(u_i) \cap Q| \ge (1-\epsilon/2)(\Delta+1)$. Moreover, each $u_i \in S$ is associated with a vertex set R_i such that* (i) *$S \cap R_i = \emptyset$, and* (ii) *$|R_i| \le 2\Delta$. Then, with probability at least $1-\exp(-\Omega(\epsilon^2\Delta))$, there are at least $p\epsilon(\Delta+1)/8$ 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 B.1, $\Pr[E_i^{\text{bad}}] = \exp(-\Omega(|Q_i|)) = \exp(-\Omega(\Delta))$. 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

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

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 $\mu = \mathrm{E}[Y] \leq \frac{p\epsilon}{2} \cdot |S| = \frac{p\epsilon^2\Delta}{4}$. 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 $X - 2Y \geq p\epsilon\Delta/7 - 2p\epsilon^2\Delta/2 > p\epsilon(\Delta+1)/8$. □

Lemma B.3 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 B.3. *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 $X \leq f_1(v)$[9] since if E_i^{good} occurs, then u_i successfully colors itself by some color $c \notin \Psi(v)$. By Lemma B.1, $\Pr[E_i^{\text{bad}}] = \exp(-\Omega(|Q_i|)) = \exp(-\Omega(\epsilon\Delta))$. 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 $j \in [1, i)$. Thus, $\Pr[X \leq t]$ is bounded from above by $\Pr[\text{Binomial}(n', p') \leq t]$, 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, $\Pr[X \leq t] \leq \exp\left(\frac{-(n'p'-t)^2}{2n'p'}\right) = \exp(-\Omega(\epsilon^2\Delta))$. □

Lemma B.4 considers the case that many pairs of neighbors of v are likely to color themselves the same color and contribute to $f_2(v)$. Notice that any ϵ-sparse vertex that does not satisfy the preconditions of Lemma B.3 *does* satisfy the preconditions of Lemma B.4.

[9] In general, X does not necessarily equal $f_1(v)$, since in the calculation of X we only consider the vertices in S, which is a subset of $N(v)$.

LEMMA B.4. *Let v be an ϵ-sparse vertex. Suppose that there is a subset $S \subseteq N(v)$ such that $|S| \geq (1-\epsilon/5)\Delta$, and for each $u \in S$, $|\Psi(u) \cap \Psi(v)| \geq (1-\epsilon/5)(\Delta+1)$. Then $f_2(v) \geq p^3\epsilon^2\Delta/2000$ with probability at least $1-\exp(-\Omega(\epsilon^2\Delta))$.*

Proof. Let $S' = \{u_1, \dots, u_k\}$ be any subset of S such that (i) $|S'| = \frac{p\epsilon\Delta}{100}$, and (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 $S', S_1, \dots, S_k$ 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*-ϵ-friends of v, and for each such nonfriend $u_i \in S'$, we have $|S \setminus (S' \cup N(u_i))| \geq |S| - |S'| - |N(u_i)| \geq \Delta((1-\epsilon/5) - p\epsilon/100 - (1-\epsilon)) > \epsilon\Delta/2$.

Order the set $S' = \{u_1, \dots, u_k\}$ in increasing order by vertex ID. Define $Q_i = \Psi(u_i) \cap \Psi(v)$. 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 $p\epsilon(\Delta+1)/8$.

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

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. 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, $X - Y \leq f_2(v)$, 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 B.2 and the definition of Q_i^{good}), and up to this negligible error 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 $X \geq \frac{p^3\epsilon^2\Delta}{1000}$ with probability $1-\exp(-\Omega(\epsilon^2\Delta))$. It is clear that $\Pr[X_i = 1] \geq p \cdot \frac{p\epsilon(\Delta+1)/8}{\Delta+1} = \frac{p^2\epsilon}{8}$.[10] Thus, $\Pr[X \leq t]$ is bounded from above by $\Pr[\text{Binomial}(n', p') \leq t]$, 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, $\Pr[X \leq t] \leq \exp(\frac{-(n'p'-t)^2}{2n'p'}) = \exp(-\Omega(\epsilon^2\Delta))$.

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

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

LEMMA B.5. *The number of vertices in $N(v)$, the neighborhood of v, that remain uncolored after* OneShotColoring *is at least $(1-1.5p)|N(v)|$ 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 $X \leq 1.5p|N(v)|$ with probability $1-\exp(-\Omega(|N(v)|))$. Since

[10]In the calculation of X, we first reveal all colors selected by vertices in $V \setminus S'$, and then we reveal the colors selected by $u_1, \dots, u_k$ in this order. The value of X_i is determined when the color selected by u_i is revealed. Regardless of the colors selected by vertices in $V \setminus S'$ and $\{u_1, \dots, u_{i-1}\}$, we have $\Pr[X_i = 1] \geq \frac{p^2\epsilon}{8}$.

a vertex participates with probability p,

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

by a Chernoff bound with $\delta = 1/2$. ◻

Acknowledgment. We thank the two reviewers for improving the quality of the paper.

REFERENCES

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

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

[3] A. Amir, O. Kapah, T. Kopelowitz, M. Naor, and E. Porat, *The family holiday gathering problem or fair and periodic scheduling of independent sets*, in Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), 2016, pp. 367–375.

[4] S. Assadi, Y. Chen, and S. Khanna, *Sublinear algorithms for* $(\Delta + 1)$ *vertex coloring*, in Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, 2019, pp. 767–786, https://doi.org/10.1137/1.9781611975482.48.

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

[6] A. Balliu, S. Brandt, J. Hirvonen, D. Olivetti, M. Rabie, and J. Suomela, *Lower bounds for maximal matchings and maximal independent sets*, in Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2019, pp. 481–497.

[7] A. Balliu, J. Hirvonen, D. Olivetti, and J. Suomela, *Hardness of minimal symmetry breaking in distributed computing*, in Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC), 2019, pp. 369–378.

[8] 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), 2015, pp. 345–354.

[9] L. Barenboim and M. Elkin, *Deterministic distributed vertex coloring in polylogarithmic time*, J. ACM, 58 (2011), 5.

[10] L. Barenboim and M. Elkin, *Distributed Graph Coloring: Fundamentals and Recent Developments*, Synth. Lect. Distrib. Comput. Theory 11, Morgan & Claypool, Williston, VT, 2013.

[11] 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 of the 2018 ACM Symposium on Principles of Distributed Computing (PODC), 2018, pp. 437–446.

[12] L. Barenboim, M. Elkin, and F. Kuhn, *Distributed* $(\Delta + 1)$*-coloring in linear (in* Δ*) time*, SIAM J. Comput., 43 (2014), pp. 72–95, https://doi.org/10.1137/12088848X.

[13] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider, *The locality of distributed symmetry breaking*, J. ACM, 63 (2016), 20.

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

[15] S. Brandt, *An automatic speedup theorem for distributed problems*, in Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC), 2019, pp. 379–388.

[16] S. Brandt, O. Fischer, J. Hirvonen, B. Keller, T. Lempiäinen, J. Rybicki, J. Suomela, and J. Uitto, *A lower bound for the distributed Lovász local lemma*, in Proceedings of the 48th ACM Symposium on the Theory of Computing (STOC), 2016, pp. 479–488.

[17] Y. Chang, M. Fischer, M. Ghaffari, J. Uitto, and Y. Zheng, *The complexity of* $(\Delta + 1)$*-coloring in congested clique, massively parallel computation, and centralized local computation*, in Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC), 2019, pp. 471–480.

[18] Y.-J. Chang, *Locality of Distributed Graph Problems*, Ph.D. thesis, University of Michigan, Ann Arbor, MI, 2019.

[19] Y.-J. Chang, Q. He, W. Li, S. Pettie, and J. Uitto, *Distributed edge coloring and a special case of the constructive Lovász local lemma*, ACM Trans. Algorithms, 16 (2019), 8.

[20] Y.-J. Chang, T. Kopelowitz, and S. Pettie, *An exponential separation between randomized and deterministic complexity in the* LOCAL *model*, SIAM J. Comput., 48 (2019), pp. 122–143, https://doi.org/10.1137/17M1117537.

[21] 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), 2018, pp. 445–456.

[22] Y.-J. Chang and S. Pettie, *A time hierarchy theorem for the* LOCAL *model*, SIAM J. Comput., 48 (2019), pp. 33–69, https://doi.org/10.1137/17M1157957.

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

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

[25] 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 of the 2015 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2015, pp. 355–370, https://doi.org/10.1137/1.9781611973730.26.

[26] M. Fischer and M. Ghaffari, *Sublogarithmic distributed algorithms for Lovász local lemma with implications on complexity hierarchies*, in Proceedings of the 31st International Symposium on Distributed Computing (DISC), 2017, 18.

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

[28] M. Ghaffari, *An improved distributed algorithm for maximal independent set*, in Proceedings of the 2016 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2016, pp. 270–277, https://doi.org/10.1137/1.9781611974331.ch20.

[29] A. Goldberg, S. Plotkin, and G. Shannon, *Parallel symmetry-breaking in sparse graphs*, in Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC), 1987, pp. 315–324.

[30] A. V. Goldberg and S. A. Plotkin, *Parallel* $(\Delta + 1)$*-coloring of constant-degree graphs*, Inform. Process. Lett., 25 (1987), pp. 241–245.

[31] D. G. Harris, *Distributed approximation algorithms for maximum matching in graphs and hypergraphs*, in Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2019, pp. 700–724.

[32] D. G. Harris, J. Schneider, and H. Su, *Distributed* $(\Delta{+}1)$*-coloring in sublogarithmic rounds*, J. ACM, 65 (2018), 19.

[33] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, J. Amer. Statist. Assoc., 58 (1963), pp. 13–30.

[34] Ö. Johansson, *Simple distributed* $\Delta + 1$*-coloring of graphs*, Inform. Process. Lett., 70 (1999), pp. 229–232.

[35] F. Kuhn, *Faster deterministic distributed coloring through recursive list coloring*, in Proceedings of the 2020 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2020, pp. 1244–1259, https://doi.org/10.1137/1.9781611975994.76.

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

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

[38] N. Linial, *Locality in distributed graph algorithms*, SIAM J. Comput., 21 (1992), pp. 193–201, https://doi.org/10.1137/0221015.

[39] N. Linial and M. E. Saks, *Low diameter graph decompositions*, Combinatorica, 13 (1993), pp. 441–454.

[40] M. Luby, *A simple parallel algorithm for the maximal independent set problem*, SIAM J. Comput., 15 (1986), pp. 1036–1053, https://doi.org/10.1137/0215074.

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

[42] M. Naor, *A lower bound on probabilistic algorithms for distributive ring coloring*, SIAM J. Discrete Math., 4 (1991), pp. 409–412, https://doi.org/10.1137/0404036.

[43] A. Panconesi and A. Srinivasan, *The local nature of* Δ*-coloring and its algorithmic applications*, Combinatorica, 15 (1995), pp. 255–280.

[44] A. Panconesi and A. Srinivasan, *On the complexity of distributed network decomposition*,

J. Algorithms, 20 (1996), pp. 356–374.

[45] M. Parter, $(\Delta + 1)$ *coloring in the congested clique model*, in Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP), 2018, 160.

[46] M. Parter and H.-H. Su, $(\Delta+1)$ *coloring in* $O(\log^* \Delta)$ *congested-clique rounds*, in Proceedings of the 32nd International Symposium on Distributed Computing (DISC), 2018, 39.

[47] D. Peleg, *Distributed Computing: A Locality-Sensitive Approach*, SIAM, Philadelphia, 2000, https://doi.org/10.1137/1.9780898719772.

[48] V. Rozhoň and M. Ghaffari, *Polylogarithmic-time deterministic network decomposition and distributed derandomization*, in Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC), to appear.

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

[50] V. H. Vu, *A general upper bound on the list chromatic number of locally sparse graphs*, Combin. Probab.& Computing, 11 (2002), pp. 103–111.