# Even Faster $(\Delta + 1)$-Edge Coloring via Shorter Multi-Step Vizing Chains

Sayan Bhattacharya* Martín Costa* Shay Solomon† Tianyi Zhang†

University of Warwick*
Tel Aviv University†

**Abstract**

Vizing's Theorem from 1964 states that any $n$-vertex $m$-edge graph with maximum degree $\Delta$ can be *edge colored* using at most $\Delta + 1$ colors. For over 40 years, the state-of-the-art running time for computing such a coloring, obtained independently by Arjomandi [1982] and by Gabow, Nishizeki, Kariv, Leven and Terada [1985], was $\tilde{O}(m\sqrt{n})$. Very recently, this time bound was improved in two independent works, by Bhattacharya, Carmon, Costa, Solomon and Zhang to $\tilde{O}(mn^{1/3})$, and by Assadi to $\tilde{O}(n^2)$.

In this paper we[1] present an algorithm that computes such a coloring in $\tilde{O}(mn^{1/4})$ time. Our key technical contribution is a subroutine for extending the coloring to one more edge within time $\tilde{O}(\Delta^2 + \sqrt{\Delta n})$. The best previous time bound of any color extension subroutine is either the trivial $O(n)$, dominated by the length of a Vizing chain, or the bound $\tilde{O}(\Delta^6)$ by Bernshteyn [2022], dominated by the length of *multi-step Vizing chains*, which is basically a concatenation of multiple (carefully chosen) Vizing chains. Our color extension subroutine produces significantly shorter multi-step Vizing chains than in previous works, for sufficiently large $\Delta$.

[1] *quasi nanos, gigantium humeris insidentes*

## Contents

## Part I
# Extended Abstract

## 1 Introduction

Let $G = (V, E)$ be a simple, undirected $m$-edge $n$-vertex graph with maximum degree $\Delta$. For an integer $\kappa \in \mathbb{N}^+$, a $\kappa$-edge coloring $\chi : E \to \{1, 2, \ldots, \kappa\}$ of the graph $G$ assigns a *color* $\chi(e)$ to every edge $e \in E$, so that any two adjacent edges receive distinct colors. The minimum $\kappa$ for which the graph $G$ admits a $\kappa$-edge coloring, or the *edge chromatic number* of $G$, cannot be smaller than $\Delta$. On the other hand, Vizing's Theorem states that $\Delta + 1$ colors are always sufficient [Viz64]

Vizing's original proof for the existence of $(\Delta + 1)$-edge coloring can rather easily be converted into an $O(mn)$ time algorithm. In two independent works from the 80s, Arjomandi [Arj82] and Gabow et al. [GNK+85] improved this runtime bound to $\tilde{O}(m\sqrt{n})$.[2] Recently, Sinnamon [Sin19] used randomization to achieve a clean bound of $O(m\sqrt{n})$. There has been no polynomial improvement over this $m\sqrt{n}$ time barrier in over 40 years, until very recently, where this time bound was improved in two independent works. Bhattacharya, Carmon, Costa, Solomon and Zhang [BCC+24] improved the time bound to $\tilde{O}(mn^{1/3})$, which provides a polynomial improvement over the $m\sqrt{n}$ time bound in the entire regime of parameters. [Ass24] achieved a time bound of $\tilde{O}(n^2)$, which in particular provides a near-linear time algorithm for dense graphs; refer to Section 1.1 for additional results in [Ass24], where more than $\Delta + 1$ colors are used. Both algorithms of [BCC+24] and [Ass24] are randomized and the running time bounds hold with high probability. Remarkably, the approaches in [BCC+24] and in [Ass24] are inherently different. In a nutshell, the key contribution of [BCC+24] is in speeding up the coloring of the last few edges whereas the key contribution of [Ass24] is in speeding up the coloring of all but the last few edges.

Despite this exciting recent progress, the following outstanding question remains open.

How fast can one compute a $(\Delta + 1)$-edge coloring of an input $m$-edge $n$-vertex graph?

By combining the results of [BCC+24] and [Ass24], one can directly get a running time of $\tilde{O}(n\sqrt{m})$.[3] Other than this direct corollary of the combination of the two works [BCC+24, Ass24], it is unclear whether any further improvement is possible: This is exactly where the contribution of the current paper lies. Specifically, we[4] prove the following theorem.

**Theorem 1.1.** *Given a simple, undirected $m$-edge $n$-vertex graph $G = (V, E)$ with maximum degree $\Delta$, we can compute a $(\Delta + 1)$-edge coloring of $G$ in $\tilde{O}(mn^{1/4})$ time with high probability.*

Our proof of this theorem builds on the recent works of [BCC+24, Ass24], as well as on a sequence of earlier works [DHZ19, Ber22, Chr23]. Our key technical contribution is a subroutine for extending a partial coloring to one more edge within time $\tilde{O}(\Delta^2 + \sqrt{\Delta n})$. The best previous time bound of any color extension subroutine is either the trivial $O(n)$, dominated by the length of a Vizing chain, or the bound $\tilde{O}(\Delta^6)$ by Bernshteyn [Ber22], dominated by the length of *multi-step Vizing chains*, which is basically a concatenation of multiple (carefully chosen) Vizing chains. There is also a much faster color extension subroutine, by Duan et al. [DHZ19]: it uses $(1 + \epsilon)\Delta$ colors within time $\tilde{O}(1/\epsilon^2)$, provided that $\Delta = \tilde{\Omega}(1)$ and $\epsilon = \tilde{\Omega}(1/\sqrt{\Delta})$; we note that the minimum number of colors

[2] We use the notation $\tilde{O}(\cdot)$ throughout to suppress polylogarithmic in $n$ factors.
[3] This observation was made via personal communication with the author of [Ass24].
[4] *quasi nanos, gigantium humeris insidentes*

achievable by [DHZ19] is $\Delta + \tilde{O}(\sqrt{\Delta})$, and the respective runtime is $\tilde{O}(\Delta)$. Even slightly reducing the number of colors below $\Delta + \tilde{O}(\sqrt{\Delta})$, while allowing a higher time of $\tilde{O}(\Delta^2)$, is currently out of reach.

Our color extension subroutine settles for the aforementioned higher running time of $\tilde{O}(\Delta^2 + \sqrt{\Delta n})$; even then, it has to drill much deeper than [DHZ19], since the transition from $\Delta + \tilde{O}(\sqrt{\Delta})$ colors to $\Delta + 1$ has to overcome numerous nontrivial technical hurdles. Ultimately, we manage to produce significantly shorter multi-step Vizing chains than in previous works for $\Delta + 1$ colors, in the regime that $\Delta$ is sufficiently large; we then demonstrate the usefulness of such a color extension subroutine in proving Theorem 1.1. A comprehensive technical overview is given in Section 2.

### 1.1 Related Work

If we allow for (sufficiently) more colors than $\Delta + 1$, then the problem becomes (dramatically) simpler. First, it was known since the 80s that, for $\Delta + \tilde{O}(\sqrt{\Delta})$ colors, the problem can be solved in $\tilde{O}(m)$ time [KS87]. A recent line of works provided algorithms with near-linear runtime for $(1+\epsilon)\Delta$-edge coloring [DHZ19, BCPS24b, EK24], culminating with the recent result of [Ass24] for $(\Delta + O(\log n))$-edge coloring in $O(m \log \Delta)$ time and a $(1+\epsilon)\Delta$-edge coloring in $O(m \log(1/\epsilon))$ time for any $\epsilon = \omega(\log n/\Delta)$.

There is a large body of work on edge coloring in restricted graph classes. First, for bipartite graphs one can compute a $\Delta$-edge coloring in $\tilde{O}(m)$ time [CH82, COS01, Alo03]. One can compute a $(\Delta+1)$-edge coloring in $\tilde{O}(m\Delta)$ time [GNK+85] for bounded degree graphs. This result of [GNK+85] from the 80s was generalized recently for bounded arboricity graphs [BCPS23], and there has been further recent work on edge coloring in bounded arboricity graphs [BCPS24a, CRV24, Kow24]. Refer to [CY89, CN90, CK08] and the references therein for works on edge coloring in planar graphs, bounded treewidth graphs and bounded genus graphs.

Finally, the edge coloring problem is receiving an extensive and growing research attention in various computational models other than the classic static sequential setting, including in dynamic algorithms [BM17, BCHN18, DHZ19, Chr23, BCPS24b, Chr24], distributed algorithms [PR01, EPS14, FGK17, GKMU18, BBKO22, CHL+20, Ber22, Chr23, Dav23], online algorithms [CPW19, BGW21, SW21, KLS+22, BSVW24, DGS24], and streaming algorithms [BDH+19, BS23, CMZ23, GS23].

### 1.2 Organization of the Rest of the Paper

To prove Theorem 1.1, we use 3 different subroutines, which we formally describe in Part II of our paper. In Section 2, we give a technical overview of our algorithm and analysis. In Sections 3 and 4, we sketch the analysis of the subroutines that we use to prove Theorem 1.1. In Part II, we provide the full version of our paper.[5]

## 2 Overview of Our Techniques

Our key contributions are to design an efficient algorithm for the following subroutine, and to show that it leads to Theorem 1.1 in combination with the machinery developed in [BCC+24] and [Ass24].

**$(\Delta + \eta)$-Color Extension Subroutine:** Here, $\eta \geq 1$ is an integer. The input to this subroutine is a graph $G = (V, E)$ with maximum degree $\Delta$, a given edge $e \in E$, and a valid coloring $\chi_{\texttt{init}} : E \setminus \{e\} \to [\Delta + \eta]$ of $G \setminus \{e\}$. The subroutine has to *extend* the *partial* coloring $\chi$ to the entire

[5]See the first paragraph in Part II for the organization of the full version of our paper.

graph $G$, by assigning some color $c \in [\Delta + \eta]$ to the uncolored edge $e \in E$ and possibly changing the colors of some edges in $E \setminus \{e\}$. Let $\chi_{\texttt{final}} : E \to [\Delta + \eta]$ be the valid $(\Delta + \eta)$-edge coloring of $G$ when the subroutine finishes execution. We define the **cost** incurred by the subroutine to be the number of edges in $G$ that change their colors during this process, i.e., the cost equals $1 + |\{e' \in E \setminus \{e\} : \chi_{\texttt{init}}(e') \neq \chi_{\texttt{final}}(e')\}|$. It is easy to observe that **the runtime of any such subroutine is at least its cost**, because the subroutine needs to spend $\Omega(1)$ time to change the color of any given edge. We now summarize a lower bound derived by Chang et al. [CHL+20].

**Theorem 2.1** ([CHL+20]). *Any $(\Delta + \eta)$-color extension subroutine has cost $\Omega((\Delta/\eta)\log(\eta n/\Delta))$.*

**Why Is This Useful?** Suppose that we had a $(\Delta + 1)$-color extension subroutine whose runtime matched the lower bound of Theorem 2.1, up to polylogarithmic factors. Then this would imply an algorithm for $(\Delta + 1)$-edge coloring that runs in $\tilde{O}(m)$ time! Below, we explain this in more detail.

We start by applying a well-known *Eulerian partition* technique [Arj82, GNK+85, Sin19]. In $\tilde{O}(m)$ time, this allows us to partition the input graph $G = (V, E)$ into two (almost) equal-sized subgraphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, such that maximum degee in each of these subgraphs is $\leq \lceil \Delta/2 \rceil$. We then recursively color $G_1$ and $G_2$ using two mutually *disjoint* palettes, each of size $\lceil \Delta/2 \rceil + 1$. This gives us a $2 \cdot (\lceil \Delta/2 \rceil + 1) \leq (\Delta + 3)$-edge coloring of $G$. We next uncolor those edges in $G$ that belong to the two *least popular* color classes. By a simple averaging argument, this leads to a partial $(\Delta + 1)$-edge coloring of $G$ with $O(m/\Delta)$ uncolored edges. We next scan through these uncolored edges, and extend the partial $(\Delta + 1)$-edge coloring to them one at a time by applying the $(\Delta + 1)$-color extension subroutine. Note that we make $O(m/\Delta)$ calls to the color extension subroutine, each call taking $\tilde{O}(\Delta)$ time. Thus, the total time spent on all these calls is $O(m/\Delta) \cdot \tilde{O}(\Delta) = \tilde{O}(m)$. At the end of the scan, we have a $(\Delta + 1)$-edge coloring of $G$. The overall runtime is captured by the following recurrence, whose solution is $T(m, \Delta) = \tilde{O}(m)$.

$$T(m, \Delta) = 2 \cdot T\left(\lceil m/2 \rceil, \lceil \Delta/2 \rceil\right) + \tilde{O}(m). \tag{1}$$

Accordingly, a natural line of attack on the problem is to design an efficient $(\Delta + 1)$-color extension subroutine. For this plan to work, however, we first need to address the following concern.

What happens if we have a $(\Delta + 1)$-color extension subroutine that runs in, say, $\tilde{O}(\Delta^2)$ time (which is reasonably fast, but falls short of matching the lower bound of Theorem 2.1)? Unfortunately, if we plug in such a color extension subroutine in the framework described above, then we spend $\tilde{O}(\Delta^2) \cdot O(m/\Delta) = \tilde{O}(m\Delta)$ total time to extend the coloring to the last $O(m/\Delta)$ uncolored edges. So, the last term in the RHS of (1) becomes $\tilde{O}(m\Delta)$ instead of $\tilde{O}(m)$. Thus, using this approach, the time taken to compute a $(\Delta + 1)$-edge coloring is now $\Omega(m\Delta)$, which is no better than the existing $\tilde{O}(m\Delta)$ time algorithms by [Arj82, GNK+85, Sin19]. So the concern is that the above framework is not *robust*: It does not give any *polynomial advantage* over the current state-of-the-art, if our $(\Delta + 1)$-color extension subroutine takes $\Omega(\Delta^2)$ time. To address this concern, we synthesize the main technical insights from [BCC+24], and prove the following lemma.

**Lemma 2.2.** *Consider a partial coloring $\chi : E \setminus E^\star \to [\Delta + 1]$ of a graph $G = (V, E)$, where $E^\star \subseteq E$ is the set of uncolored edges. Let $U^\star \subseteq V$ be a vertex cover of $G^\star := (V, E^\star)$. Then there is a randomized algorithm that* extends *$\chi$ to $\Omega(|E^\star|)$ edges of $E^\star$ in $\tilde{O}(|E^\star|\Delta + \Delta m|U^\star|/|E^\star|)$ expected time. Specifically, the algorithm terminates with a partial coloring $\chi' : E \setminus E' \to [\Delta + 1]$ of $G$, with $E' \subseteq E$ being the set of uncolored edges, such that $E' \subseteq E^\star$ and $|E^\star \setminus E'| = \Omega(|E^\star|)$.*

The above lemma shows how to efficiently extend a partial $(\Delta + 1)$-edge coloring to a constant fraction of a *batch* of uncolored edges, when they admit a small vertex cover. For comparison, as implicit in [BCC+24], the runtime bound for the same task is

$$\tilde{O}\left(|E^\star|\Delta + \min_{\tau\geq 1}\left\{\frac{\Delta m|U^\star|\tau}{|E^\star|} + \frac{|E^\star|n}{\tau}\right\}\right),$$

which is always worse than the new bound we derive in Lemma 2.2; refer to Section 6 (see the remark following Lemma 6.1) for a detailed discussion on this matter. In addition, our proof of Lemma 2.2 is arguably much simpler and more intuitive than the analysis in [BCC+24]; we defer the complete, self-contained proof of this lemma to Section 9.

We are now ready to address the concern we pointed out after recurrence (1). Lemma 2.2, along with the recent breakthrough result of [Ass24], implies Lemma 2.3 stated below. This gives us the desired tradeoff between the runtime of a $(\Delta+1)$-color extension subroutine and the runtime of a $(\Delta+1)$-edge coloring algorithm. Section 3 outlines the main idea behind the proof of Lemma 2.3.

**Lemma 2.3.** *Let there be a $(\Delta+1)$-color extension subroutine with $\tilde{O}(\Delta^\gamma)$ expected runtime, for some $\gamma \geq 1$. Then there is a $(\Delta+1)$-edge coloring algorithm with $\tilde{O}\big(mn^{(\gamma-1)/(2\gamma)}\big)$ expected runtime.*

If we set $\gamma = 1$ in the above lemma, then as expected it leads to a $\tilde{O}(m)$ time $(\Delta+1)$-edge coloring algorithm; and as long as $\gamma < 3$, we get a polynomial improvement over the state-of-the-art $\tilde{O}(mn^{1/3})$ time bound of [BCC+24]. Thus, ideally we would like to obtain a $(\Delta+1)$-color extension subroutine with (say) an expected runtime of $\tilde{O}(\Delta^{2.99})$. But this brings us in front of a significant technical challenge, as explained below.

**A Major Challenge:** In recent years, an influential line of work [DHZ19, SV19, CHL+20, GP20, Ber22, Chr23] has addressed the question of finding a *small augmenting subgraph* to extend a partial coloring to an uncolored edge, primarily from a different vantage point of distributed algorithms. Many of these results were obtained using *multi-step Vizing chains*, a generalization of the central object used in Vizing's original proof [Viz64]. Couched in our language, this is closely related to designing a color extension subroutine with small *cost*. In particular, for a $(\Delta+1)$-color extension subroutine, the current state-of-the-art bounds are by [Chr23] and [Ber22], who respectively achieve $O(\Delta^7 \log n)$ and $O(\Delta^6 \log^2 n)$ costs using multi-step Vizing chains. It therefore remains an outstanding open question to design such a subroutine with a cost, and moreover with a *runtime*, of $\tilde{O}(\Delta^{2.99})$, as demanded by Lemma 2.3. We demonstrate that this barrier can be *bypassed* by instead achieving the time bound summarized in the theorem below; in fact, proving this theorem is our main technical contribution.

**Theorem 2.4.** *Given a graph $G = (V, E)$ and a partial $(\Delta+1)$-edge coloring $\chi$ of $G$ with an uncolored edge $e \in E$, we can extend $\chi$ to the edge $e$ in $\tilde{O}(\Delta^2 + \sqrt{\Delta n})$ expected time.*

We present a detailed outline of the proof of Theorem 2.4 in Section 4. For now, we focus on emphasizing the following two conceptual, take-home messages. First, Theorem 2.4 is sufficient for making a polynomial improvement over the state-of-the-art $\tilde{O}(mn^{1/3})$ runtime for computing a $(\Delta+1)$-edge coloring [BCC+24]. As a simple sanity check, consider the following two cases.

(i) $\Delta < n^{1/4}$. Here, we can apply any existing $\tilde{O}(m\Delta)$ time algorithm [Arj82, GNK+85, Sin19] to compute a $(\Delta+1)$-edge coloring in $\tilde{O}(mn^{1/4})$ time.

(ii) $\Delta > n^{1/4}$. Here, it is easy to verify that $\Delta^{2.5} \geq \sqrt{\Delta n}$, and hence Theorem 2.4 implies a $(\Delta+1)$-color extension subroutine with $\tilde{O}(\Delta^{2.5})$ expected runtime. So we can set $\gamma = 2.5$ in Lemma 2.3, to get a $(\Delta+1)$-edge coloring algorithm with an expected runtime of $\tilde{O}(mn^{3/10})$.

Thus, in both cases we obtain a polynomial improvement over the previous $\tilde{O}(mn^{1/3})$ runtime bound of [BCC+24]. In Section 7, we perform a more refined analysis, and derive that Theorem 2.4 actually implies a $(\Delta+1)$-edge coloring algorithm with $\tilde{O}(mn^{1/4})$ expected runtime (see Theorem 7.3). This leads us to our main result, as stated in Theorem 1.1.

The second message we wish to emphasize is this: The starting point of our approach for proving Theorem 2.4 is the multi-step Vizing chain construction of [DHZ19]. Specifically, [DHZ19] designed a $((1+\epsilon)\Delta)$-color extension subroutine with an expected runtime of $\tilde{O}(1/\epsilon^2)$, as long as $\Delta = \tilde{\Omega}(1)$ and $\epsilon = \tilde{\Omega}(1/\sqrt{\Delta})$. A priori, it seems that such a bound will not be suitable for our purpose, because we want to set $\epsilon = 1/\Delta$. We now outline, at a very high level, the algorithm of [DHZ19] and how we build on top of it.

*In the ensuing paragraphs, we assume that the reader is already familiar with the proof of Vizing's theorem and concepts like "alternating paths", "fans" and "Vizing chains".*

**The [DHZ19] Algorithm:** We are given an input graph $G = (V, E)$, and a partial $((1+\epsilon)\Delta)$-edge coloring $\chi$ in $G$ with one uncolored edge $e = (u, v)$. The algorithm of [DHZ19] proceeds in $R = \Theta(\log n)$ *rounds*. For each round $i \in \{1, \ldots, R\}$, it samples a subset of colors $\mathcal{C}_i \subseteq [\Delta+1] \setminus \left(\bigcup_{j<i} \mathcal{C}_j\right)$ of size $|\mathcal{C}_i| = \Theta(\log n/\epsilon)$ independently and u.a.r. We refer to $\mathcal{C}_i$ as the *palette* for round $i$. For technical reasons, [DHZ19] want these palettes across different rounds to be mutually disjoint. In addition, for each $i \in [R]$, they want that under any fixed partial coloring $\chi'$ of $G$, every vertex $v$ has at least one *missing color* in $\mathcal{C}_i$.[6] To ensure these two properties, it is easy to verify that we must have $|R| \cdot |\mathcal{C}_i| = \Theta(\log^2 n/\epsilon) < \epsilon\Delta$, the RHS being the *slack*, in terms of the number of extra available colors. This necessitates the requirement that $\epsilon^2 = \tilde{\Omega}(1/\Delta)$, and hence $\epsilon = \tilde{\Omega}(1/\sqrt{\Delta})$.

Let $u_1 := u$, $v_1 := v$, $e_1 := (u_1, v_1)$ and $\chi_1 := \chi$. At the start of a given round $i \in [R]$, [DHZ19] have a partial coloring $\chi_i$ and an uncolored edge $e_i = (u_i, v_i)$ w.r.t. $\chi_i$. Using only the colors from the palette $\mathcal{C}_i$, they identify a Vizing chain starting from an endpoint (say) $u_i$ of $e_i$. If the alternating path corresponding to this Vizing chain has length less than $L := \tilde{\Theta}(1/\epsilon^2)$, then they *apply* the Vizing chain to extend the partial coloring to $e_i$, and their algorithm terminates. Otherwise, they pick some $\alpha_i \in [L]$ independently and u.a.r., and *shift* the position of the uncolored edge *from* $e_i$ *to* the $\alpha_i^{th}$ edge (say) $e_{i+1} = (u_{i+1}, v_{i+1})$ on the concerned alternating path (say) $P_i$. This is done by appropriately shifting the colors on the Vizing fan and the first $\alpha_i$ edges of $P_i$, which leads to a new partial coloring $\chi_{i+1}$. The algorithm then proceeds to round $(i+1)$.

The choice of the value of $L$ is dictated by the fact that for technical reasons, [DHZ19] have to ensure that $L = \Omega(|\mathcal{C}_i|^2)$. The main technical contribution of [DHZ19] is to show that this algorithm terminates within $R$ rounds, w.h.p.[7]

**Our Approach:** At a very high-level, we diverge from [DHZ19] in two major aspects. First, since we need to set $\epsilon = 1/\Delta$, we cannot afford to have these separate palettes $\{\mathcal{C}_i\}_i$ across different rounds, and so we get rid of them altogether. Morally, we observe that all we need is the following *key property*: The pair of colors on the concerned alternating path in each round $i \in [R]$ is disjoint from the ones used in previous rounds. As long as this key property holds, the [DHZ19] analysis continues to work just fine. (This assertion has a big caveat associated with it, namely, the complications that arise from the Vizing fans; see Section 4.5 for a discussion on those complications. But we ignore the fans for now.)

[6]We say that a color is missing at $v$ if it is *not* assigned to any of the edges incident on $v$. Since $v$ has degree at most $\Delta$, it has at least $\epsilon\Delta$ missing colors.

[7]The in-expectation guarantee follows because if the algorithm fails, then it can always revert back to the trivial implementation of Vizing's proof to extend the partial coloring to one edge in $O(n)$ time.

The second point of divergence from [DHZ19] arises because the key property might not hold after a certain number of rounds. To address this issue, our main insight is to increase the value of the parameter $L$ to be $:= \tilde{\Theta}(1/\epsilon^2 + \sqrt{\Delta n}) = \tilde{\Theta}(\Delta^2 + \sqrt{\Delta n})$. With this new increased value of $L$, we derive the following crucial implication (see Lemma 4.4). Let $i \in [R]$ be the first round such that: most of the $L$ choices for the value of $\alpha_i$ lead to a scenario where the subsequent round $i+1$ will not satisfy the key property. Then with sufficiently large probability, the algorithm will terminate in the next round (i.e., in round $i+1$). This leads us to a *win-win* framework. Either round $i$ is such that with good probability we can continue to apply the analysis of [DHZ19] in the subsequent round, or with good probability our algorithm actually terminates in the subsequent round. Section 4 contains a much more detailed exposition of this analysis.

Finally, we need to overcome further significant obstacles to deal with the Vizing fans. The complete proof of Theorem 2.4, which handles these obstacles, is deferred to Section 8.[8]

## 3 Proof (Sketch) of Lemma 2.3

By applying Lemma 2.2 at most $O(\log n)$ times, where in each of these applications we have at most $|E^*|$ and at least $|L|$ uncolored edges, we derive the following corollary.

**Corollary 3.1.** *Suppose that we receive as input: (i) a partial $(\Delta+1)$-coloring $\chi$ of $G$ as specified in Lemma 2.2 and (ii) a parameter $L \in \{1, \dots, |E^\star|\}$. Then in $\tilde{O}(|E^\star|\Delta + \Delta m|U^\star|/L)$ expected time, we can obtain a partial $(\Delta+1)$-coloring $\chi'' : E \setminus E'' \to [\Delta+1]$ of $G$, with $E''$ being the set of uncolored edges, such that $E'' \subseteq E^\star$ and $|E''| \leq L$.*

To convey the main idea behind the proof of Lemma 2.3, we will assume that the input graph $G = (V, E)$ is *almost* $\Delta$-regular, that is, the degree of each vertex $v \in V$ is $\Omega(\Delta)$ and hence $m = \Theta(n\Delta)$. We also assume that $\Delta = \omega(\log n)$; otherwise, we can apply an existing algorithm [Arj82, GNK+85, Sin19] to compute a $(\Delta+1)$-edge coloring of $G$ in $\tilde{O}(m\Delta) = \tilde{O}(m)$ time.

We sample each vertex $v \in V$ into a set $U^\star \subseteq V$ independently with probability $(\kappa \log n)/\Delta$, for a sufficiently large constant $\kappa > 1$. Let $E^\star := \{(u, v) \in E : \{u, v\} \cap U^\star \neq \emptyset\}$ denote the set of edges incident on $U^\star$, and let $E_0 := E \setminus E^\star$ denote the set of remaining edges in $G$. Define the subgraphs $G^\star := (V, E^\star)$ and $G_0 := (V, E_0)$.

Consider any vertex $v \in V$. Its degree in $G$ is at most $\Delta$, and each of its neighbors gets sampled in $U^\star$ with probability $(\kappa \log n)/\Delta$. Thus, applying standard Chernoff bounds, we infer that the degree of $v$ in $G_0$ is at most $(\Delta - \kappa' \log n)$ w.h.p., for a sufficiently large constant $\kappa' > 1$ that depends on $\kappa$. Taking a union bound over all $v \in V$, we get that whp the maximum degree in $G_0$ is at most $(\Delta - \kappa' \log n)$. Based on this observation, we now compute a $(\Delta+1)$-coloring $\chi$ of $G_0$ in $\tilde{O}(m)$ time, by invoking the algorithm of [Ass24]. Note that $\chi$ is a partial $(\Delta+1)$-edge coloring of the entire graph $G$, with $E^\star$ being the set of uncolored edges and $U^\star$ being a vertex cover of $G^\star := (V, E^\star)$, just as in the statement of Lemma 2.2.

At this point, we fix an $L \in \{1, \dots, |E^\star|\}$ whose value will be determined later. Next, we apply Corollary 3.1 on $(\chi, G, U^\star, E^\star, L)$ to obtain a partial coloring $\chi'' : E \setminus E'' \to [\Delta+1]$, with $E'' \subseteq E$ being the set of remaining uncolored edges and $|E''| \leq L$. It now remains to extend the partial $(\Delta+1)$-coloring $\chi''$ to the edges in $E''$. Towards this end, we fork into one of the following cases.

**Case (i): $\Delta^\gamma > n$.** In this case, we scan through the edges in $E''$. While considering an edge $e \in E''$ during this scan, we extend the current partial coloring to $e$, *using a $(\Delta+1)$-color extension subroutine implied by Vizing's original proof which runs in $O(n)$ time.* Thus, overall we spend $O(n|E''|) = O(nL)$ time to extend the partial $(\Delta+1)$-coloring $\chi''$ to all the edges in $E''$.

[8] The proof in Section 8 is self-contained and uses slightly different notation than the proof sketch in Section 4.

**Case (ii): $\Delta^\gamma \leq n$.** Here, the basic set up remains the same as in Case (i) above, except the following: While considering an edge $e \in E''$ during the scan, *we apply the* $(\Delta+1)$*-color extension subroutine that runs in* $\tilde{O}(\Delta^\gamma)$ *expected time.* Thus, overall we spend $\tilde{O}(\Delta^\gamma |E''|) = \tilde{O}(\Delta^\gamma L)$ expected time to extend the partial $(\Delta+1)$-coloring $\chi''$ to all the edges in $E''$.

It is easy to see that at the end of the above process, we obtain a $(\Delta+1)$-edge coloring of the entire graph $G = (V, E)$. We now focus on analyzing the runtime of this algorithm. We first recall that each vertex $v \in V$ is sampled into $U^\star$ with probability $(\kappa \log n)/\Delta$. Thus, by standard Chernoff bounds we have the following guarantees w.h.p.

$$|U^\star| = \tilde{O}(n/\Delta), \text{ and hence } |E^\star| \leq |U^\star|\Delta = \tilde{O}(n). \tag{2}$$

We have already observed that it takes $\tilde{O}(m)$ time to compute the partial coloring $\chi$. Given $\chi$, Corollary 3.1 allows us to obtain the partial coloring $\chi''$ in expected time $\tilde{O}(|E^\star|\Delta + \Delta m|U^\star|/L) = \tilde{O}(n\Delta + mn/L) = \tilde{O}(m + n^2\Delta/L)$. The first equality holds because of (2), and the second inequality holds because we have assumed that the input graph is almost regular. Accordingly, the overall runtime of the algorithm depends on whether we are in Case (i) or Case (ii), as follows.

If we are in Case (i), then the total expected runtime of the algorithm is given by $T := \tilde{O}(m) + \tilde{O}(m + n^2\Delta/L) + O(nL)$. Setting $L = \sqrt{n\Delta}$, we get

$$T = \tilde{O}(m + n\sqrt{n\Delta}) = \tilde{O}(m + n\Delta\sqrt{n/\Delta}) = \tilde{O}(m\sqrt{n/\Delta}) = \tilde{O}\left(mn^{(\gamma-1)/(2\gamma)}\right),$$

where the last inequality holds because $\Delta^\gamma > n$ (which implies that $1/\Delta < 1/n^{1/\gamma}$).

In contrast, if we are in Case (ii), then the total expected runtime of the algorithm is given by $T := \tilde{O}(m) + \tilde{O}(m + n^2\Delta/L) + \tilde{O}(\Delta^\gamma L)$. Setting $L = n/\Delta^{(\gamma-1)/2}$, we get

$$T = \tilde{O}\left(m + n\Delta^{(\gamma+1)/2}\right) = \tilde{O}\left(m + n\Delta \cdot \Delta^{(\gamma-1)/2}\right) = \tilde{O}\left(m\Delta^{(\gamma-1)/2}\right) = \tilde{O}\left(mn^{(\gamma-1)/(2\gamma)}\right).$$

where the last inequality holds because $\Delta^\gamma \leq n$. This concludes the proof (sketch) of Lemma 2.3.

## 4 Proof (Sketch) of Theorem 2.4

We define the parameters:

$$\ell := 100 \log n, \quad L := 10^3 \ell^2 (\Delta^2 + \sqrt{\Delta n}) \tag{3}$$

To convey the main conceptual ideas behind our analysis, in this section we will explain the proof of Theorem 2.4 under Assumption 4.1, stated below. This assumption clearly holds, for example, on bipartite graphs. For those readers familiar with the proof of Vizing's theorem, this assumption will allow us to ignore the Vizing fans altogether, and we will be able to focus only on the alternating paths, which are more intuitive to reason about.

**Assumption 4.1.** *The graph $G = (V, E)$ does not contain any odd cycle of length $\leq 2L + 3$.*

In Section 4.1, we introduce some key notations and terminologies regarding alternating paths. Next, we describe our randomized algorithm in Section 4.2. In Section 4.3, we introduce a recursion tree (which we refer to as the "meta-tree") which encodes all possible execution-paths of our algorithm, and present a few basic properties of this meta-tree. We show that our algorithm essentially performs a random walk on this meta-tree, and state Lemma 4.2 which guarantees that this random walk terminates quickly. We prove Lemma 4.2, which implies Theorem 2.4, in Section 4.4. Finally, we conclude our discussion in Section 4.5 by pointing out the remaining significant technical challenges that we need to overcome, if we are to get rid of Assumption 4.1.

## 4.1 Preliminaries

**Alternating Paths and Types:** Consider a partial $(\Delta+1)$-edge coloring $\chi : E \to [\Delta+1] \cup \{\perp\}$ in the input graph $G = (V, E)$. Consider a path $P = ((v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k))$ in $G$, where $(v_{i-1}, v_i) \in E$ is the $i^{th}$ edge on the path, for all $i \in [k]$. We say that $P$ is an **alternating path** (w.r.t. $\chi$) iff there exist two distinct colors $c, c' \in [\Delta+1]$ that satisfy the following conditions.

1. The colors on the consecutive edges of $P$ alternate between $c$ and $c'$. Specifically, this means that $\chi(v_{i-1}, v_i) \in \{c, c'\}$ for all $i \in [k]$, and $\chi(v_{i-1}, v_i) \neq \chi(v_i, v_{i+1})$ for all $i \in [k-1]$.

2. The path $P$ is *maximal*. Thus, for each vertex $u \in \{v_0, v_k\}$, either $c \in \mathtt{miss}_\chi(u)$ or $c' \in \mathtt{miss}_\chi(u)$, where $\mathtt{miss}_\chi(u) \subseteq [\Delta+1]$ denotes the set of *missing colors* at $u$ under $\chi$ (i.e., these are the colors that are *not* assigned to any edge in $G$ incident on $u$).

Let $\tau = \{c, c'\}$. We refer to $\tau$ as being the **type** of the alternating path $P$. We let $\mathtt{length}(P) = k$ denote the length of the path $P$. We also say that the path $P$ **starts** at $v_0$ and **ends** at $v_k$. For $i \in [k]$, we let $P_{\leq i} = ((v_0, v_1), (v_1, v_2), \ldots, (v_{i-1}, v_i))$ denote the **length-$i$ prefix** of $P$. For $i > k$, we let $P_{\leq i} := P$. Finally, note that an alternating path always comes with an associated **orientation**. In particular, there is another alternating path with the same set of edges as $P$, but in reverse order.

Our algorithm in Section 4.2 will use two basic subroutines, as described below.

**The Subroutine Apply$(\chi, e, P)$:** Here, the input is a partial $(\Delta+1)$-edge coloring $\chi$ of $G$, an uncolored edge $e = (u, v)$ and an alternating path $P$ starting from $u$ that is of type $\tau = \{\alpha_u, \alpha_v\}$, where $\alpha_u \in \mathtt{miss}_\chi(u) \setminus \mathtt{miss}_\chi(v)$ and $\alpha_v \in \mathtt{miss}_\chi(v) \setminus \mathtt{miss}_\chi(u)$. (If either $\alpha_u \in \mathsf{miss}_\chi(u) \cap \mathsf{miss}_\chi(v)$ or $\alpha_v \in \mathsf{miss}_\chi(u) \cap \mathsf{miss}_\chi(v)$, our algorithm will not make the call $\mathsf{Apply}(\chi, e, P)$.) Crucially, it is guaranteed that $\mathtt{length}(P) \leq 2L + 2$. W.l.o.g., suppose that $P = ((v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k))$, where $k = \mathtt{length}(P)$ and $v_0 = u$, and let $c_k = \chi(v_{k-1}, v_k) \in \{\alpha_u, \alpha_v\}$. It is easy to verify that under Assumption 4.1, we have $v_k \neq v$, for otherwise the path $P$ along with the edge $(u, v)$ would create an odd cycle of length $\leq 2L + 3$. The subroutine updates the coloring $\chi$ as follows.

- $\chi(v_{i-1}, v_i) \leftarrow \chi(v_i, v_{i+1})$ for all $i \in [k-1]$.
- $\chi(v_{k-1}, v_k) \leftarrow c$, where $c$ is the unique color in $\{\alpha_u, \alpha_v\} \setminus \{c_k\}$.
- $\chi(u, v) \leftarrow \alpha_v$.

At the end of the above operations, the partial coloring $\chi$ gets extended to the edge $e$. The subroutine returns the updated partial coloring. In summary, the subroutine *applies* the alternating path $P$ to extend the partial coloring to $e$.

**The Subroutine Shift$(\chi, e, P_{\leq i})$:** Here, the input is a partial $(\Delta+1)$-edge coloring $\chi$ of $G$, an uncolored edge $e = (u, v)$ and a length-$i$ prefix $P_{\leq i}$ of an alternating path $P$ starting from $u$ that is of type $\tau = \{\alpha_u, \alpha_v\}$, where $\alpha_u \in \mathtt{miss}_\chi(u) \setminus \mathtt{miss}_\chi(v)$ and $\alpha_v \in \mathtt{miss}_\chi(v) \setminus \mathtt{miss}_\chi(u)$. (If either $\alpha_u \in \mathsf{miss}_\chi(u) \cap \mathsf{miss}_\chi(v)$ or $\alpha_v \in \mathsf{miss}_\chi(u) \cap \mathsf{miss}_\chi(v)$, our algorithm will not make the call $\mathsf{Shift}(\chi, e, P_{\leq i})$.) Crucially, we are also guaranteed that $i \leq 2L + 2$. W.l.o.g., let $P_{\leq i} = ((v_0, v_1), (v_1, v_2), \ldots, (v_{i-1}, v_i))$, where $v_0 = u$. Due to Assumption 4.1, it is again easy to verify that $v_i \neq v$. The subroutine updates $\chi$ as follows.

- $\chi(v_{j-1}, v_j) \leftarrow \chi(v_j, v_{j+1})$ for all $j \in [i-1]$.
- $\chi(v_{i-1}, v_i) \leftarrow \perp$.

- $\chi(u, v) \leftarrow \alpha_v$.

At the end of the above operations, in the partial coloring $\chi$ the position of the uncolored edge gets *shifted* from $e$ to $(v_{i-1}, v_i)$. The subroutine returns the updated coloring.

## 4.2 Our Algorithm

We have a graph $G = (V, E)$, and a partial $(\Delta + 1)$-edge coloring $\chi$ of $G$ with one uncolored edge $e = (u, v)$. We need to extend $\chi$ to $e$. We do this by identifying two colors $c_u \in [\Delta + 1] \setminus \mathsf{miss}_\chi(u)$ and $c_v \in [\Delta + 1] \setminus \mathsf{miss}_\chi(v)$, and then calling Algorithm 1 with input $(G, \chi, e = (u, v), \{c_u, c_v\})$.

While parsing the pseudocode of Algorithm 1, the reader should think of $c_u$ and $c_v$ as **blocking colors**. The algorithm considers an alternating path $P$ starting from $u$, such that the type of this alternating path, given by $\{\alpha_u, \alpha_v\}$, is disjoint from $\{c_u, c_v\}$. Note that it is always possible to find such a type, for the following reason. Since $u$ has degree at most $\Delta$ in $G$ and the edge $e = (u, v)$ is currently uncolored, we have $|\mathtt{miss}_\chi(u)| \geq (\Delta+1) - (\Delta - 1) = 2$. As $c_u \notin \mathtt{miss}_\chi(u)$, it must be the case that $\mathtt{miss}_\chi(u) \setminus \{c_u, c_v\} \neq \emptyset$, and so there exists an appropriate color $\alpha_u$ that we can pick from $\mathtt{miss}_\chi(u) \setminus \{c_u, c_v\}$. A similar argument holds for $\alpha_v$. Also, by induction, it is easy to verify that all future recursive calls to the algorithm will continue to satisfy the same property with regard to the blocking colors. Specifically, if the algorithm is called with input $(G, \chi, (u', v'), \{c_1, c_2\})$, then $\mathtt{miss}_\chi(w) \cap \{c_1, c_2\} \neq \emptyset$ for each $w \in \{u', v'\}$. The reason for having these blocking colors will become apparent later on (see Observation 4.3 and the proof of Lemma 4.4).

**Algorithm 1:** $\mathsf{ExtendColoring}(G, \chi, e = (u, v), \{c_u, c_v\})$

```
1  Find two colors $\alpha_u \in \mathsf{miss}_\chi(u) \setminus \{c_u, c_v\}$ and $\alpha_v \in \mathsf{miss}_\chi(v) \setminus \{c_u, c_v\}$
2  if $\exists c \in \{\alpha_u, \alpha_v\}$ such that $c \in \mathsf{miss}_\chi(u) \cap \mathsf{miss}_\chi(v)$ then
3      $\chi(u, v) \leftarrow c$
4      return $\chi$
5  Let $P := (e_1, \dots, e_k)$ be the $\{\alpha_u, \alpha_v\}$-alternating path in $G$ (w.r.t. $\chi$) starting from $u$
6  if $k \leq 2L + 2$ then
7      $\chi \leftarrow \mathsf{Apply}(\chi, e, P)$
8      return $\chi$
9  else
10     Sample $i \in \{1, \dots, L\}$ independently and u.a.r.
11     Let $P_{\leq i} = (e_1, \dots, e_i)$ denote the prefix of $P$ consisting of its first $i$ edges
12     $\chi \leftarrow \mathsf{Shift}(\chi, e, P_{\leq i})$
13     $\mathsf{ExtendColoring}(G, \chi, e_i, \{\alpha_u, \alpha_v\})$
```

The remainder of Algorithm 1 is very natural and intuitive. If there exists some color $c \in \{\alpha_u, \alpha_v\}$ that is missing *both* at $u$ and $v$, then the algorithm simply assigns the color $c$ to $e$, and terminates. Otherwise, if the length of the alternating path $P$ is at most $2L + 2$, then it extends the partial coloring to $e$ by applying the alternating path, and terminates. Finally, if $\mathtt{length}(P) > 2L + 2$, then it picks some $i \in [L]$ u.a.r., shifts the position of the uncolored edge from $e$ to the $i^{th}$ edge on $P$, and makes a recursive call to itself.

## 4.3 The Meta-Tree

We now define a (possibly infinite) tree $\mathcal{T}$ that captures all possible execution paths taken by our recursive algorithm, on a given input. To clearly distinguish it from the input graph $G$, we refer to $\mathcal{T}$ as a **meta-tree** and its vertices as **meta-nodes**. We next introduce some key notations.

The meta-tree $\mathcal{T}$ is rooted at a meta-node $r$. We let $V(\mathcal{T})$ denote the set of all meta-nodes in $\mathcal{T}$. Every meta-node $x \in V(\mathcal{T})$ corresponds to a recursive **call** of Algorithm 1, and the root-to-leaf path from $r$ to $x$ in $\mathcal{T}$ corresponds to the execution of our recursive algorithm leading to this call.[9] For each such meta-node $x$, we denote the state of an entity $\Phi$ at the start of the corresponding call of Algorithm 1 by $\Phi^{(x)}$. For example, we use the symbols $\chi^{(x)}$, $P^{(x)}$ and $\tau^{(x)}$ respectively to denote the following entities at the start of the concerned call of Algorithm 1: (i) the current partial coloring $\chi$, (ii) the alternating path $P$ (see Line 5 of Algorithm 1) and (iii) the type $\{\alpha_u, \alpha_v\}$ of the path $P$ (see Line 5 of Algorithm 1).

If a meta-node $x$ is *not* a leaf in $\mathcal{T}$, then it has exactly $L$ children, one for each choice of $i \in [L]$ in Line 10 of Algorithm 1. In contrast, a meta-node $x$ is a leaf in $\mathcal{T}$ iff the corresponding call of Algorithm 1 terminates at either Line 4 or Line 8. We will refer to the leaves in $\mathcal{T}$ as **terminals**.

**Random Walks in $\mathcal{T}$:** An execution of our algorithm defines a **random walk** on $\mathcal{T}$ that starts at the root and, at each step, independently and uniformly samples a random child of the current meta-node. The random walk ends when, and if, it reaches a terminal meta-node: At that point the algorithm succeeds in extending the initial partial coloring $\chi$ to the uncolored edge given to it as part of the input. Using standard data structures, it is easy to ensure that the algorithm spends $\tilde{O}(L)$ time at each meta-node it visits during this random walk, because the colors of at most $O(L)$ edges get changed during any given call of Algorithm 1. Since $L = \tilde{O}(\Delta^2 + \sqrt{\Delta n})$, Theorem 2.4 follows from Lemma 4.2 and standard tools for boosting the success probability of a randomized algorithm. We derive some basic properties of the meta-tree in Section 4.3.1. Subsequently, we use these properties to prove Lemma 4.2 in Section 4.4.

**Lemma 4.2.** *With probability $\Omega(1/\log n)$, the random walk on $\mathcal{T}$ executed by our algorithm ends at a terminal vertex that lies within depth $O(\log n)$ from the root $r$.*

### 4.3.1 Basic Properties of the Meta-Tree

Recall that $\tau^{(x)}$ denotes the type of the alternating path $P^{(x)}$ at a meta-node $x$. Consider a non-terminal meta-node $x$ at **depth** $= k$ (say), and suppose that $(x_0, x_1, \ldots, x_k)$ is the unique path in $\mathcal{T}$ from the root to $x$ (i.e., $r = x_0$ and $x = x_k$). We say that the meta-node $x$ is **dirty** iff $\tau^{(x)} \cap \left(\tau^{(x_0)} \cup \cdots \cup \tau^{(x_{k-1})}\right) \neq \emptyset$, and **clean** otherwise. We also say that $x$ is **contaminated** iff at least $L/(10\ell)$ of its children are dirty. Note that a contaminated meta-node itself might be clean.

**Remark:** We emphasize that according to our definitions *only* the non-leaf meta-nodes are classified as being either clean or dirty. Thus, the set of meta-nodes is partitioned into three subsets: terminal, dirty and clean. Furthermore, a subset of non-terminal meta-nodes are contaminated. This implies that some of the contaminated meta-nodes are clean, the rest being dirty.

We next derive a few key properties that will be useful in proving Lemma 4.2 later on.

**Observation 4.3.** *Consider any non-terminal meta-node $x \in V(\mathcal{T})$, and let $y$ be any child of $x$. If $y$ is* not *a terminal, then we must have $\tau^{(x)} \cap \tau^{(y)} = \emptyset$.*

*Proof.* Follows from Line 1 and Line 13 of Algorithm 1. □

**Lemma 4.4.** *Consider any contaminated meta-node $x \in V(\mathcal{T})$ at a depth $\leq \ell$ in the meta-tree $\mathcal{T}$. Then at least $L/(40\ell)$ many children of $x$ are terminal meta-nodes.*

[9] While giving the full proof in Section 8, we formulate our algorithm iteratively instead of recursively and refer to **iterations** instead of **calls**.

*Proof.* Let $(x_0, x_1, \ldots, x_k)$ denote the unique path from the root to $x$ in $\mathcal{T}$, with $r = x_0$ and $x = x_k$. Thus, at most $2(k+1) \le 2(\ell+1) \le 4\ell$ distinct colors appear in the set $\tau^{(x_0)} \cup \tau^{(x_1)} \cup \cdots \cup \tau^{(x_k)} = \mathcal{C}^\star$ (say); because $|\tau^{(x_i)}| = 2$ for all $i \in [k]$. Let $K^\star := \{\tau \in \binom{[\Delta+1]}{2} : \tau \cap \mathcal{C}^\star \neq \emptyset\}$ denote the collection of types with at least one color from $\mathcal{C}^\star$. Note that $|K^\star| \le |\mathcal{C}^\star|(\Delta+1) \le 4\ell(\Delta+1) \le 8\ell\Delta$. Let $D$ denote the set of dirty children of $x$. By definition, for each meta-node $y \in D$, we have $\tau^{(y)} \in K^\star$. Furthermore, since $x$ is contaminated, it follows that $|D| \ge L/(10\ell)$.

Say that a meta-node is **trivial** iff the corresponding call of Algorithm 1 ends at Line 4. Thus, every trivial meta-node is terminal, but not vice versa. Let $D_t \subseteq D$ denote the set of trivial meta-nodes that belong to $D$. If at least half of the meta-nodes in $D$ are trivial, then $|D_t| \ge |D|/2 \ge L/(20\ell)$, and the lemma follows. Thus, for the rest of the proof we assume that:

$$|D \setminus D_t| \ge |D|/2 \ge L/(20\ell). \tag{4}$$

Since $\tau^{(y)} \cap \tau^{(x)} = \emptyset$ for all $y \in D \setminus D_t$ (see Observation 4.3), all the alternating paths in the collection $\mathcal{P}' := \{P^{(y)}\}_{y \in D \setminus D_t}$ exist *simultaneously* in the graph $G$ w.r.t. the partial $(\Delta+1)$-edge coloring $\chi^{(x)}$. To be more precise, this means that for every meta-node $y \in D \setminus D_t$ and every edge $e \in P^{(y)}$, we have $\chi^{(y)}(e) = \chi^{(x)}(e)$. Furthermore, we have already inferred that each path $P^{(y)} \in \mathcal{P}'$ is of a type $\tau^{(y)} \in K^\star$. Next, note that the total length of all possible alternating paths of a given type (w.r.t. a specific partial coloring) is at most $2n$. This holds because such paths are vertex-disjoint, except the same path being possibly counted twice from two opposite directions. Thus, we have $\sum_{P \in \mathcal{P}'} \texttt{length}(P) \le 2n|K^\star| \le 16\ell\Delta n$. Hence, the average length of an alternating path $P^{(y)} \in \mathcal{P}'$ is at most $16\ell\Delta n/|\mathcal{P}'| = 16\ell\Delta n/|D \setminus D_t| \le 320\ell^2\Delta n/L \le L$, where the second-last inequality follows from (4) and the last inequality[10] follows from (3). This implies that at least half of the alternating paths in $\mathcal{P}'$ have length at most $2L$. So, at least half of the meta-nodes $y \in D \setminus D_t$ have $\texttt{length}\left(P^{(y)}\right) \le 2L$; and such meta-nodes are terminals. We therefore conclude that at least $|D \setminus D_t|/2 \ge L/(40\ell)$ children of $x$ are terminal meta-nodes. $\square$

We say that a meta-node $x \in V(\mathcal{T})$ is **congenitally clean** iff every ancestor of $x$, along with $x$ itself, is clean. Next, consider any meta-node $x \in V(\mathcal{T})$ at depth (say) $k$ in $\mathcal{T}$. Let $(x_0, x_1, \ldots, x_k)$ denote the unique path from the root to $x$ in $\mathcal{T}$, with $r = x_0$ and $x = x_k$. Then we refer to the ordered tuple of types $\left(\tau^{(x_0)}, \tau^{(x_1)}, \ldots, \tau^{(x_k)}\right)$ as the **transcript** of $x$. We now upper bound the number of congenitally clean vertices with the same transcript.

**Lemma 4.5.** *Let $\tau_0, \ldots, \tau_i$ be a sequence of types. Then there can be at most $n$ congenitally clean meta-nodes with transcript $= (\tau_0, \ldots, \tau_i)$.*

*Proof.* For $j \in [0, i]$, let $\Gamma_j \subseteq V(\mathcal{T})$ denote the set of congenitally clean meta-nodes with transcript $= (\tau_0, \ldots, \tau_j)$. Note that every meta-node in $\Gamma_j$ is at depth $= j$ in $\mathcal{T}$.

**Claim 4.6.** *For all $j \in [0, i]$, the collection $\left\{P^{(x)}_{\le L}\right\}_{x \in \Gamma_j}$ of length-$L$ prefixes are vertex-disjoint.*

Setting $j = i$ in Claim 4.6, it follows that the size of the set $\Gamma_i$ is at most the maximum possible number of vertex-disjoint paths in the input graph $G$, which in turn, is at most $n$. This implies the lemma. Accordingly, from now on we focus on proving Claim 4.6.

We will prove Claim 4.6 via induction on $j$. Since $\Gamma_0 = \{r\}$, the claim trivially holds if $j = 0$. By induction hypothesis, we now assume that there exists an index $j^\star \in [0, i-1]$ such that the claim holds for all $j \le j^\star$. Under this assumption, we will show that the claim holds for $j = j^\star + 1$.

[10]This is the only place in our analysis where we require $L$ to be $\tilde{\Omega}(\sqrt{\Delta n})$.

If there is a color that appears more than once across the types $\tau_0, \ldots, \tau_{j^\star+1}$, then a meta-node with the transcript $(\tau_0, \ldots, \tau_{j^\star+1})$ is not congenitally clean, and so $\Gamma_j = \emptyset$ for all $j \in [j^\star + 1, i]$. Henceforth, we assume that the types $\tau_0, \ldots, \tau_{j^\star+1}$ are mutually disjoint.

Consider any two distinct meta-nodes $x, y \in \Gamma_{j^\star+1}$. Let $u^{(x)}$ and $u^{(y)}$ respectively denote the starting points of the alternating paths $P^{(x)}$ and $P^{(y)}$. Our induction hypothesis implies that $u^{(x)} \neq u^{(y)}$. Since the types $\tau_0, \ldots, \tau_{j^\star+1}$ are mutually disjoint, both the alternating paths $P^{(x)}$ and $P^{(y)}$ exist *simultaneously* in $G$ w.r.t. the initial partial coloring $\chi^{(r)}$. In other words, either they are two completely disjoint type-$\tau_{j^\star+1}$ alternating paths in $G$ w.r.t. $\chi^{(r)}$, or essentially the same path (with the same set of edges) but with a different orientation. In the first case, their length-$L$ prefixes $P^{(x)}_{\leq L}$ and $P^{(y)}_{\leq L}$ are clearly vertex-disjoint. In the second case, we note that $x$ and $y$ are *not* terminal meta-nodes (because every meta-node in $\Gamma_{j^\star+1}$ is congenitally clean, by definition); hence both $P^{(x)}$ and $P^{(y)}$ have length $\geq 2L + 2$, and so $P^{(x)}_{\leq L}$ and $P^{(y)}_{\leq L}$ are also vertex-disjoint. This concludes the proof of the claim. □

## 4.4 Analyzing the Random Walk on the Meta-Tree: Proof of Lemma 4.2

Our analysis will crucially rely on the behavior of the random walk within a certain **critical subtree** $\mathcal{T}^\star$ of $\mathcal{T}$. We define this critical subtree below, and then summarize a few of its key properties.

The "critical subtree" $\mathcal{T}^\star$ is obtained by starting with $\mathcal{T}$, and then deleting every meta-node $x \in V(\mathcal{T})$ that satisfies at least one of the following conditions: (i) $x$ has a dirty ancestor in $\mathcal{T}$, (ii) $x$ has a contaminated ancestor in $\mathcal{T}$, and (iii) $x$ is at a depth strictly greater than $\ell$ in $\mathcal{T}$. We let $V(\mathcal{T}^\star) \subseteq V(\mathcal{T})$ denote the set of meta-nodes in the critical subtree.

Note that if the root $r$ is itself a terminal, then Lemma 4.2 trivially holds because the random walk ends at $r$. Thus, for the rest of the proof we assume that **the root $r$ is *not* a terminal**.

**Observation 4.7.** *$\mathcal{T}^\star$ is a connected subtree of $\mathcal{T}$, rooted at $r$, and $r$ is* not *a leaf in $\mathcal{T}^\star$. Also, for every $x \in V(\mathcal{T}) \setminus V(\mathcal{T}^\star)$, the unique path in $\mathcal{T}$ from $r$ to $x$ passes through some leaf in $\mathcal{T}^\star$.*

*Proof.* Since the $r$ is at depth 0 and does not have any ancestor, it belongs to $\mathcal{T}^\star$. Furthermore, we have assumed that $r$ is *not* a terminal. This implies that, by definition, $r$ is a congenitally clean meta-node. So all the children of $r$ are part of $\mathcal{T}^\star$, and hence $r$ is *not* a leaf in $\mathcal{T}^\star$.

Next, note that if a meta-node $x$ is in $\mathcal{T}^\star$, then each of its siblings and each of its ancestors is also in $\mathcal{T}^\star$. This implies the observation. □

**Observation 4.8.** *Every non-leaf meta-node in $\mathcal{T}^\star$ is congenitally clean and not contaminated.*

*Proof.* Consider any meta-node $x \in V(\mathcal{T}^\star)$. If $x$ is terminal, then it is a leaf in $\mathcal{T}$ itself, and hence also a leaf in $\mathcal{T}^\star$. In contrast, if $x$ is either contaminated or dirty, then it cannot have any descendant in $\mathcal{T}^\star$. Thus, for $x$ to be a non-leaf meta-node in $\mathcal{T}^\star$, it must be clean and not contaminated. Since we can infer the same for every ancestor of $x$, the observation follows. □

The above observations help us gain an intuitive understanding of the critical subtree $\mathcal{T}^\star$. Define the **core** (resp. **boundary**) of $\mathcal{T}^\star$ to be the set of its non-leaf (resp. leaf) meta-nodes. Every meta-node within the core is congenitally clean and not contaminated (see Observation 4.8). The random walk on $\mathcal{T}$ undertaken by our algorithm starts at the root $r$, which is part of the core of $\mathcal{T}^\star$ (see Observation 4.7). For a certain number of steps the random walk stays within the core. After that, at some point in time the random walk exits the core by reaching a meta-node (say) $x$ at the boundary of $\mathcal{T}^\star$ (see Observation 4.7). If $x$ is terminal, then the random walk ends at $x$. Otherwise, it ventures out of $V(\mathcal{T}^\star)$ in the subsequent step, and never comes back to $V(\mathcal{T}^\star)$ in future.

The above discussion also implies that for every meta-node $x \in V(\mathcal{T}^\star)$, its depth in $\mathcal{T}^\star$ is the same as its depth in $\mathcal{T}$. Accordingly, from this point onward we will use the phrase "the depth of a meta-node" without explicitly referring to the underlying meta-tree.

**Observation 4.9.** *Let $\mathcal{Z}^\star$ denote the set of leaves in $\mathcal{T}^\star$. Then $\mathcal{Z}^\star$ is partitioned into four subsets:*

- $\mathcal{Z}^\star_{\texttt{t}} := \{x \in \mathcal{Z}^\star : x \text{ is terminal}\}$.
- $\mathcal{Z}^\star_{\texttt{cc-cont}} := \{x \in \mathcal{Z}^\star : x \text{ is congenitally clean and contaminated}\}$.
- $\mathcal{Z}^\star_{\texttt{cc-not-cont}} := \{x \in \mathcal{Z}^\star : x \text{ is congenitally clean and } \textit{not} \text{ contaminated}\}$.
- $\mathcal{Z}^\star_{\texttt{d}} := \{x \in \mathcal{Z}^\star : x \text{ is dirty}\}$.

*Proof.* This holds because the set of meta-nodes is partitioned into three substes: terminal, clean and dirty. Finally, every clean meta-node in $\mathcal{T}^\star$ is congenitally clean, because it cannot have any dirty ancestor. ☐

**Corollary 4.10.** *Let $\mathcal{E}^\star_{d}, \mathcal{E}^\star_{t}, \mathcal{E}^\star_{cc\text{-}cont}$ and $\mathcal{E}^\star_{cc\text{-}not\text{-}cont}$ respectively denote the events that the random walk undertaken by our algorithm reaches a meta-node in $\mathcal{Z}^\star_{\texttt{d}}, \mathcal{Z}^\star_{t}, \mathcal{Z}^\star_{cc\text{-}cont}$ and $\mathcal{Z}^\star_{cc\text{-}not\text{-}cont}$. These four events are mutually exclusive and exhaustive, and hence:*

$$\Pr[\mathcal{E}^\star_{d}] + \Pr[\mathcal{E}^\star_{t}] + \Pr[\mathcal{E}^\star_{cc\text{-}cont}] + \Pr[\mathcal{E}^\star_{cc\text{-}not\text{-}cont}] = 1.$$

*Proof.* Follows from Observation 4.7 and Observation 4.9. ☐

**Observation 4.11.** *Every meta-node in $\mathcal{Z}^\star_{cc\text{-}not\text{-}cont}$ is at depth $\ell$.*

*Proof.* Consider any meta-node $x \in \mathcal{Z}^\star_{\texttt{cc-not-cont}}$. By definition, the depth of $x$ is no more than $\ell$. Suppose that $x$ is at a depth (say) $k < \ell$. Since $x$ is clean, it has $L$ children in the meta-tree $\mathcal{T}$.

Let $y$ be any child of $x$ in $\mathcal{T}$. Since $x$ is congenitally clean, $y$ does not have any dirty ancestor. Also, since $x$ is *not contaminated*, $y$ cannot have any contaminated ancestor; for otherwise $x$ itself would have had the same contaminated ancestor and so $x$ would not be part of $\mathcal{T}^\star$. Finally, the meta-node $y$ is at depth $= k+1 \leq \ell$. We therefore conclude that $y$ is part of $\mathcal{T}^\star$. But this contradicts our assumption that $x$ is a leaf in $\mathcal{T}^\star$. So, the meta-node $x$ must be at depth $k = \ell$. ☐

Armed with these basic observations about the critical subtree, we are now ready to prove Lemma 4.2. Our strategy will be to show that the event $\mathcal{E}^\star_{\texttt{cc-cont}} \cup \mathcal{E}^\star_{t}$ occurs with constant probability, and conditioned on this event, the random walk ends within $O(\log n)$ steps with probability $\Omega(1/\log n)$.

**Claim 4.12.** *We have* $\Pr[\mathcal{E}^\star_{cc\text{-}not\text{-}cont}] \leq 1/10$.

*Proof.* By Observation 4.11, every meta-node $x \in \mathcal{Z}^\star_{\texttt{cc-not-cont}}$ is at depth $\ell$. As there are $\binom{\Delta+1}{2} \leq \Delta^2$ possible types, the meta-nodes at depth $\ell$ have at most $(\Delta^2)^{\ell+1} = \Delta^{2(\ell+1)}$ possible transcripts. For each transcript, there are at most $n$ meta-nodes in $\mathcal{Z}^\star_{\texttt{cc-not-cont}}$ (see Lemma 4.5). Thus, we get:

$$|\mathcal{Z}^\star_{\texttt{cc-not-cont}}| \leq \Delta^{2(\ell+1)} n.$$

Next, consider any meta-node $x \in \mathcal{Z}^\star_{\texttt{cc-not-cont}}$, and let $(x_0, x_1, \dots, x_\ell)$ be the unique path from the root $r$ to $x$ in $\mathcal{T}$ (and in $\mathcal{T}^\star_\ell$), with $r = x_0$ and $x = x_\ell$. By our construction of the meta-tree $\mathcal{T}$, every internal meta-node on this path has exactly $L$ children in $\mathcal{T}$. Thus, the random walk taken by our algorithm traverses this path (and ends up at $x$) with probability $1/L^{\ell}$. Summing these probabilities over all $x \in \mathcal{Z}^{\star}_{\texttt{cc-not-cont}}$, we get:

$$\Pr\left[\mathcal{E}^{\star}_{\texttt{cc-not-cont}}\right] \leq \left|\mathcal{Z}^{\star}_{\texttt{cc-not-cont}}\right| \cdot (1/L^{\ell}) \leq \Delta^{2(\ell+1)} n / L^{\ell} < 1/10,$$

where the last inequality follows from (3). □

**Claim 4.13.** *We have* $\Pr\left[\mathcal{E}^{\star}_{\texttt{d}}\right] \leq 1/5$.

*Proof.* Consider any $i \in [0, \ell-1]$. Let $\mathcal{E}_i$ denote the event that after $i$ recursive calls of Algorithm 1, the concerned random walk is at a meta-node (say) $x_i$ that belongs to the core of $\mathcal{T}^{\star}$, and as a corollary, the event $\mathcal{E}^{\star}_{\texttt{d}}$ has not yet taken place. By Observation 4.7, we have:

$$\Pr[\mathcal{E}_0] = 1. \tag{5}$$

We will next prove the following inequality.

$$\Pr\left[\,\overline{\mathcal{E}^{\star}_{d}} \cup \mathcal{E}_{i+1} \,\middle|\, \mathcal{E}_i\,\right] \geq 1 - \frac{1}{10\ell} \text{ for all } i \in [0, \ell-1]. \tag{6}$$

Towards this end, fix any $i \in [0, \ell-1]$, and condition on the event $\mathcal{E}_i$. Since $x_i$ is part of the core of $\mathcal{T}^{\star}$, it is congenitally clean and not contaminated (see Observation 4.8). Thus, the meta-node $x_i$ has $L$ children in $\mathcal{T}$ and at most $L/(10\ell)$ of these children are dirty. Accordingly, with probability at least $1 - 1/(10\ell)$, in the very next step the random walk moves on to a non-dirty child (say) $y$ of $x$. Such a non-dirty child $y$ is either: (1) terminal, or (2) congenitally clean and contaminated, or (3) congenitally clean and not contaminated. In the former two cases, we are guaranteed that the event $\mathcal{E}^{\star}_{\texttt{d}}$ cannot occur, because the event $\mathcal{E}^{\star}_{\texttt{t}} \cup \mathcal{E}^{\star}_{\texttt{cc-cont}}$ has already taken place and this is mutually exclusive with the event $\mathcal{E}^{\star}_{\texttt{d}}$ (see Corollary 4.10). We next consider the third case. Here, if $i < \ell - 1$, then meta-node $y$ is still within the core of $\mathcal{T}^{\star}$, and hence the event $\mathcal{E}_{i+1}$ has occurred. Otherwise, if $i = \ell - 1$, then the event $\mathcal{E}^{\star}_{\texttt{cc-not-cont}}$ has occurred which again is mutually exclusive with the event $\mathcal{E}^{\star}_{d}$ (see Corollary 4.10). This concludes the proof of inequality (6).

From (5) and (6), we infer that $\Pr\left[\,\overline{\mathcal{E}^{\star}_{d}}\,\right] \geq (1 - 1/(10\ell))^{\ell} \geq 4/5$. Thus, we get $\Pr\left[\mathcal{E}^{\star}_{d}\right] = 1 - \Pr\left[\,\overline{\mathcal{E}^{\star}_{d}}\,\right] \leq 1/5$. □

**Corollary 4.14.** *We have* $\Pr\left[\mathcal{E}^{\star}_{t}\right] + \Pr\left[\mathcal{E}^{\star}_{cc\text{-}cont}\right] \geq 7/10$.

*Proof.* Follows from Corollary 4.10, Claim 4.12 and Claim 4.13. □

Let $\mathcal{E}^{\star}$ denote the event that the random walk taken by our algorithm ends at a terminal meta-node at a depth $\leq \ell + 1$. It is trivial to note that $\Pr\left[\mathcal{E}^{\star} | \mathcal{E}^{\star}_{\texttt{t}}\right] = 1$. Henceforth, we condition on the event $\mathcal{E}^{\star}_{\texttt{cc-cont}}$. This means that the concerned random walk has reached some contaminated and congenitally clean meta-node $x$ (say) at depth $\leq \ell$. The meta-node $x$ has $L$ children in $\mathcal{T}$, and by Lemma 4.4 at least $1/(40\ell)$-fraction of its children are terminal. Thus, with probability at least $1/(40\ell)$, in the very next step the random walk moves on to a terminal child of $x$, at depth $\leq \ell + 1$. To summarize, we deduce that $\Pr\left[\mathcal{E}^{\star} | \mathcal{E}^{\star}_{\texttt{t}} \cup \mathcal{E}^{\star}_{\texttt{cc-cont}}\right] \geq 1/(40\ell)$. Lemma 4.2 now follows from (3) and Corollary 4.14.

## 4.5 Getting Rid of Assumption 4.1: The Major Technical Hurdles

If we remove Assumption 4.1, our algorithm might now fail if, given an uncolored edge $e = (u, v)$, it finds a sufficiently short alternating path $P$ starting at $u$ and ending at $v$. In this case, our algorithm runs $\mathsf{Apply}(\chi, e, P)$, but since $P$ has both $u$ *and* $v$ as endpoints, this does not produce a proper coloring. In order to deal with this case, we need to use *Vizing fans* in order to construct this alternating path $P$. In addition to making the algorithm more technical, this leads to the following two major hurdles that we need to overcome.

**Hurdle 1:** Let $x$ and $y$ be two distinct congenitally clean meta-nodes with the same transcript $(\tau_0, \dots, \tau_i)$. Previously, by Claim 4.6, we had that the paths $P^{(x)}_{\leq L}$ and $P^{(y)}_{\leq L}$ were vertex-disjoint. However, *this is no longer necessarily the case.* Every time we construct a fan around some vertex $u$, we make changes to the colors of the edges around $u$, even if they have colors that are not contained in any of the types of the alternating paths that have been used so far. Thus, the alternating path $P^{(x)}$ of some congenitally clean meta-node $x$ might not be a maximal alternating path in the original coloring $\chi^{(r)}$. We say that such a meta-node $x$ is *damaged.* Since each fan only changes the colors of at most $\Delta$ edges, each of which is contained in at most $O(\Delta)$ alternating paths, and our algorithm only runs for $\tilde{O}(1)$ steps, we can argue that each meta-node has at most $\tilde{O}(\Delta^2)$ many damaged children. By taking $L$ to be sufficiently large, we can ensure that (with probability $\Omega(1)$) we do not encounter any damaged meta-nodes in a random walk (see Lemma 8.7).

**Hurdle 2:** Let $e = (u, v)$ be an uncolored edge and $c_u \in \mathsf{miss}_\chi(u)$, $c_v \in \mathsf{miss}_\chi(v)$ be blocking colors. Previously, our algorithm always found an alternating path $P$ starting at $u$ that did not use either $c_u$ or $c_v$. The ability to find an alternating path that can avoid such blocking colors is crucial for the proof of Lemma 4.4. However, if we use Vizing fans to find alternating paths, then we can no longer guarantee that we can avoid using these blocking colors. In the case that we cannot avoid these blocking colors, we use a modified Vizing fan construction to find a $\{c_u, c_v\}$-alternating path that does not start at either $u$ or $v$ (see Lemma 8.3). Similarly to Lemma 4.4, we consider a different case where an $\Omega(1/\ell)$-fraction of the children of a meta-node have this property. In this situation, we can guarantee that $\Omega(L/\ell)$ of the alternating paths corresponding to these children must be vertex-disjoint, and thus have an average length of $\tilde{O}(\sqrt{\Delta n})$, leading to $\Omega(L/\ell)$ many of these children being terminal.

In Section 8, we give the complete proof of Theorem 2.4 without Assumption 4.1.

## Part II
# Full Version

In this part of the paper, we provide the full proof of Theorem 1.1, which we restate at the start of Section 7. In Section 5, we define the basic notations used throughout the rest of the paper. In Section 6, we describe the main algorithmic components that we use to construct our algorithm. In Section 7, we describe and analyse our algorithm. Finally, in Sections 8 and 9, we prove Theorem 6.4 and Lemma 6.1 respectively.

We note that this part of the paper is fully self-contained and uses slightly different terminology and notation than the extended abstract.

## 5 Basic Notations

Let $G = (V, E)$ be an undirected simple graph on $n$ vertices and $m$ edges with maximum vertex degree $\Delta$. For any partial $(\Delta+1)$-edge coloring $\chi$ of $G = (V, E)$, we say that a neighbor $v$ of vertex $u$ is a *colored neighbor* (respectively, *uncolored neighbor*) of $u$ if $\chi(u, v) \neq \perp$ (resp., $\chi(u, v) = \perp$). For any vertex $u \in V$, let $\mathsf{miss}_\chi(u) \subseteq [\Delta+1]$ be the set of colors which are not used around $u$ by $\chi$. Let $N_\chi(u)$ denote the set of all colored neighbors of $u$ in $G$. We use $\deg_G^\chi(u)$ to denote the number of *uncolored* edges incident on $u$ w.r.t. $\chi$. For any pair of different colors $\{x, y\}$, a simple path $P$ between two endpoints $s, t$ is an $\{x, y\}$-alternating path under $\chi$, if all edges on $P$ are colored either $x$ or $y$; plus, $P$ is maximal if $\{x, y\} \cap \mathsf{miss}_\chi(s) \neq \emptyset$ and $\{x, y\} \cap \mathsf{miss}_\chi(t) \neq \emptyset$.

## 6 The Algorithmic Components

In this section, we describe the subroutines that we use to construct our algorithm.

### 6.1 Coloring Stars

We prove the following lemma following the basic ideas from the recent work of [BCC$^+$24]. The proof of this lemma hinges on the same key technical observations but with a simpler algorithm and better runtime bound, whose full proof is deferred to Section 9.

**Lemma 6.1.** *There is an algorithm* `ColorLightStars` *that, given a graph $G$, a partial $(\Delta+1)$-edge coloring $\chi$ of $G$ and subset $U^\star \subseteq V$ such that:*

- *$|\mathsf{miss}_\chi(u)| \geq d$ for all $u \in U^\star$ for some positive integer $d \in \mathbb{N}$,*
- *there are $\lambda$ uncolored edges incident on $U^\star$,*

*extends the coloring $\chi$ to $\Omega(\lambda)$ uncolored edges incident on $U^\star$ in $\tilde{O}(\lambda\Delta + \Delta m/d)$ expected time.*

**Remark.** As mentioned in the technical overview of Section 2, previously in [BCC$^+$24], the runtime bound for the same task is $\tilde{O}\left(\lambda\Delta + \min_{\tau\geq 1}\{\frac{\Delta m\tau}{d} + \frac{\lambda n}{\tau}\}\right)$, which is always worse than the time bound of Lemma 6.1. To clarify, note that in the technical overview of [BCC$^+$24], the authors claimed the same runtime bound $\tilde{O}(\lambda\Delta + \Delta m/d)$ in their Lemma 2.1 as we claim in the current paper (see Corollary 6.2, which we derive from Lemma 6.1), but their Lemma 2.1 was only given as part of an informal technical overview, provided merely for the sake of conveying the high-level ideas. What [BCC$^+$24] actually achieved implicitly (in the formal part of their paper) is the weaker runtime bound of $\tilde{O}\left(\lambda\Delta + \min_{\tau \geq 1}\{\frac{\Delta m \tau}{d} + \frac{\lambda n}{\tau}\}\right)$, which was sufficient for their $\tilde{O}(mn^{1/3})$ time algorithm, but it is insufficient for our goal of achieving a runtime of $\tilde{O}(mn^{1/4})$.

In our final algorithm, we use `ColorLightStars` to extend the coloring $\chi$ to vertices with sufficiently low degree. To deal with high degree vertices, we also need a variant of the algorithm that we call `ColorHeavyStars`, which follows as a corollary from Lemma 6.1.

**Corollary 6.2.** *There is an algorithm* `ColorHeavyStars` *that, given a graph $G$, a partial $\Delta+1$ edge coloring $\chi$ of $G$ and subset $U^\star \subseteq V$ such that there are $\lambda$ uncolored edges incident on $U^\star$, extends the coloring $\chi$ to $\Omega(\lambda)$ uncolored edges incident on $U^\star$ in $\tilde{O}(\lambda\Delta + \Delta m|U^\star|/\lambda)$ expected time.*

*Proof.* For each integer $p$, let $U_p^\star := \{u \in V \mid \deg_G^\chi(u) \in [2^p, 2^{p+1})\}$.[11] Let $p \in \mathbb{N}$ be the value that maximizes $\sum_{u \in U_p^\star} \deg_G^\chi(u)^2$. Let $\lambda'$ denote the number of uncolored edges incident on $U_p^\star$ and $d = 2^p$. Then it follows that

$$\begin{aligned}
\lambda' \cdot 2d &= \sum_{u \in U_p^\star} \deg_G^\chi(u) \cdot 2d \geq \sum_{u \in U_p^\star} \deg_G^\chi(u)^2 \geq \frac{1}{\log n} \cdot \sum_{u \in U^\star} \deg_G^\chi(u)^2 \\
&\geq \frac{1}{\log n} \cdot \left( \frac{1}{\sqrt{|U^\star|}} \cdot \sum_{u \in U^\star} \deg_G^\chi(u) \right)^2 = \frac{1}{\log n} \cdot \frac{\lambda^2}{|U^\star|}, \qquad (7)
\end{aligned}$$

where the last inequality follows from the Cauchy-Schwarz inequality. It follows that $\lambda^2/(\lambda'|U^\star|) \leq d \cdot 2\log n$. The set $U_p^\star$ has the properties that

1. $|\mathsf{miss}_\chi(u)| \geq d$ for all $u \in U_p^\star$,
2. there are $\lambda'$ uncolored edges incident on $U_p^\star$.

Thus, applying the algorithm `ColorLightStars` (Lemma 6.1) with the set $U_p^\star$, we can color $\Omega(\lambda')$ uncolored edges incident on $U_p^\star$ in expected time

$$\tilde{O}\left(\lambda'\Delta + \frac{\Delta m}{d}\right) \leq \tilde{O}\left(\lambda'\Delta + \frac{\Delta m|U^\star|}{\lambda} \cdot \frac{\lambda'}{\lambda}\right) \leq \tilde{O}\left(\lambda\Delta + \frac{\Delta m|U^\star|}{\lambda}\right) \cdot \frac{\lambda'}{\lambda}. \qquad (8)$$

We can repeat this process until the total number of edges to which we have extended the coloring exceeds $\lambda/2$. Let $\lambda_i$ denote the number of uncolored edges at the start of the $i$ iteration and let $\lambda_i'$ denote the number of edges that we extend the coloring to in the $i^{th}$ iteration. Suppose that we perform $t$ iterations in total. Then, for all $i \in [t]$, we have that $\lambda/2 \leq \lambda_i \leq \lambda$. It follows from Equation (8) that the expected total running time of our algorithm is at most

$$\sum_{i=1}^{t} \tilde{O}\left(\lambda_i\Delta + \frac{\Delta m|U^\star|}{\lambda_i}\right) \cdot \frac{\lambda_i'}{\lambda_i} \leq \tilde{O}\left(\lambda\Delta + \frac{\Delta m|U^\star|}{\lambda}\right) \cdot \left(\frac{1}{\lambda} \cdot \sum_{i=1}^{t} \lambda_i'\right) \leq \tilde{O}\left(\lambda\Delta + \frac{\Delta m|U^\star|}{\lambda}\right).$$

Thus, the expected total time spent handling these calls to `ColorLightStars` is upper bounded by $\tilde{O}(\lambda\Delta + \Delta m|U^\star|/\lambda)$. □

## 6.2 Near-Vizing Coloring

The following theorem is the main technical result in the recent work of [Ass24]. While this is not strictly necessary to prove Theorem 6.3, it significantly simplifies the technical details of our algorithm by allowing us to avoid using Euler partitioning and recursion.

**Theorem 6.3.** *There is an algorithm* `NearVizingColoring` *that, given a graph $G$, computes a $\Delta + 300\log n$ edge coloring $\chi$ of $G$ in $\tilde{O}(m)$ time with high probability.*

[11] Recall that $\deg_G^\chi(u)$ denotes the number of uncolored edges incident on $u$ w.r.t. $\chi$.

## 6.3 Color Extension Theorem

The following theorem is our main technical contribution. This theorem is a generalization of the results of [DHZ19] to the case where we only have access to $\Delta + 1$ colors instead of $\Delta + \tilde{\Omega}(\sqrt{\Delta})$. We defer the proof of this theorem to Section 8.

**Theorem 6.4.** *There is an algorithm* `FastVizingExtension` *that, given a graph $G$, a partial $\Delta + 1$ edge coloring $\chi$ of $G$ and an uncolored edge $e$, extends the coloring $\chi$ to the edge $e$ in $\tilde{O}(\min\{\Delta^2 + \sqrt{\Delta n}, n\})$ expected time.*

## 7 Our Algorithm

In this section, we prove Theorem 1.1, which we restate below.

**Theorem 7.1** (Theorem 1.1 Restated)**.** *Given a simple, undirected $m$-edge $n$-node graph $G = (V, E)$ with maximum degree $\Delta$, we can compute a $(\Delta + 1)$-edge coloring of $G$ in $\tilde{O}(mn^{1/4})$ time with high probability.*

Let $G = (V, E)$ be a graph with maximum degree $\Delta$. We assume that $\Delta \geq \kappa n^{1/4} \log n$, where $\kappa := 10^4$ is a constant. Otherwise, we can use the $\tilde{O}(m\Delta)$ time algorithm of [Sin19] to $\Delta + 1$ edge color $G$ in $\tilde{O}(mn^{1/4})$ time. Let $V_{\mathsf{lo}} := \{u \in V \mid \deg_G(u) \leq \Delta/2\}$ be the *low degree* nodes, $V_{\mathsf{hi}} := V \setminus V_{\mathsf{lo}}$ be the *high degree* nodes, $n_{\mathsf{lo}} := |V_{\mathsf{lo}}|$ and $n_{\mathsf{hi}} := |V_{\mathsf{hi}}|$. Given the graph $G$, we now describe how our algorithm computes a $(\Delta + 1)$-edge coloring $\chi$ of the graph $G$ in 2 phases.

### Phase 1: Extracting Stars

Our algorithm begins by finding a subset $U \subseteq V$ that satisfies the following:

$$\text{(I) } \Delta(G[V \setminus U]) \leq \Delta - 300 \log n, \text{ (II) } |U_{\mathsf{hi}}| \leq \frac{|V_{\mathsf{hi}}|}{\Delta} \cdot \frac{3\kappa \log n}{2}, \text{ (III) } \lambda_{\mathsf{init}} \leq \frac{m}{\Delta} \cdot 10\kappa \log n, \tag{9}$$

where $U_{\mathsf{hi}} := U \cap V_{\mathsf{hi}}$ and $\lambda_{\mathsf{init}} := \sum_{u \in U} \deg_G(u)$. We use the following algorithm in order to obtain such a set $U$.

**Algorithm 2:** `ExtractStars`$(G)$

```
1 for $10 \log n$ iterations do
2     Sample $U \subseteq V$ by placing each $u \in V$ into $U$ independently with probability $\kappa \log n / \Delta$
3     if $|V_{\mathsf{hi}}| < \Delta/4$ then
4         $U \leftarrow U \cap V_{\mathsf{lo}}$
5     if $U$ satisfies Equation (9) then
6         return $U$
```

The following lemma summarises the behaviour of Algorithm 2.

**Lemma 7.2.** *Algorithm 2 runs in $\tilde{O}(m)$ time and returns a set $U$ satisfying Equation (9) w.h.p.*

*Proof.* Each iteration of the algorithm can be implemented to run in $\tilde{O}(m)$ time in the obvious way. We now argue that each iteration of the algorithm finds a set $U$ which satisfies Equation (9) with probability at least $1/2$ by lower bounding the probability that the set $U$ satisfies each of the conditions.

**Condition (I):** We first consider the case that $|V_{\mathsf{hi}}| \geq \Delta/4$. Given some $u \in V$, we want to show that $\deg_{V \setminus U}(u) \leq \Delta - 300 \log n$ w.h.p. If $\deg_V(u) \leq \Delta/2$, then this holds trivially since $\Delta \geq \kappa \log n$.

Otherwise, if $\deg_V(u) \geq \Delta/2$, we have that $\mathbb{E}[\deg_U(u)] = \deg_V(u) \cdot (\kappa/\Delta) \log n \geq (\kappa/2) \log n$ and it follows from Chernoff bounds that

$$\Pr\left[\deg_U(u) < \frac{\kappa}{4} \cdot \log n\right] \leq \exp\left(-\frac{\kappa}{16} \cdot \log n\right) \leq n^{-\kappa/16}.$$

Thus, $\deg_U(u) \geq 300 \log n$ with probability at least $1 - n^{-\kappa/16}$. Taking a union bound, we get that Condition (I) holds w.h.p. Now, consider the case that $|V_{\mathsf{hi}}| < \Delta/4$. Given some $u \in V$, we want to show that $\deg_{V \setminus U_{\mathsf{lo}}}(u) \leq \Delta - 300 \log n$ w.h.p., where $U_{\mathsf{lo}}(u) = U \cap V_{\mathsf{lo}}(u)$. If $\deg_V(u) \leq \Delta/2$, then this again holds trivially. Otherwise, $u$ has at least $\Delta/4$ neighbors in $V_{\mathsf{lo}}$. It follows that $\mathbb{E}[\deg_{U_{\mathsf{lo}}}(u)] \geq (\kappa/4) \log n$ and, applying Chernoff bounds, we get that

$$\Pr\left[\deg_{U_{\mathsf{lo}}}(u) < \frac{\kappa}{8} \cdot \log n\right] \leq \exp\left(-\frac{\kappa}{32} \cdot \log n\right) \leq n^{-\kappa/32}.$$

Taking a union bound, we again get that Condition (I) holds w.h.p.

**Condition (II):** If $|V_{\mathsf{hi}}| < \Delta/4$, then $U_{\mathsf{hi}} = \emptyset$, and Condition (II) holds trivially. If $|V_{\mathsf{hi}}| \geq \Delta/4$, we have that $\mathbb{E}[|U_{\mathsf{hi}}|] = (|V_{\mathsf{hi}}|/\Delta) \cdot \kappa \log n \geq (\kappa/4) \log n$. Applying Chernoff bounds, we get that

$$\Pr\left[|U_{\mathsf{hi}}| \geq \frac{3}{2} \cdot \frac{|V_{\mathsf{hi}}|}{\Delta} \cdot \kappa \log n\right] \leq \exp\left(-\frac{1}{10} \cdot \frac{|V_{\mathsf{hi}}|}{\Delta} \cdot \kappa \log n\right) \leq \exp\left(-\frac{\kappa}{40} \cdot \log n\right) \leq n^{-\kappa/40}.$$

**Condition (III):** We first note that

$$\mathbb{E}\left[\sum_{u \in U} \deg_G(u)\right] = \sum_{u \in V} \deg_G(u) \cdot \Pr[u \in U] = \frac{2m\kappa \log n}{\Delta}.$$

By Markov's inequality, it follows that

$$\Pr\left[\sum_{u \in U} \deg_G(u) \geq \frac{10m\kappa \log n}{\Delta}\right] \leq \frac{1}{5}.$$

By taking a union bound, we have that $U$ satisfies Equation (9) with probability at least $1/2$. $\square$

**Phase 2: Coloring the Graph**

After finding a set $U$ that satisfies Equation (9), our algorithm proceeds to edge coloring the graph $G$. Using the fact that $U$ satisfies Condition (I), we can apply `NearVizingColoring` (Theorem 6.3) in order to produce a $(\Delta+1)$-edge coloring $\chi$ of $G[V \setminus U]$. We then use `ColorLightStars` (Lemma 6.1) to extend $\chi$ to all of the edges incident on nodes in $U_{\mathsf{lo}} := \{u \in U \mid \deg_G(u) \leq \Delta/2\}$. Finally, we use `ColorHeavyStars` (Corollary 6.2) to extend $\chi$ to all of the edges incident on nodes in $U_{\mathsf{hi}} := U \setminus U_{\mathsf{lo}}$. Algorithm 3 gives the full description of the second phase of our algorithm using the subroutines from Section 6. We write $n_{\mathsf{hi}} = |V_{\mathsf{hi}}|$.

**Algorithm 3:** FastColoring$(G, U)$

```
// Step 1: Efficiently color the non-star edges using slack
$\chi \leftarrow$ NearVizingColoring$(G[V \setminus U])$
// Step 2: Extend $\chi$ to the light stars
$\lambda_{\mathsf{lo}} := \sum_{u \in U_{\mathsf{lo}}} \deg_G^{\chi}(u)$
while $\lambda_{\mathsf{lo}} \geq 1$ do
    $\chi \leftarrow$ ColorLightStars$(G, \chi, U_{\mathsf{lo}})$
// Step 3: Extend $\chi$ to the heavy stars
$\lambda_{\mathsf{hi}} := \sum_{u \in U_{\mathsf{hi}}} \deg_G^{\chi}(u)$
$\tau := \sqrt{m n_{\mathsf{hi}} / \min\{\Delta^2 + \sqrt{\Delta n}, n\}}$
while $\lambda_{\mathsf{hi}} > \tau$ do
    $\chi \leftarrow$ ColorHeavyStars$(G, \chi, U_{\mathsf{hi}})$
while $\lambda_{\mathsf{hi}} \geq 1$ do
    Let $e$ be an uncolored edge
    $\chi \leftarrow$ FastVizingExtension$(G, \chi, e)$
return $\chi$
```

The following theorem summarizes the properties of Algorithm 3.

**Theorem 7.3.** *Algorithm 3 produces a $(\Delta + 1)$-edge coloring of the graph $G$ in expected time $\tilde{O}(mn^{1/4})$.*

Theorem 1.1 follows directly from Theorem 7.3 and Lemma 7.2 using standard probabilistic amplification.

## 7.1 Analysis

It follows from the properties of the set $U$ and the correctness of the subroutines used by our algorithm that Algorithm 3 always returns a $(\Delta + 1)$-edge coloring of $G$. It remains to bound the expected running time of our algorithm. It follows directly from Theorem 6.3 that Step 1 of our algorithm takes $\tilde{O}(m)$ expected time. We now show that Steps 2 and 3 of our algorithm take $\tilde{O}(m)$ and $\tilde{O}(mn^{1/4})$ expected time, respectively.

### The Running Time of Step 2

We first note that, for all $u \in U_{\mathsf{lo}}$, we have $|\mathsf{miss}_G^{\chi}(u)| \geq \Delta/2$. Applying Lemma 6.1, it follows that Step 2 of our algorithm runs for $\tilde{O}(1)$ iterations, where each iteration takes expected time $\tilde{O}(\lambda_{\mathsf{lo}}\Delta + m)$. By the properties of the vertex set $U$ (refer to Equation (9)) obtained by Algorithm 2, we have $\lambda_{\mathsf{lo}} \leq \lambda_{\mathsf{init}} \leq \tilde{O}(m/\Delta)$. Thus, Step 2 takes $\tilde{O}(m)$ expected time in total.

### The Running Time of Step 3

Applying Corollary 6.2 and Theorem 6.4, it follows that Step 3 of our algorithm takes expected time

$$\tilde{O}\left(\lambda_{\mathsf{hi}}\Delta + \frac{\Delta m |U_{\mathsf{hi}}|}{\tau}\right) + \tilde{O}\left(\tau \cdot \min\{\Delta^2 + \sqrt{\Delta n}, n\}\right).$$

By the properties of the vertex set $U$ (refer to Equation (9)), we have $\lambda_{\mathsf{hi}} \leq \lambda_{\mathsf{init}} \leq \tilde{O}(m/\Delta)$ and $|U_{\mathsf{hi}}| \leq \tilde{O}(n_{\mathsf{hi}}/\Delta)$, hence the expected running time of Step 3 of our algorithm is bounded by

$$\begin{aligned}\tilde{O}\Big(m + \frac{mn_{\mathsf{hi}}}{\tau}\Big) + \tilde{O}\Big(\tau \cdot \min\{\Delta^2 + \sqrt{\Delta n}, n\}\Big) &= \tilde{O}(m) + \tilde{O}\left(\sqrt{mn_{\mathsf{hi}} \cdot \min\{\Delta^2 + \sqrt{\Delta n}, n\}}\right) \\ &= \tilde{O}(m) + \tilde{O}\Big(\sqrt{mn_{\mathsf{hi}}} \cdot \min\{\Delta + (\Delta n)^{1/4}, \sqrt{n}\}\Big).\end{aligned}$$

Let $\Gamma := \sqrt{mn_{\mathsf{hi}}} \cdot \min\{\Delta + (\Delta n)^{1/4}, \sqrt{n}\}$. We now show that $\Gamma \leq 4mn^{1/4}$, which implies that Step 3 has an expected running time of $\tilde{O}(mn^{1/4})$. We consider the following 3 cases.

*Case 1:* $\Delta \geq \sqrt{n}$. Then $\min\{\Delta + (\Delta n)^{1/4}, \sqrt{n}\} \leq \sqrt{n}$, thus

$$\Gamma \leq \sqrt{mn_{\mathsf{hi}}} \cdot \sqrt{n} = m\sqrt{n_{\mathsf{hi}} n/m} < m\sqrt{4n/\Delta} = 2m\sqrt{n/\Delta} \leq 2mn^{1/4},$$

where the penultimate and last inequalities hold as $\Delta n_{\mathsf{hi}} \leq 4m$ and $\Delta \geq \sqrt{n}$, respectively.

*Case 2:* $n^{1/3} \leq \Delta < \sqrt{n}$. Then $\min\{\Delta + (\Delta n)^{1/4}, \sqrt{n}\} \leq 2\Delta$, thus

$$\Gamma \leq 2\sqrt{mn_{\mathsf{hi}}} \cdot \Delta = 2\sqrt{m\Delta n_{\mathsf{hi}}} \cdot \sqrt{\Delta} \leq 4m\sqrt{\Delta} \leq 4mn^{1/4},$$

where the penultimate and last inequalities hold as $\Delta n_{\mathsf{hi}} \leq 4m$ and $\Delta \leq \sqrt{n}$.

*Case 3:* $\Delta < n^{1/3}$. Then $\min\{\Delta + (\Delta n)^{1/4}, \sqrt{n}\} \leq 2(\Delta n)^{1/4}$, thus

$$\Gamma \leq 2\sqrt{mn_{\mathsf{hi}}} \cdot (\Delta n)^{1/4} = 2m\sqrt{n_{\mathsf{hi}}/m} \cdot (\Delta n)^{1/4} \leq 2m\sqrt{4/\Delta} \cdot (\Delta n)^{1/4} = 4m(n/\Delta)^{1/4} \leq 4mn^{1/4},$$

where the penultimate inequality holds as $\Delta n_{\mathsf{hi}} \leq 4m$.

## 8 Color Extension to Edges

In this section, we prove the following theorem.

**Theorem 8.1** (Theorem 6.4 Restated)**.** *Given a undirected simple graph $G = (V, E)$ on $n$ vertices and maximum degree $\Delta$, as well as and a partial $(\Delta + 1)$-edge coloring $\chi$ of $G$ with an uncolored edge $e$, we can extend $\chi$ to the edge $e$ in $\tilde{O}(\Delta^2 + \sqrt{\Delta n})$ expected time.*

This section is self-contained (excluding the basic notations from Section 5) and uses slightly different notation than the sketch of this proof in Section 4.

### 8.1 Preliminaries

We now describe the main components used in our multi-step Vizing chain construction.

#### The Algorithm VizingFan

The first component of our algorithm is a standard Vizing fan construction [Sin19, DHZ19] but with one simple modification. The algorithm VizingFan takes as input some uncolored edge $(u, v)$ and colors $c_u \in \mathsf{miss}_\chi(u)$ and $c_v \in \mathsf{miss}_\chi(v)$. It then proceeds to either extend $\chi$ to the uncolored edge $(u, v)$ by shifting colors around $u$, or constructs a Vizing fan around the vertex $u$ while ensuring that (1) none of the vertices $v_i$ participating in the fan have $c_u \in \mathsf{miss}_\chi(v_i)$, and (2) that the color of the first vertex in the fan is not $c_v$. Algorithm 4 gives a formal description of this procedure.

**Algorithm 4:** VizingFan$(\chi, u, v, c_u, c_v)$

```
1  $i \leftarrow 0$ and $v_0 \leftarrow v$
2  Let $c_0 \in \mathsf{miss}_\chi(v_0) \setminus \{c_v\}$
3  while $c_i \notin \{c_0, \dots, c_{i-1}\}$ and $c_i \notin \mathsf{miss}_\chi(u)$ do
4      Let $(u, v_{i+1})$ be the edge with color $\chi(u, v_{i+1}) = c_i$
5      Let $c_{i+1} \in \mathsf{miss}_\chi(v_{i+1})$
6      if $c_u \in \mathsf{miss}_\chi(v_{i+1})$ then
7          $c_{i+1} \leftarrow c_u$
8      $i \leftarrow i + 1$
9  if $c_i \in \mathsf{miss}_\chi(u)$ then
10     for $j = 0 \dots i$ do
11         $\chi(u, v_j) \leftarrow c_j$
12     return $\chi$
13 else
14     return $(v_0, c_0), \dots, (v_i, c_i)$
```

We refer to the sequence $(v_0, c_0), \dots, (v_i, c_i)$ returned by Algorithm 4 as a Vizing fan. We say that a Vizing fan is $c$-primed if $c = c_i$.

## The Algorithm Vizing

We now give an algorithm Vizing which builds a Vizing chain using VizingFan as a subroutine and extends the coloring $\chi$ to the uncolored edge $(u, v)$. The algorithm also takes colors $c_u \in \mathsf{miss}_\chi(u)$ and $c_v \in \mathsf{miss}_\chi(v)$ as input. We make some small modifications to the standard implementation of this algorithm for technical reasons that will become clear later on. Algorithm 5 gives a formal description of this procedure.

**Algorithm 5:** Vizing$(\chi, u, v, c_u, c_v)$

```
1  if VizingFan$(\chi, u, v, c_u, c_v)$ extends $\chi$ then
2      $\chi \leftarrow$ VizingFan$(\chi, u, v, c_u, c_v)$
3      return $\chi$
4  $F = (v_0, c_0), \dots, (v_k, c_k) \leftarrow$ VizingFan$(\chi, u, v, c_u, c_v)$
5  if $F$ is not $c_v$-primed then
6      Let $c \in \mathsf{miss}_\chi(u) \setminus \{c_u\}$
7      Let $P$ denote the maximal $\{c, c_k\}$-alternating path starting at $u$
8  if $F$ is $c_v$-primed then
9      Let $c_i$ be the first appearance of $c_v$ in $c_0, \dots, c_k$
10     Let $P$ denote the maximal $\{c_u, c_v\}$-alternating path starting at $v_i$
11 Extend $\chi$ to $(u, v)$ by flipping the path $P$ and shifting colors around $u$ (see Lemma 8.2)
12 return $\chi$
```

Thus, running Vizing$(\chi, u, v, c_u, c_v)$ extends the coloring $\chi$ to the uncolored edge $(u, v)$ by shifting colors around the vertex $u$ and flipping the colors of the alternating path $P$. The analysis of the case where $F$ is not $c_v$-primed is the same as in the standard Vizing chain construction [Sin19]. The case where $F$ is $c_v$-primed follows from a similar argument. The following lemma summarises the behaviour of this algorithm.

**Lemma 8.2.** *Algorithm 5 extends the coloring $\chi$ to the edge $(u, v)$ in time $\tilde{O}(\Delta + |P|)$.*

*Proof.* We begin by showing that the path $P$ is well-defined. In the case that $F$ is not $c_v$-primed, we know that $c \in \mathsf{miss}_\chi(u)$ and $c_k \notin \mathsf{miss}_\chi(u)$, so there is a $\{c, c_k\}$-alternating path starting at $u$. In the case that $F$ is $c_v$-primed, we know by the properties of our Vizing fan construction that $c_v \in \mathsf{miss}_\chi(v_i)$ and $c_u \notin \mathsf{miss}_\chi(v_i)$, so there is a $\{c_u, c_v\}$-alternating path starting at $u$. Thus, in both cases, the path $P$ is well-defined. We now describe how to extend the coloring $\chi$ to $(u, v)$ in both cases.

If $F$ is not $c_v$-primed: Let $c_j$ be the first appearance of $c_k$. We consider the cases where the $P$ does or does not have $v_j$ as an endpoint. If $P$ does not end at $v_j$, then we can shift the colors of the first $j+1$ edges in the fan by setting $\chi(u, v_0) \leftarrow c_0, \dots, \chi(u, v_j) \leftarrow c_j$ and flip the colors of the alternating path $P$. If $P$ does end at $v_j$, then we flip the colors of the alternating path $P$, shift the colors of the fan by setting $\chi(u, v_0) \leftarrow \chi(u, v_1), \dots, \chi(u, v_{k-1}) \leftarrow \chi(u, v_k)$, and set $\chi(u, v_k) \leftarrow c_k$. Note that, while shifting the fan, we have $\chi(u, v_{i+1}) = c \neq c_k$.

If $F$ is $c_v$-primed: We consider the cases where the $P$ does or does not have $u$ as an endpoint. If $P$ does not end at $u$, then the edge $(u, v_{i+1})$ (which has color $c_v$) is not contained in $P$. Thus, we can shift the colors of the first $i$ edges in the fan by setting $\chi(u, v_0) \leftarrow c_0, \dots, \chi(u, v_{i-1}) \leftarrow c_{i-1}$, flip the colors of the alternating path $P$, and set $\chi(u, v_i) \leftarrow c_u$. If $P$ does end at $u$, then the edge $(u, v_{i+1})$ is contained in $P$. Thus, we can flip the colors of the alternating path $P$, shift the colors of the fan by setting $\chi(u, v_0) \leftarrow \chi(u, v_1), \dots, \chi(u, v_{k-1}) \leftarrow \chi(u, v_k)$, and set $\chi(u, v_k) \leftarrow c_v$. Note that, while shifting the fan, we have $\chi(u, v_{i+1}) = c_u \neq c_i$.

Using standard data structures, we can implement this algorithm to run in time $\tilde{O}(\Delta + |P|)$. $\square$

The following lemma summarises the key properties of the path $P$ considered by Algorithm 5.

**Lemma 8.3.** *The path $P$ considered by the algorithm satisfies one of the following properties:*

1. *$P$ is a maximal $\{c', c''\}$-alternating path in $\chi$ starting at $u$ where $\{c', c''\} \cap \{c_u, c_v\} = \emptyset$.*
2. *$P$ is a maximal $\{c_u, c_v\}$-alternating path in $\chi$ starting at neither $u$ nor $v$.*[12]

These properties of the path $P$ will be crucial in our analysis later on. We refer to an alternating path satisfying Condition 1 (resp. Condition 2) above as *non-overlapping* (resp. *overlapping*).

## The Algorithm TruncatedVizing

We now give an algorithm TruncatedVizing, which takes the same input as Vizing along with an additional argument $t \in \mathbb{N}$. It then proceeds to extend the coloring $\chi$ to the edge $e$ in the same way as Vizing, with one difference: If the maximal alternating path $P$ that is flipped by Vizing has length greater than $t$, the algorithm only flips the first $t-1$ edges of $P$ and leaves the $t^{th}$ edge in the path $P$ uncolored. It then returns the new coloring obtained after applying this procedure along with the edge left uncolored (as long as $P$ had length greater than $t$). Algorithm 6 gives a formal description of this procedure.

[12]Note that, in this case, the path $P$ starts at the vertex $v_i$ of the fan such that $c_i$ is the first appearance of $c_v$ in $c_0, \dots, c_k$. Since $c_0 \neq c_v$, we know that $v_i \neq v$.

**Algorithm 6:** $\textsf{TruncatedVizing}(\chi, u, v, c_u, c_v, t)$

```
1  Let $P = e_1, \ldots, e_t$ denote the alternating path considered by $\textsf{Vizing}(\chi, u, v, c_u, c_v)$
2  if $|P| \leq t$ then
3      $\chi \leftarrow \textsf{Vizing}(\chi, u, v, c_u, c_v)$
4      return $\chi$
5  if $P$ is non-overlapping then
6      Let $(u, v_i)$ be the first edge in $P$
7      for $j = 0 \ldots i-1$ do
8          $\chi(u, v_j) \leftarrow c_j$
9  if $P$ is overlapping then
10     Let $v_i$ be the first vertex in $P$
11     for $j = 0 \ldots i-1$ do
12         $\chi(u, v_j) \leftarrow c_j$
13     $\chi(u, v_i) \leftarrow c_u$
14 Flip the colors of the first $t-1$ edges in the $\{c', c''\}$-alternating path $P$
15 $\chi(e_t) \leftarrow \perp$
16 $(u', v') \leftarrow e_t$
17 Let $c'_u \in \textsf{miss}_\chi(u') \cap \{c', c''\}$ and $c'_v \in \textsf{miss}_\chi(v') \cap \{c', c''\}$
18 return $\chi, u', v', c'_u, c'_v$
```

## 8.2 Our Algorithm

We define parameters $\ell := 10^2 \log n$ and $L := 10^3 \ell^2(\Delta^2 + \sqrt{\Delta n})$. Our algorithm is given an uncolored edge $(u, v)$ and starts by attempting to construct a Vizing chain by calling $\textsf{Vizing}$. If the Vizing chain has length $\Omega(L)$, then our algorithm instead calls $\textsf{TruncatedVizing}$ and randomly truncates the Vizing chain after $O(L)$ steps. It then repeats this process, moving the uncolored edge around the graph by randomly truncating long Vizing chains, until it finds a short Vizing chain and successfully extends the coloring. Lemma 8.4 gives a formal description of this procedure.

**Algorithm 7:** $\textsf{ExtendColoring}(\chi, u, v)$

```
1  Let $c_u \in \textsf{miss}_\chi(u)$ and $c_v \in \textsf{miss}_\chi(v)$
2  repeat
3      Let $P = u_1, \ldots, u_{k+1}$ be the alternating path considered by $\textsf{Vizing}(\chi, u, v, c_u, c_v)$
4      if $k \leq 2L + 2$ then
5          $\chi \leftarrow \textsf{Vizing}(\chi, u, v, c_u, c_v)$
6          return $\chi$
7      Sample $i \sim [L]$ independently and u.a.r.
8      $(\chi, u, v, c_u, c_v) \leftarrow \textsf{TruncatedVizing}(\chi, u, v, c_u, c_v, i+1)$
```

Using standard data structures, we can implement each iteration of Algorithm 7 to run in time $\tilde{O}(L) \leq \tilde{O}(\Delta^2 + \sqrt{\Delta n})$. The following lemma, which we prove in Section 8.3, shows that the algorithm terminates after $\tilde{O}(1)$ many iterations with constant probability.

**Lemma 8.4.** *Algorithm 7 terminates within $\ell + 1$ iterations with probability at least $1/(160 \log n)$.*

## 8.3 Analysis

In order to analyse our algorithm, we define a (possibly infinite) rooted meta-tree $\mathcal{T}$ which captures information about all of the different possible executions of our algorithm.

**The Meta-Tree $\mathcal{T}$**

The meta-tree $\mathcal{T}$ is rooted at a meta-node $r$. Every meta-node $x \in \mathcal{T}$ corresponds to an iteration of Algorithm 7, and the root-to-$x$ path from $r$ to $x$ in $\mathcal{T}$ corresponds to the execution of Algorithm 7 leading to this iteration. For each such meta-node $x$, we denote the state of an object $\Phi$ in Algorithm 7 at the start of the corresponding iteration of the algorithm by $\Phi^{(x)}$, e.g. $\chi^{(x)}$ denotes the coloring $\chi$ at the start of the corresponding iteration of the algorithm. Given a meta-node $x$, $P^{(x)}$ denotes the alternating path considered during the corresponding iteration of the algorithm. Consider the following definition:

- We say that $x$ is *terminal* if the length of $P^{(x)}$ is at most $2L + 2$.

Each meta-node $x$ has no children in $\mathcal{T}$ if it is terminal. Otherwise, $x$ has exactly $L$ children in $\mathcal{T}$, each one corresponding to a different choice of edge in $P^{(x)}$ for truncating the alternating path.[13]

**Random Walks in $\mathcal{T}$:** We first note that there is a one-to-one correspondence between root-to-leaf paths in $\mathcal{T}$ and executions of Algorithm 7. Furthermore, an execution of our algorithm defines a *random walk* on $\mathcal{T}$ that starts at the root and, at each step, independently and uniformly samples a random child of the current meta-node (as long as the current meta-node is not terminal). Our algorithm successfully extends the coloring $\chi$ if and only if this random walk encounters a terminal meta-node. To this end, we show that such a random walk encounters a terminal meta-node within $O(\log n)$ steps with probability $\Omega(1/\log n)$, proving Lemma 8.4.

**Classifications of Meta-Nodes:** Given a meta-node $x \in \mathcal{T}$, let $\tau^{(x)}$ denote the colors of the alternating path $P^{(x)}$ that we consider during this iteration. We refer to any pair of colors as a *type*, and to $\tau^{(x)}$ as the type of $x$. Let $\tilde{P}^{(x)}$ denote the alternating path obtained by truncating the $\tau^{(x)}$-alternating path $P^{(x)}$ after the first $L + 1$ edges.[14] Let $r = x_0, \dots, x_i = x$ denote the root-to-$x$ path in $\mathcal{T}$. Consider the following definitions:

- We say that $x$ is *$\alpha$-dirty* if $\tau^{(x)} \cap (\tau^{(x_0)} \cup \dots \cup \tau^{(x_{i-2})}) \neq \emptyset$ and $\tau^{(x)} \cap \tau^{(x_{i-1})} = \emptyset$.
- We say that $x$ is *$\beta$-dirty* if $\tau^{(x)} = \tau^{(x_{i-1})}$.
- We say that $x$ is *$\alpha$-contaminated* (resp. *$\beta$-contaminated*) if it at least a $1/(10\ell)$ proportion of its children are *$\alpha$-dirty* (resp. *$\beta$-dirty*).
- We say that $x$ is *damaged* if it is **not** dirty and $P^{(x)}$ is **not** a maximal $\tau^{(x)}$-alternating path in $\chi^{(r)}$ such that $\chi^{(r)}(e) = \chi^{(x)}(e)$ for all $e \in P^{(x)}$.

Note that a meta-node $x$ cannot be both $\alpha$-dirty and $\beta$-dirty, but it can be both $\alpha$-contaminated and $\beta$-contaminated. We say that a meta-node $x$ is dirty (resp. contaminated) if it is $\alpha$-dirty or $\beta$-dirty (resp. $\alpha$-contaminated or $\beta$-contaminated). Furthermore, it follows from Lemma 8.3 that either $\tau^{(x)} = \tau^{(x_{i-1})}$ or $\tau^{(x)} \cap \tau^{(x_{i-1})} = \emptyset$, so $x$ is not dirty if and only if $\tau^{(x)} \cap (\tau^{(x_0)} \cup \dots \cup \tau^{(x_{i-1})}) = \emptyset$.

[13] Intuitively, one can visualise constructing $\mathcal{T}$ by starting at the root $r$, where $\chi^{(r)} = \chi$ and $(u^{(r)}, v^{(r)}) = (u, v)$, and considering all possible executions of Algorithm 7 for the different random choices made by the algorithm.

[14] Note that the alternating path $\tilde{P}^{(x)}$ is *not* maximal, while the alternating path $P^{(x)}$ *is* maximal.

**Properties of the Meta-Tree $\mathcal{T}$**

We now describe some properties of the meta-tree $\mathcal{T}$ that will be useful while analyzing the behaviour of random walks in $\mathcal{T}$. We begin with the following lemmas which show that contaminated meta-nodes have lots of terminal children.

**Lemma 8.5.** *Let $x$ be an $\alpha$-contaminated meta-node within the first $\ell$ levels of $\mathcal{T}$. Then at least a $1/(40\ell)$ proportion of the children of $x$ are terminal.*

*Proof.* Let $r = x_0, \dots, x_i = x$ denote the root-to-$x$ path in $\mathcal{T}$. Since $x$ is $\alpha$-contaminated, it has at least $L/(10\ell)$ $\alpha$-dirty children $y_1, \dots, y_k$. Since $i \le \ell$, at most $2\ell$ colors appear in $\tau^{(x_0)} \cup \dots \cup \tau^{(x_{i-2})}$. Thus, each of the $\alpha$-dirty children $y_1, \dots, y_k$ has one of at most $2\ell(\Delta+1) \le 4\ell\Delta$ many types. Since $\tau^{(x)}$ is disjoint from the types of the $\alpha$-dirty children of $x$, we have that the alternating paths $P^{(y_1)}, \dots, P^{(y_k)}$ corresponding to the children of $x$ are all maximal alternating paths with the same colors in $\chi^{(y_1)}$ and thus have a total length of at most $4\ell\Delta n$ (since the total length of the maximal alternating paths of any type is at most $n$). Furthermore, each alternating path in the collection $P^{(y_1)}, \dots, P^{(y_k)}$ appears at most twice (once in each orientation), so we have at least $L/(20\ell)$ distinct alternating paths. Thus, by an averaging argument, at least half of these paths have length at most $8\ell\Delta n \cdot (20\ell/L) \le L$, and hence correspond to terminal meta-nodes. □

**Lemma 8.6.** *Let $x$ be a $\beta$-contaminated meta-node within the first $\ell$ levels of $\mathcal{T}$. Then at least a $1/(20\ell)$ proportion of the children of $x$ are terminal.*

*Proof.* Since $x$ is $\beta$-contaminated, it has at least $L/(10\ell)$ $\beta$-dirty children $y_1, \dots, y_k$. Let $\tau$ denote the type of $x$ (which is the same as the types of $y_1, \dots, y_k$). Consider the coloring $\chi^{(y_1)}$ and the uncolored edge $(u^{(y_1)}, v^{(y_1)})$. Let $P'$ (resp. $P''$) denote the $\tau$-alternating path starting at $u^{(y_1)}$ (resp. $v^{(y_1)}$), and let $w'$ (resp. $w''$) denote it's other endpoint. Let $Q$ denote the path or cycle obtained by concatenating $P'$, $(u^{(y_1)}, v^{(y_1)})$, and $P''$. Then the colorings $\chi^{(y_1)}, \dots, \chi^{(y_k)}$ only differ on the colors of the edges in $Q$. Furthermore, in each of these colorings, $Q$ always consists of either 1 or 2 $\tau$-alternating paths and an uncolored edge.

Now, consider some meta-node $y_i$ and it's corresponding vertex $u^{(y_i)}$ in $Q$. There is an edge $f^{(y_i)}$ connecting $u^{(y_i)}$ to the first vertex on the $\tau$-alternating path $P^{(y_i)}$. We either have that the other endpoint of $f^{(y_i)}$ is $w'$ or $w''$, or that $P^{(y_i)}$ is vertex disjoint from $Q$. In the former case, we say that $y_i$ is a *bad extension point*. Since there are only $2\Delta$ many edges incident on $w'$ and $w''$, there are at most $2\Delta$ bad extension points. Let $\{z_1, \dots, z_{k'}\} \subseteq \{y_1, \dots, y_k\}$ be the subset of the $y_i$ that are not bad extension points. Now, consider the collection of $\tau$-alternating paths $P^{(z_1)}, \dots, P^{(z_{k'})}$. Since these are all vertex disjoint from $Q$, they are all maximal $\tau$-alternating paths with the same colors in $\chi^{(y_1)}$, and hence have a total length of at most $n$ without counting repeated paths. Furthermore, each alternating path in the collection $P^{(z_1)}, \dots, P^{(z_{k'})}$ appears at most $2\Delta$ times since each endpoint of each path has maximum degree $\Delta$, and thus can only be incident on at most $\Delta$ edges $f^{(y_i)}$ connecting the path to some $u^{(y_i)}$. Thus, their total length (counting repeated paths) is at most $2\Delta n$. By an averaging argument, at least half of these paths have length at most

$$\frac{4\Delta n}{k'} \le 4\Delta n \cdot \frac{20\ell}{L} \le L$$

since $k' \ge L/(10\ell) - 2\Delta \ge L/(20\ell)$. It follows that at least a $1/(20\ell)$ proportion of the children of $x$ are terminal. □

The following lemma shows that meta-nodes cannot have many damaged children.

**Lemma 8.7.** *Let $x$ be a meta-node within the first $\ell$ levels of $\mathcal{T}$. Then at most a $1/(10\ell)$ proportion of the children of $x$ are damaged.*

*Proof.* We begin with the following structural claim about alternating paths.

**Claim 8.8.** *Let $X \subseteq [\Delta+1]$ a be set of at most $2\ell$ colors and let $\chi$ and $\chi'$ be $(\Delta+1)$-edge colorings such that the set $D := \{e \in E \mid \{\chi(e), \chi'(e)\} \setminus X \neq \emptyset, \chi(e) \neq \chi'(e)\}$ has size at most $\Delta\ell$. For any $(\Delta+1)$-edge coloring $\tilde{\chi}$, let $\mathcal{P}(\tilde{\chi})$ denote the set of all maximal alternating paths w.r.t. $\tilde{\chi}$ with colors in $[\Delta+1] \setminus X$. Then we have that $|\mathcal{P}(\chi) \oplus \mathcal{P}(\chi')| \leq 6\Delta^2\ell$.*

*Proof.* Consider the following process: Start with the coloring $\chi_0 := \chi$, uncolor each of the edges in $D$ one by one to obtain a sequence of colorings $\chi_1, \dots, \chi_{|D|}$, then recolor each of the edges in $D$ with the color it receives under $\chi'$ to obtain a sequence of colorings $\chi_{|D|+1}, \dots, \chi_{2|D|}$. Then we can observe that $\mathcal{P}(\chi') = \mathcal{P}(\chi_{2|D|})$ since any edge that receives a different color under $\chi'$ and $\chi_{2|D|}$ has colors in $X$ under both. Furthermore, for any $0 \leq i < 2|D|$, we have that $|\mathcal{P}(\chi_i) \oplus \mathcal{P}(\chi_{i+1})| \leq 3\Delta$. This is because each edge can belong to at most $\Delta$ different maximal alternating paths (one for each other color) and changing the color of an edge can cause 2 maximal alternating paths to merge into 1, and vice versa. It follows that

$$|\mathcal{P}(\chi) \oplus \mathcal{P}(\chi')| \leq \sum_{i=0}^{2|D|-1} |\mathcal{P}(\chi_i) \oplus \mathcal{P}(\chi_{i+1})| \leq 6\Delta|D| \leq 6\Delta^2\ell.$$

□

Let $x$ be a meta-node within the first $\ell$ levels of $\mathcal{T}$ with root-to-$x$ path $x_0, \dots, x_i$. If $x$ is terminal, then we are done. Otherwise, let $y_1, \dots, y_k$ denote the children of $x$ whose types are disjoint from the types $\tau^{(x_0)}, \dots, \tau^{(x_i)}$ of its ancestors (note that any child of $x$ which is not in $\{y_1, \dots, y_k\}$ is dirty and hence not damaged). Let $e \in E$ be an edge such that $\chi^{(r)}(e) \notin \tau^{(x_0)} \cup \dots \cup \tau^{(x_i)}$ and $\chi^{(r)}(e) \neq \chi^{(y_j)}(e)$ for some $y_j$. Then we know that $e$ was involved in a fan construction in one of the iterations corresponding to the meta-nodes $x_0, \dots, x_i$. Since there are at most $\Delta$ such edges per iteration, there are at most $\Delta\ell$ such edges in total. We can observe that each alternating path in the collection $P^{(y_1)}, \dots, P^{(y_k)}$ appears at most twice (once in each orientation) and that each of these paths receives the same colors and is a maximal alternating path in each of the colorings $\chi^{(y_1)}, \dots, \chi^{(y_k)}$. Thus, applying the preceding claim with $X = \tau^{(x_0)} \cup \dots \cup \tau^{(x_i)}$, $\chi = \chi^{(r)}$ and $\chi' = \chi^{(y_1)}$, it follows that at most $6\Delta^2\ell$ of the alternating paths in $P^{(y_1)}, \dots, P^{(y_k)}$ are not maximal alternating paths with the same colors in $\chi^{(r)}$. Hence, at most $12\Delta^2\ell$ of the meta-nodes $y_1, \dots, y_k$ are damaged. Since $12\Delta^2\ell/L \leq 1/(10\ell)$, it follows that at most an $1/(10\ell)$ proportion of the children of $x$ are damaged. □

The following lemma is crucial for showing that there cannot be many long walks that do not contain any dirty or damaged meta-nodes.

**Lemma 8.9.** *Let $\tau_0, \dots, \tau_i$ be a sequence of types. Then there exist at most $n$ meta-nodes $x$ with root-to-$x$ path $x_0, \dots, x_i$ such that $\tau^{(x_j)} = \tau_j$ and $x_j$ is not dirty, damaged or terminal for all $x_j$.*

*Proof.* Let $\Gamma(\tau_0, \dots, \tau_i)$ denote the set of all such meta-nodes. We now prove the following claim.

**Claim 8.10.** *For any distinct meta-nodes $x, y \in \Gamma(\tau_0, \dots, \tau_i)$, the $\tau_i$-alternating paths $\tilde{P}^{(x)}$ and $\tilde{P}^{(y)}$ are vertex disjoint.*

*Proof.* We prove this by induction. For any type $\tau_0$ we have that $\Gamma(\tau_0) \subseteq \{x_0\}$. Thus, this claim holds trivially for $i = 0$. For the inductive step, suppose that the claim holds for some $0 \le j-1 \le i-1$. If the type $\tau_j$ shares a color with any of the types $\tau_0, \dots, \tau_{j-1}$, then $\Gamma(\tau_0, \dots, \tau_j) = \emptyset$ since the types of the meta-nodes on the root-to-$x$ path of a non-dirty meta-node $x$ must be disjoint. Thus, we can assume that the types $\tau_0, \dots, \tau_j$ are disjoint. Now, let $x, y \in \Gamma(\tau_0, \dots, \tau_j)$ and let $(u^{(x)}, v^{(x)})$ and $(u^{(y)}, v^{(y)})$ be the uncolored edges in the iterations of our algorithm corresponding to meta-nodes $x$ and $y$ respectively. By the induction hypothesis, the vertices $u^{(x)}$ and $u^{(y)}$ are distinct since they either lie on different positions of the same $\tau_{j-1}$-alternating path or on different vertex disjoint paths. The $\tau_j$-alternating paths $P^{(x)}$ and $P^{(y)}$ constructed by our algorithm have $u^{(x)}$ and $u^{(y)}$ as endpoints respectively. Since $x$ and $y$ are not dirty and not damaged, these alternating paths are also maximal $\tau_j$-alternating paths with the exact same colors under $\chi^{(r)}$, so they are either distinct paths or the same path but with different orientations. In the first case, $\tilde{P}^{(x)}$ and $\tilde{P}^{(y)}$ are clearly disjoint. In the second case, we note that $x$ and $y$ are not terminal; hence, $P^{(x)}$ has length at least $2L + 3$, so $\tilde{P}^{(x)}$ and $\tilde{P}^{(y)}$ are also vertex disjoint since they have length at most $L + 1$. □

Thus, the size of $\Gamma(\tau_0, \dots, \tau_i)$ is upper bounded by the maximum size of a collection of vertex disjoint paths, which is at most $n$. □

## Analyzing the Random Walk

Consider the following definition of a *good* walk.

- We say that a walk $x_0, x_1, \dots$ in the meta-tree $\mathcal{T}$ is *good* if it encounters either a terminal or contaminated meta-node within the first $\ell$ steps.

The following lemma bounds the probability that a good random walk has length at most $\ell + 1$.

**Lemma 8.11.** *Let $x_0, x_1, \dots$ be a random walk in $\mathcal{T}$. Given that the random walk is good, it has length at most $\ell + 1$ with probability $1/(40\ell)$.*

*Proof.* Since the random walk is good, we know that $x_i$ is terminal or contaminated for some $i$. If $x_i$ is terminal, then the random walk has length $i \le \ell$. If $x_i$ is $\alpha$-contaminated, then it follows from Lemma 8.5 that $x_{i+1}$ is terminal (and hence that the random walk has length at most $i + 1$) with probability at least $1/(40\ell)$. Similarly, if $x_i$ is $\beta$-contaminated, then it follows from Lemma 8.6 that the random walk has length at most $i + 1$ with probability at least $1/(20\ell)$. □

It follows from Lemma 8.11 that it suffices to lower bound the probability that a random walk in the meta-tree $\mathcal{T}$ is good. More precisely, we have the following. Let $\mathcal{W}_\mathcal{T}$ denote the collection of all walks in $\mathcal{T}$ and let $(x_i)_i \sim \mathcal{W}_\mathcal{T}$ denote a random walk. Then we have that

$$\begin{aligned}\Pr_{(x_i)_i \sim \mathcal{W}_\mathcal{T}}[|(x_i)_i| \le \ell + 1] &\ge \Pr_{(x_i)_i \sim \mathcal{W}_\mathcal{T}}[|(x_i)_i| \le \ell + 1 \mid (x_i)_i \text{ is good}] \cdot \Pr_{(x_i)_i \sim \mathcal{W}_\mathcal{T}}[(x_i)_i \text{ is good}] \\ &\ge \frac{1}{40\ell} \cdot \Pr_{(x_i)_i \sim \mathcal{W}_\mathcal{T}}[(x_i)_i \text{ is good}]. \qquad (10)\end{aligned}$$

Here, the first inequality follows from conditioning on the event that the random walk $(x_i)_i$ is good, and the second inequality follows from Lemma 8.11.

**The Meta-Subtree $\mathcal{T}'$:** We define a meta-subtree $\mathcal{T}'$ of $\mathcal{T}$ as follows: Start with the meta-tree $\mathcal{T}$ and remove all of the (strict) descendants of contaminated meta-nodes. We can observe that the probability of a random walk in $\mathcal{T}$ being good is the same as the probability of a random walk in $\mathcal{T}'$ having length at most $\ell$, i.e. that

$$\Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}}}[(x_i)_i \text{ is good}] = \Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}'}}[|(x_i)_i| \leq \ell]. \tag{11}$$

To see why this is true, consider a mapping $\phi : \mathcal{W}_{\mathcal{T}} \longrightarrow \mathcal{W}_{\mathcal{T}'}$ which maps a walk $(x_i)_i$ in $\mathcal{T}$ to a walk $\phi((x_i)_i)$ which is defined as the prefix of $(x_i)_i$ which is contained in $\mathcal{T}'$. Then we can observe that $(x_i)_i$ is good if and only if $|\phi((x_i)_i)| \leq \ell$.

**The Meta-Subtree $\mathcal{T}''$:** We define a meta-subtree $\mathcal{T}''$ of $\mathcal{T}'$ as follows: Start with the meta-tree $\mathcal{T}'$ and remove all of the meta-nodes that are dirty or damaged (and their descendants) from $\mathcal{T}'$. It turns out that bounding the length of random walks in $\mathcal{T}''$ is much easier than bounding the length of random walks in $\mathcal{T}'$. Thus, we only want to consider random walks in $\mathcal{T}'$ which are also contained in $\mathcal{T}''$ for the first $\ell$ steps. In particular, we use the following bound

$$\Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}'}}[|(x_i)_i| \leq \ell] \geq \Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}''}}[|(x_i)_i| \leq \ell] \cdot \Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}'}}[(x_i)_{i \leq \ell} \subseteq \mathcal{T}'']. \tag{12}$$

Given any meta-node $x \in \mathcal{T}'$ that has children, we know that at most a $1/(5\ell)$ proportion of the children of $x$ are dirty since $x$ is not contaminated.[15] Furthermore, it follows from Lemma 8.7 that at most a $1/(10\ell)$ proportion of the children of $x$ are damaged. It follows that a random walk in $\mathcal{T}'$ does not encounter any dirty or damaged meta-nodes within the first $\ell$ steps with probability at least $(1 - 3/(10\ell))^\ell \geq 1 - 3/10 = 7/10$. Thus, we have that

$$\Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}'}}[(x_i)_{i \leq \ell} \subseteq \mathcal{T}''] \geq \frac{7}{10}. \tag{13}$$

Combining these inequalities, it follows that

$$\Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}}}[|(x_i)_i| \leq \ell + 1] \geq \frac{1}{80\ell} \cdot \Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}''}}[|(x_i)_i| \leq \ell]. \tag{14}$$

We now lower bound the probability that a random walk in $\mathcal{T}''$ terminates within at most $\ell$ steps.

**Random Walks in $\mathcal{T}''$:** Let $\mathcal{T}''_{\ell+1}$ denote the meta-nodes at depth $\ell + 1$ in $\mathcal{T}''$. Given a walk $(x_i)_i$ in $\mathcal{T}''$, $(x_i)_i$ has length greater than $\ell$ if and only if $(x_i)_i$ contains some meta-node from $\mathcal{T}''_{\ell+1}$, thus

$$\Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}''}}[|(x_i)_i| > \ell] = \Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}''}}[\mathcal{T}''_{\ell+1} \cap (x_i)_i \neq \emptyset].$$

**Lemma 8.12.** *For any $x \in \mathcal{T}''_{\ell+1}$, we have that $\Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}''}}[x \in (x_i)_i] \leq (2/L)^{\ell+1}$.*

*Proof.* Give some non-leaf meta-node $y \in \mathcal{T}''$, we can see that $y$ has at least $L \cdot (1 - 3/(10\ell)) \geq L/2$ children in $\mathcal{T}''$. This is because $y$ is not contaminated (otherwise it would be a leaf) and hence has at most $L/(5\ell)$ dirty children in $\mathcal{T}'$ and at most $L/(10\ell)$ damaged children in $\mathcal{T}'$. Thus, the probability that a random walk in $\mathcal{T}''$ contains a child $y'$ of $y$ given that it contains $y$ is at most $1/(L/2)$. It follows that the probability that a random walk in $\mathcal{T}''$ contains a meta-node $x \in \mathcal{T}''_{\ell+1}$ is at most $(2/L)^{\ell+1}$. □

[15] Recall that since $x$ is neither $\alpha$ nor $\beta$-contaminated, at most a $1/(10\ell)$ proportion of its children at $\alpha$-dirty and at most a $1/(10\ell)$ proportion of its children at $\beta$-dirty

**Lemma 8.13.** *$|\mathcal{T}''_{\ell+1}| \leq \Delta^{2\ell} nL$.*

*Proof.* Recall the definition of $\Gamma$ from the proof of Lemma 8.9. Let $x \in \mathcal{T}''_{\ell+1}$ and let $x_0, \dots, x_{\ell+1}$ denote the root-to-$x$ path in $\mathcal{T}$. Then it must be the case that $x_{\ell+1}$ is the child of some meta-node in $\Gamma(\tau^{(x_0)}, \dots, \tau^{(x_\ell)})$ since it's parent cannot be terminal. It follows that every meta-node in $\mathcal{T}''_{\ell+1}$ is a child of some meta-node in

$$\bigcup_{\tau \in \binom{[\Delta+1]}{2}^{\ell}} \Gamma(\tau).$$

Since any meta-node has at most $L$ children and $|\Gamma(\tau)| \leq n$ by Lemma 8.9, it follows that

$$|\mathcal{T}''_{\ell+1}| \leq L \cdot \sum_{\tau \in \binom{[\Delta+1]}{2}^{\ell}} |\Gamma(\tau)| \leq L \cdot (\Delta^2)^{\ell} \cdot n \leq \Delta^{2\ell} nL.$$

□

It follows from Lemmas 8.12 and 8.13 that

$$\Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}''}}[\mathcal{T}''_{\ell+1} \cap (x_i)_i \neq \emptyset] \leq \sum_{x \in \mathcal{T}''_{\ell+1}} \Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}''}}[x \in (x_i)_i] \leq |\mathcal{T}''_{\ell+1}| \cdot \left(\frac{2}{L}\right)^{\ell+1} \leq nL \cdot \left(\frac{2\Delta^2}{L}\right)^{\ell+1} \leq \frac{1}{2}.$$

Hence, we have that

$$\Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}''}}[|(x_i)_i| \leq \ell] = 1 - \Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}''}}[|(x_i)_i| > \ell] \geq \frac{1}{2}. \tag{15}$$

Combining Equations (14) and (15), it follows that

$$\Pr_{(x_i)_i \sim \mathcal{W}_{\mathcal{T}}}[|(x_i)_i| \leq \ell + 1] \geq \frac{1}{160\ell}. \tag{16}$$

In other words, a random walk in the meta-tree $\mathcal{T}$ terminates within $\ell + 1$ steps with probability at least $1/(80\ell)$.

## 8.4 Proof of Theorem 8.1

Suppose that we run Algorithm 7 for $\kappa = (\ell + 1) \cdot 1600 \log^2 n$ iterations. It follows from Lemma 8.4 that, given an arbitrary input, the probability of the algorithm successfully extending the coloring within $\ell + 1$ iterations is at least $1/(160 \log n)$. Thus, we can split these $\kappa$ updates into batches of $\ell + 1$ iterations, where the probability that the algorithm successfully extends $\chi$ within some batch is at least $1/(160 \log n)$. Thus, the probability that the algorithm fails to extend the coloring within $\kappa$ iterations it at most

$$\left(1 - \frac{1}{160 \log n}\right)^{1600 \log^2 n} \leq \exp\left(-\frac{1600 \log^2 n}{160 \log n}\right) = \frac{1}{n^{10}}.$$

In the low probability event that the algorithm does not successfully extend the coloring within $\kappa$ iterations, then we can extend it in $O(n)$ time using Vizing's original algorithm. Thus, our algorithm runs in time $\kappa \cdot \tilde{O}(L) \leq \tilde{O}(\Delta^2 + \sqrt{\Delta n})$ both with high probability and in expectation.

# 9 Color Extension to Stars

The idea of *coloring* stars was first proposed in [BCC+24]. In this section, we provide a simpler algorithm with improved runtime bound.

**Lemma 9.1** (Lemma 6.1 Restated)**.** *There is an algorithm* `ColorLightStars` *that, given a graph $G$, a partial $(\Delta+1)$-edge coloring $\chi$ of $G$ and subset $U^\star \subseteq V$ such that:*

- $|\mathsf{miss}_\chi(u)| \geq d$ *for all $u \in U^\star$ for some positive integer $d \in \mathbb{N}$,*
- *there are $\lambda$ uncolored edges incident on $U^\star$,*

*extends the coloring $\chi$ to $\Omega(\lambda)$ uncolored edges incident on $U^\star$ in $\tilde{O}(\lambda\Delta + \Delta m/d)$ expected time.*

## 9.1 Recap on Vizing's algorithm

Our algorithm will again use the original Vizing's algorithm in [Viz64] that extends any partial edge coloring $\chi$ by one more colored edge. Since we will not need the involved modifications as in previous sections and only use the basic version in a slightly different way, for reader's convenience, we will describe it again here with simpler terms.

Let $(u,v) \in E$ be any uncolored edge under $\chi$; that is, $\chi(u,v) = \perp$. For any vertex $w \in V$, specify an arbitrary color $c_\chi(w) \in \mathsf{miss}_\chi(w)$, and also specify an arbitrary color $z \in \mathsf{miss}_\chi(v)$ different from $c_\chi(v)$. Then find a sequence of distinct neighbors $v = v_0, v_1, v_2, \ldots, v_k$ of $u$ such that the following holds; this sequence $v_1, v_2, \ldots, v_k$ is usually called a *Vizing fan.*

- For any $2 \leq i \leq k$, $\chi(u, v_i) = c_\chi(v_{i-1})$, and $z = \chi(u, v_1)$.
- Either $c_\chi(v_k) \in \mathsf{miss}_\chi(u)$, or there exists an index $1 \leq j < k$ such that $\chi(u, v_j) = c_\chi(v_k)$.

If $c_\chi(v_k) \in \mathsf{miss}_\chi(u)$, then rotate the coloring around $u$ as: $\chi(u, v_i) \leftarrow \chi(u, v_{i+1}), 0 \leq i < k$, and $\chi(u, v_k) \leftarrow c_\chi(v_k)$.

Now, let us assume $c_\chi(v_k) \notin \mathsf{miss}_\chi(u)$, so there must exist an index $1 \leq j < k$ such that $\chi(u, v_j) = c_\chi(v_k)$. Take an arbitrary color $x \in \mathsf{miss}_\chi(u)$, and define $y = \chi(u, v_j)$. Let $P$ be the maximal $\{x, y\}$-alternating path beginning at $u$.

(1) $P$ does not end at $v_{j-1}$.

Apply a rotation operation: $\chi(u, v_i) \leftarrow \chi(u, v_{i+1}), 0 \leq i < j$, and flip the maximal $\{x, y\}$-alternating path $P$. Finally, assign $\chi(u, v_j) \leftarrow x$.

(2) $P$ ends at $v_{j-1}$.

Flip the color of the maximal $\{x, y\}$-alternating path from $u$, then apply a rotation operation: $\chi(u, v_i) \leftarrow \chi(u, v_{i+1}), 0 \leq i < k$, then assign color $\chi(u, v_k) \leftarrow y$.

The runtime of Vizing's procedure is bounded by $\tilde{O}(\Delta + |P|)$.

## 9.2 Algorithm Description

Throughout our color extension procedure, we will be repeatedly assigning proper colors to edges in $S$ defined below while decreasing the size of some sets $\mathsf{miss}_\chi(u), u \in U^\star$. Therefore, we will keep a terminal subset $W \subseteq U^\star$ with the property that for any $u \in W$, we have $|\mathsf{miss}_\chi(u)| \geq d/2$. Initially, set $W \leftarrow U^\star$.

We will maintain the following data structures.

- A set $S$ of currently uncolored edges incident on $W$. At the beginning we choose $S$ to be those $\lambda$ uncolored edges incident on $W$, which is initially $U^\star$. While initially $S$ is the set of uncolored edges incident on $U^\star$, throughout the execution of the algorithm some uncolored edges may be removed from $S$; however, as we will argue (refer to Section 9.4), whenever a constant fraction of edges have been removed from $S$, it implies that a constant fraction of the original set of $\lambda$ uncolored edges have been colored.

  We assign all edges in $S$ a direction: for each edge $(v, u) \in S$ such that $u \in W$, orient this edge from $v$ to $u$; if both $u, v$ are in $W$, orient this edge arbitrarily.

- For each vertex $v \in V$, our algorithm will explicitly maintain a color $c_\chi(v) \in \mathsf{miss}_\chi(v)$.

- For each vertex $u \in W$ and each of its neighbor $v$ such that $\chi(u, v) = \perp$, maintain a tentative color $\mathsf{clr}_\chi(v, u) \in \mathsf{miss}_\chi(v)$ with the following two properties.

  - $\mathsf{clr}_\chi(v, u) \neq c_\chi(v)$ for all $u \in W$ such that $\chi(u, v) = \perp$;
  - $\mathsf{clr}_\chi(v, u) \neq \mathsf{clr}_\chi(v, u')$ for all pairs of distinct neighbors $u, u'$ of $v$ in $W$ such that $\chi(u, v) = \chi(u', v) = \perp$.

  Note that such choices of $\mathsf{clr}_\chi(\cdot, \cdot)$ which are compatible with $c_\chi(\cdot)$ always exist since the palette size is $\Delta + 1$.

- For each color $x$, maintain a list $\mathsf{lst}_\chi(u, x) = \{(v, u) \in S \mid u \in W, x = \mathsf{clr}_\chi(v, u)\}$.

We first argue that these data structures can be maintained efficiently.

**Lemma 9.2.** *Suppose the partial coloring $\chi$ and the set $S$ has undergone $k$ changes. Then, the data structures $\mathsf{miss}_\chi(\cdot), c_\chi(\cdot), \mathsf{clr}_\chi(\cdot, \cdot)$ can be maintained in $\tilde{O}(k)$ time.*

*Proof.* For each vertex $v \in V$, maintain a list of colors $\mathsf{clr}_\chi(v) = \{\mathsf{clr}_\chi(v, u) \mid (v, u) \in S\}$, and also maintain the set $\mathsf{miss}_\chi(v) \setminus \mathsf{clr}_\chi(v)$. To recover the data structures after $k$ changes to $\chi$ and $S$, consider the following two steps.

(1) Temporarily assign $c_\chi(v) \leftarrow \perp$. Initialize a color set $C_v \leftarrow \mathsf{miss}_\chi(v) \setminus \mathsf{clr}_\chi(v)$ using our old version of the data structure. Also, let $S_v$ be the set of new edges $(v, u) \in S$ added to $S$, and temporarily assign $\mathsf{clr}_\chi(v, u) \leftarrow \perp$ for the moment.

    Then, for each old color $x$ removed around $v$ in the new $\chi$, add $x$ to $C_v$. For each new color $x$ added around $v$, remove $x$ from $C_v$, and also if there is currently an edge $(v, u) \in S$ such that $\mathsf{clr}_\chi(v, u) = x$, temporarilly assign $\mathsf{clr}_\chi(v, u) \leftarrow \perp$ and add $(v, u)$ to $S_v$. Each of these steps can be implemented in $O(\log \Delta)$ time.

(2) We can see that $S_v$ is currently the set of edges $(v, u) \in S$ such that $\mathsf{clr}_\chi(v, u) = \perp$, and $C_v = \mathsf{miss}_\chi(v) \setminus \mathsf{clr}_\chi(v)$. Since there are $\Delta + 1$ colors, we know that $|C_v| \geq |S_v| + 1$. Therefore, we can allocate $C_v$ properly to $\mathsf{clr}_\chi(v, u), (v, u) \in S_v$ as well as $c_\chi(v)$ without any conflicts.

The total runtime of the above two steps is $\tilde{O}(k)$. □

For each vertex $u \in U^\star$, define a directed graph $F_\chi(u)$ in the following way; this digraph $F_\chi(u)$ will not be maintained by our algorithm, but only computed on-the-fly when needed.

- **Vertices.** The vertex set of digraph $F_\chi(u)$ is the set of all colored neighbors $N_\chi(u)$ of $u$.
- **Edges.** For each $z, w \in N_\chi(u)$, if $\chi(u,w) = c_\chi(z)$, then add a directed edge $(z,w)$ to $F_\chi(u)$.

So by definition, the out-degree of each vertex in $F_\chi(u)$ is at most one, and so $F_\chi(u)$ is a directed pseudo-forest; in particular, note that any weakly connected component in $F_\chi(u)$ is a directed tree plus at most one extra edge.

**Definition 9.3** (uncolored edge classification)**.** *For any directed edge $(v,u) \in S$ such that $u \in W$, the directed edge $(v,u)$ is classified into one of the following four types.*

- *If $\mathsf{clr}_\chi(v,u) \in \mathsf{miss}_\chi(u)$, then the directed edge $(v,u)$ is called* ***ready****.*
- *Otherwise, if there exists another uncolored directed edge $(v',u) \in S$ such that $\mathsf{clr}_\chi(v \to u) = \mathsf{clr}_\chi(v' \to u)$, then the directed edge $(v,u)$ is called* ***social****.*
- *Otherwise, let $w \in N_\chi(u)$ be the unique vertex such that $\chi(u,w) = \mathsf{clr}_\chi(v,u)$. Let $T^w_\chi(u) \subseteq F_\chi(u)$ be the weakly connected component that contains vertex $w$. If $w$ is the only vertex $w' \in T^w_\chi(u)$ such that $|\mathsf{lst}_\chi(u, \chi(u,w'))| = 1$, then the directed edge $(v,u)$ is called* ***independent****.*
- *Otherwise, the directed edge $(v,u)$ is called* ***lonely****.*

Our main color extension algorithm is described as follows.

Define a length parameter $L = \frac{200\Delta m}{d\lambda}$. The algorithm consists of iterations of random color extension, which are repeated as long as $|S| \geq \frac{3\lambda}{4}$. We start by uniformly sampling an uncolored edge $(v,u) \in S$ incident on $W$, and then sampling uniformly at random a color $x \in \mathsf{miss}_\chi(u)$. The execution of an iteration splits into the following four cases.

(1) If $(v,u)$ is classified as ready, then assign $\chi(v,u) \leftarrow \mathsf{clr}_\chi(v,u)$. In this case, this iteration of color extension would be considered *successful*.

(2) Otherwise, check if $(v,u)$ is social, which can be done by checking if $|\mathsf{lst}_\chi(u, \mathsf{clr}_\chi(v,u))| > 1$. If so, let $P$ be the maximal $\{x,y\}$-alternating path starting from vertex $v$ where $y = \mathsf{clr}_\chi(v,u)$.

If $|P| \leq L$ and $P$ does not terminate at vertex $u$, then flip this maximal $\{x,y\}$-alternating path $P$, and assign $\chi(v,u) \leftarrow x$. In this case, this iteration of color extension would be considered successful.

(3) Otherwise, construct the entire graph $F_\chi(u)$ and check if $(v,u)$ is independent. If so, try to apply Vizing's procedure described in Section 9.1 to extend $\chi$ to $(v,u)$, where vertex $v$ uses the missing color $\mathsf{clr}_\chi(v,u)$, vertices $w$ use missing colors $c_\chi(w), \forall w \in N_\chi(u)$, and $u$ uses the missing color $x \in \mathsf{miss}_\chi(u)$. Let $Q$ be the maximal $\{x,z\}$-alternating path which is the Vizing chain for $(v,u)$; in Vizing's procedure, $Q$ could be an empty path.

If $|Q| \leq L$, then complete the Vizing procedure, which extends $\chi$ to $(v,u)$. In this case, this iteration of color extension would be considered successful.

(4) Otherwise, suppose $\chi(u,w) = \mathsf{clr}_\chi(v,u)$ and let $T^w_\chi(u) \subseteq F_\chi(u)$ be the weakly connected component containing $w$. Since $(v,u)$ is not independent, by Definition 9.3, there exists another edge $(v',u) \in S$ and $w' \in T^w_\chi(u)$, such that $\mathsf{clr}_\chi(v',u) = \chi(u,w')$ and $|\mathsf{lst}_\chi(u, \chi(u,w'))| = 1$. This implies that $(v',u) \in S$ is also lonely.

Let $T \subseteq T^w_\chi(u)$ be a spanning tree such that all edges are in the direction from children to parents in $T^w_\chi(u)$. Let $t \in T$ be the least common ancestor of $w, w'$ in tree $T$, and define $w_0(= v), w_1(= w), \dots, w_l = t$ and $w'_0(= v'), w'_1(= w'), \dots, w'_k = t$ be the tree paths from $w, w'$ to $t$. As $t$ is the least common ancestor, these two tree paths are internally disjoint.

Perform the following color shifts to $\chi$:

$$\chi(u, w_i) \leftarrow \chi(u, w_{i+1}), \forall\, 0 \leq i < l - 1$$

$$\chi(u, w'_j) \leftarrow \chi(u, w'_{j+1}), \forall\, 0 \leq j \leq k - 1$$

Then, uncolor the edges $\chi(u, w_{l-1}), \chi(u, w'_{k-1}) \leftarrow \perp$. After that, update the edge set

$$S \leftarrow S \cup \{(w_{l-1}, u), (w'_{k-1}, u)\} \setminus \{(v, u), (v', u)\}$$

Finally, assign tentative colors

$$\mathsf{clr}_\chi(w_{l-1}, u), \mathsf{clr}_\chi(w'_{k-1}, u) \leftarrow \chi(u, t)$$

After the above steps (in all four cases), we maintain the data structures regarding $\mathsf{clr}_\chi(\cdot,\cdot), c_\chi(\cdot), \mathsf{lst}_\chi(\cdot,\cdot)$ in the straightforward manner. Finally, if the size of some set $\mathsf{miss}_\chi(u_0), u_0 \in W$ drops below $d/2$, remove $u_0$ from $W$, and also remove all edges $(v, u_0)$ from $S$.

## 9.3 Runtime Analysis

**Lemma 9.4.** *Given any partial edge coloring $\chi$ of $G = (V, E)$ and any color $x \in \{1, 2, \dots, \Delta+1\}$, for any $y \in \{1, 2, \dots, \Delta+1\} \setminus \{x\}$, let $L_{x,y}$ denote the total length of all maximal $\{x, y\}$-alternating paths of lengths at least 2. Then $\sum_{y, y\neq x} L_{x,y} < 3m$.*

*Proof.* For any maximal $\{x, y\}$-alternating path $P$ such that $|P| \geq 2$, the number of edges in $P$ with color $x$ is at most twice the number of edges in $P$ with color $y$. Therefore, taking a summing over all $y$ such that $y \neq x$ and all such maximal $\{x, y\}$-alternating paths, we have

$$\sum_{y, y\neq x} L_{x,y} \leq 3 \cdot |\{e \in E \mid \chi(e) \neq x\}| < 3m.$$

□

**Lemma 9.5.** *The runtime of each iteration of color extension is $\tilde{O}(\Delta + L)$.*

*Proof.* First, let us analyze the runtime in each of the four cases.

- In Step (1), the runtime is $O(1)$.
- In Step (2), checking if an edge $(v, u) \in S$ is social takes $O(1)$ time. Checking if the length of the maximal alternating path $P$ exceeds $L$ takes $\tilde{O}(L)$ time by tracing this path $P$ until it ends or reaches length $L + 1$. Flipping this path also takes $\tilde{O}(L)$ time.
- In Step (3), constructing the entire digraph $F_\chi(u)$ takes $\tilde{O}(\Delta)$ time. Deciding if $(v, u)$ is independent also takes $\tilde{O}(\Delta)$ time by going over all vertices $w'$ in the weakly connected component $T^w_\chi(u) \subseteq F_\chi(u)$ containing the unique vertex $w \in N_\chi(u)$ such that $\chi(u, w) = \mathsf{clr}_\chi(v, u)$ and checking if $|\mathsf{lst}_\chi(u, \chi(u, w'))| = 1$. After that, running Vizing's procedure takes time $\tilde{O}(\Delta + L)$.

- In Step (4), finding the other edge $(v', u) \in F$ and $w' \in T^{w}_{\chi}(u)$ takes $\tilde{O}(\Delta)$ time. Later on, finding the two tree paths and shifting the colors also take time $\tilde{O}(\Delta)$.

In each of the four cases, to recover the validity of the data structures $\mathsf{clr}_{\chi}(\cdot,\cdot), c_{\chi}(\cdot), \mathsf{lst}_{\chi}(\cdot,\cdot)$, it takes $\tilde{O}(1)$ for each color change to $\chi$. As we have just proved that the total number of changes to $\chi$ and $S$ is bounded by $O(\Delta+L)$, the runtime to maintain $\mathsf{clr}_{\chi}(\cdot,\cdot), c_{\chi}(\cdot), \mathsf{lst}_{\chi}(\cdot,\cdot)$ would be $\tilde{O}(\Delta+L)$, according to Lemma 9.2. Finally, since each iteration adds one colored edge, $W$ loses at most 2 vertices in each iteration, which takes runtime $O(\Delta)$ to update $S$. □

So it suffices to analyze the total number of iterations. The following claim serves as the basis of our analysis.

**Lemma 9.6.** *During our algorithm, whenever we have $\mathsf{clr}_{\chi}(v_1, u_0) = \mathsf{clr}_{\chi}(v_2, u_0)$ for two different edges $(v_1, u_0), (v_2, u_0) \in S$, meaning that both $(v_1, u_0), (v_2, u_0)$ are social, they will stay in $S$ as two social edges until one of the following events occur.*

1. *Vertex $v_1$ or $v_2$ is incident on a newly colored edge $(v, u)$ with new color $\chi(v, u) \leftarrow \mathsf{clr}_{\chi}(v_1, u_0)$.*
2. *$v_1$ or $v_2$ becomes an endpoint of a flipped maximal alternating path $P$ or $Q$ computed in Step (2) or Step (3), which successfully extends $\chi$ to one more uncolored edge.*

*Proof.* If there are no successful color extensions since we have the equality $\mathsf{clr}_{\chi}(v_1, u_0) = \mathsf{clr}_{\chi}(v_2, u_0)$, then the partial coloring $\chi$ could only be modified by Step (4). Since both $(v_1, u_0)$ and $(v_2, u_0)$ are social, by the description of Step (4), $(v_1, u_0)$ and $(v_2, u_0)$ will stay in $S$ after this iteration of Step (4). Furthermore, during Step (4), the new color added around vertex $v_b, \forall b \in \{1, 2\}$, can only be $c_{\chi}(v_b)$ or $\mathsf{clr}_{\chi}(v_b, u'), u' \neq u$. Since our data structure has guaranteed that $c_{\chi}(v_b) \neq \mathsf{clr}_{\chi}(v_b, u_0)$ and $\mathsf{clr}_{\chi}(v_b, u') \neq \mathsf{clr}_{\chi}(v_b, u_0), u'_0 \neq u_0$, both tentative colors $\mathsf{clr}_{\chi}(v_1, u_0)$ and $\mathsf{clr}_{\chi}(v_2, u_0)$ will not change, and so $(v_1, u_0)$ and $(v_2, u_0)$ will remain social.

We may henceforth assume that there has been a successful color extension. Further, we assume that neither $v_1$ nor $v_2$ is incident on the newly colored edge $(v, u)$ such that $\chi(v, u)$ is assigned $\mathsf{clr}_{\chi}(v_1, u_0)$, otherwise event 1 occurs and we are done. Consequently, the sets $\mathsf{miss}_{\chi}(v_b), b \in \forall\{1, 2\}$, could change only in one of the following two cases.

- $v_b$ is an endpoint of the maximal alternating path $P$ or $Q$ in Step (2) (3). Thus, event 2 occurs.
- $v_b$ lies on the Vizing fan in Step (3).

  In this case, since we assumed that event 1 does not occur and so $(v_b, u_0)$ is not colored right away, the new possible color added around vertex $v_b$ can only be $c_{\chi}(v_b)$ or $\mathsf{clr}_{\chi}(v_b, u'_0)$ for some $u'_0 \neq u_0$. Since our data structure has guaranteed that $c_{\chi}(v_b) \neq \mathsf{clr}_{\chi}(v_b, u_0)$ and $\mathsf{clr}_{\chi}(v_b, u') \neq \mathsf{clr}_{\chi}(v_b, u_0), u'_0 \neq u_0$, the tentative color $\mathsf{clr}_{\chi}(v_b, u_0)$ does not change. Hence, $(v_1, u_0), (v_2, u_0)$ will remain social.

□

**Corollary 9.7.** *After each successful color extension, at most 8 social edges are turned into non-social ones.*

*Proof.* Consider any successful color extension, where an uncolored edge $(v, u)$ has obtained a color. By Lemma 9.6, a social edge $(v_1, u_0) \in S$ becomes no longer social after a successful iteration if one of the following conditions holds.

- Vertex $v_1$ is incident on the newly colored $(v,u)$ which happens to be assigned the same color $\chi(v,u) = \mathsf{clr}_\chi(v_1,u_0)$, or $v_1$ is the endpoint of the maximal alternating path $P$ or $Q$.

  In this case, there are at most 4 choices for this possible social edge $(v_1,v_0)$.

- There exists a unique social edge $(v_2,u_0) \neq (v_1,u_0)$ such that $\mathsf{clr}_\chi(v_1,u_0) = \mathsf{clr}_\chi(v_2,u_0)$, and $(v_2,u_0)$ is colored or $v_2$ is the endpoint of the maximal alternating path $P$ or $Q$.

  In this case, there are also at most 4 choices for this possible social edge $(v_1,v_0)$.

Therefore, overall, at most 8 social edges would no longer be social after a successful extension. □

At any moment, let $m_{\mathsf{rdy}}, m_{\mathsf{scl}}, m_{\mathsf{ind}}, m_{\mathsf{lon}}$ be the total number edges in $S$ that are ready, social, independent, and lonely, respectively, according to Definition 9.3, and therefore $|S| = m_{\mathsf{rdy}} + m_{\mathsf{scl}} + m_{\mathsf{ind}} + m_{\mathsf{lon}}$. Define a potential function $\Phi = 10|S| + m_{\mathsf{lon}}$. Consider any single iteration of random color extension which picks a random edge $(v,u) \in S$.

**Lemma 9.8.** *If $m_{\mathsf{rdy}} \geq |S|/4$, then with probability $\geq 1/4$, the edge $(v,u)$ gets colored in Step (1), and thus in this case the potential $\Phi$ drops by at least 2.*

*Proof.* This is straightforward according to our algorithm description. □

**Lemma 9.9.** *If $m_{\mathsf{scl}} \geq |S|/4$, then with probability $\geq \frac{3}{80}$, the edge $(v,u)$ gets colored in Step (2), and thus in this case the potential $\Phi$ drops by at least 2.*

*Proof.* For each social edge $e = (v,u)$ and color $x \in \mathsf{miss}_\chi(u)$, define $P_x(e)$ to be the maximal $\{x,y\}$-alternating path starting at vertex $v$ where $y = \mathsf{clr}_\chi(v,u)$.

Note that for any fixed $u \in W$ and $x \in \mathsf{miss}_\chi(u)$, the number of different paths $P_x(v,u)$ over social edges $(v,u)$ incident on $u$ is precisely the number of such edges, since any two distinct edges lead to different paths. Fixing an arbitrary missed color $x \in \mathsf{miss}_\chi(u)$ for each vertex $u \in W$, the total number of different paths over social edges incident on each vertex of $W$ with respect to its arbitrarily fixed missed color is at least $\frac{1}{2} \cdot m_{\mathsf{scl}}$, where the factor of $1/2$ stems from the fact that the same path may be counted twice, as two different social edges $e,'e$ could be the endpoints edges of the same alternating path. Since $|\mathsf{miss}_\chi(u)| \geq d/2$ for each $u \in W$ and as every different missing color $x$ at $u$ leads to distinct maximal alternating paths, the collection $\mathcal{P}$ of all different paths $P_x(e)$, over all $m_{\mathsf{scl}}$ social edges $e = (v,u), u \in W$ and over all $x \in \mathsf{miss}_\chi(u)$, satisfies:

$$|\mathcal{P}| \;\geq\; \frac{1}{2} \cdot \frac{d}{2} \cdot m_{\mathsf{scl}} \;=\; \frac{dm_{\mathsf{scl}}}{4}. \tag{17}$$

Define $\mathcal{P}'$ to be the sub-collection of $\mathcal{P}$, which includes all paths in $\mathcal{P}$ that do not terminate at vertex $u$. We argue that $|\mathcal{P}'| \geq |\mathcal{P}/2$, which by Equation (17) yields $|\mathcal{P}'| \geq \frac{dm_{\mathsf{scl}}}{8}$. In fact, for any $u \in W$, any color $x \in \mathsf{miss}_\chi(u)$ and any other color $y$, let $v_1, v_2, \ldots, v_{k\geq 2}$ be all uncolored neighbors of $u$ such that $(v_i,u) \in S$ and $\mathsf{clr}_\chi(v_i,u) = y$. Then, there exists at most one index $1 \leq i \leq k$, such that the maximal $\{x,y\}$-alternating path $P_x(v_i,u)$ starting at vertex $v_i$ terminates at $u$ (otherwise $u$ would be incident on more than one edge colored $y$). Therefore, all other maximal $\{x,y\}$-alternating paths $P_x(v_j,u), j \neq i$ do not end at $u$. It follows that $|\mathcal{P}'| \geq |\mathcal{P}/2$.

Lemma 9.4 implies that the total length of all maximal alternating paths whose lengths are at least 2 is at most $3(\Delta+1)m$. Therefore, the average length of paths in $\mathcal{P}'$ is at most

$$\frac{24(\Delta+1)m}{dm_{\mathsf{scl}}} \leq \frac{128(\Delta+1)m}{d\lambda} < 0.7L,$$

from which we conclude that the expected length of a path, conditioned on the event that $(v,u) \in \mathcal{P}'$, is bounded by $0.7L$. Using Markov's inequality, the edge $(v,u)$ gets colored with probability $\geq 0.3$, conditioned on the event that $(v,u) \in \mathcal{P}'$. Therefore, the overall success probability of Step (2) is at least $0.3 \cdot \frac{1}{8} = \frac{3}{80}$. $\square$

**Lemma 9.10.** *If $m_{\mathsf{ind}} \geq |S|/4$, then with probability $\geq 0.15$, the edge $(v,u)$ gets colored in Step (3), and thus in this case the potential $\Phi$ drops by at least 2.*

*Proof.* For each independent edge $e = (v,u)$ and color $x \in \mathsf{miss}_\chi(u)$, define $Q_x(e)$ to be the maximal $\{x,z\}$-alternating path starting at vertex $u$ if we try to perform Vizing's procedure to extend $\chi$ to $e$ around the colored neighborhood $N_\chi(u)$, where vertex $v$ uses the missing color $\mathsf{clr}_\chi(v,u)$, vertices $w$ use missing colors $c_\chi(w), \forall w \in N_\chi(u)$, and $u$ uses the missing color $x \in \mathsf{miss}_\chi(u)$.

We argue that for any fixed $u \in W$ and $x \in \mathsf{miss}_\chi(u)$, all the maximal alternating paths $Q_x(\cdot, u)$ that correspond to different independent edges are different. Indeed, for any pair of independent edges $(v_1,u),(v_2,u)$, denoting by $(u,w_1)$ and $(u,w_2)$ the first edges of maximal alternating paths $Q_x(v_1,u)$ and $Q_x(v_2,u)$, respectively, then by definition of independent edges definition 9.3, $w_1, w_2$ should belong to different weakly connected components of the digraph $F_\chi(u)$, and therefore $w_1 \neq w_2$.

Since $|\mathsf{miss}_\chi(u)| \geq d/2$ for each $u \in W$, the collection $\mathcal{Q}$ of all different paths $Q_x(e)$, over all $m_{\mathsf{ind}}$ independent edges $e = (v,u), u \in W$ and over all $x \in \mathsf{miss}_\chi(u)$, satisfies:

$$|\mathcal{Q}| \;\geq\; \frac{1}{2} \cdot \frac{d}{2} \cdot m_{\mathsf{ind}} \;=\; \frac{d m_{\mathsf{ind}}}{4},$$

where the extra factor of $1/2$ stems from the fact that two different independent edges $e, e'$ could be the endpoints edges of the same alternating path.

Lemma 9.4 implies that the total length of paths in $\mathcal{Q}$ whose lengths are at least 2 is at most $3(\Delta+1)m$. Therefore, the average length of paths in $\mathcal{Q}$ is at most

$$\frac{12(\Delta+1)m}{d m_{\mathsf{ind}}} \;\leq\; \frac{64(\Delta+1)m}{d\lambda} \;<\; 0.4L,$$

from which we conclude that the expected length of a path, conditioned on event that edge $(v,u)$ is independent, is bounded by $0.4L$. Using Markov's inequality, the edge $(v,u)$ gets colored with probability $\geq 0.6$, conditioned on the event that edge $(v,u)$ is independent. Therefore, the overall success probability of Step (2) is at least 0.15. $\square$

**Lemma 9.11.** *If $m_{\mathsf{lon}} \geq |S|/4$, then with probability $\geq 0.25$, the potential function $\Phi$ drops by 2 after Step (4).*

*Proof.* Since $m_{\mathsf{lon}} \geq |S|/4$, with probability at least 0.25, the random edge $(v,u)$ would be lonely. In this case, we would execute Step (4) which turns two lonely edges $(v,u),(v',u)$ into social ones. Thus, $\Phi$ decreases by 2. $\square$

## 9.4 Proof of Lemma 6.1

By Lemma 9.8, Lemma 9.9, Lemma 9.10, and Lemma 9.11, each iteration of the random color extension procedure decreases $\Phi$ by at least 2 with constant probability. Since the initial potential value of $\Phi$ is set as $10|S| + m_{\mathsf{lon}} \leq 11\lambda$, it follows that the algorithm terminates after $O(\lambda \log n)$ iterations with high probability. By Lemma 9.5, each iteration takes time $\tilde{O}(\Delta + L)$, hence the total running time is bounded by $\tilde{O}(\lambda \log n(\Delta + L)) = \tilde{O}\left(\Delta\lambda + \frac{\Delta m}{d}\right)$ with high probability.

As for the total number of newly colored edges, $|S|$ decreases either when a new edge in $S$ gets colored or when a vertex $u \in U^\star$ is removed from $W$. In the latter case, when a vertex $u \in U^\star$ is removed from $W$, since $|\mathsf{miss}_\chi(u)|$ was at least $d$ at the beginning, there must have been at least $d/2$ newly colored edges incident on $u$ when $u$ has left $W$. Consequently, we can charge the loss of at most $d/2$ units to the size of $|S|$ due to the removal of $u$ from $S$ to the at least $d/2$ newly colored edges incident on $u$; however, a newly colored edge may be charged by its two endpoints. In other words, if the size of $S$ has reduced by $\delta$ units, at least $\delta/3$ edges of $S$ have been colored. Finally, noting that our algorithm terminates whenever the size of $S$ drops below $\frac{3\lambda}{4}$, it follows that at this stage we must have colored at least $\frac{\lambda}{12}$ new edges, which concludes the proof of Lemma 6.1.

## References

[Alo03] Noga Alon. A simple algorithm for edge-coloring bipartite multigraphs. *Information Processing Letters*, 85(6):301–302, 2003.

[Arj82] Eshrat Arjomandi. An efficient algorithm for colouring the edges of a graph with $\Delta+1$ colours. *INFOR: Information Systems and Operational Research*, 20(2):82–101, 1982.

[Ass24] Sepehr Assadi. Faster vizing and near-vizing edge coloring algorithms. *CoRR*, abs/2405.13371, 2024.

[BBKO22] Alkida Balliu, Sebastian Brandt, Fabian Kuhn, and Dennis Olivetti. Distributed edge coloring in time polylogarithmic in $\Delta$. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing*, pages 15–25, 2022.

[BCC$^+$24] Sayan Bhattacharya, Din Carmon, Martín Costa, Shay Solomon, and Tianyi Zhang. Faster $(\Delta+1)$-Edge Coloring: Breaking the $m\sqrt{n}$ Time Barrier. In *65th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2024.

[BCHN18] Sayan Bhattacharya, Deeparnab Chakrabarty, Monika Henzinger, and Danupon Nanongkai. Dynamic algorithms for graph coloring. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1–20. SIAM, 2018.

[BCPS23] Sayan Bhattacharya, Martín Costa, Nadav Panski, and Shay Solomon. Density-sensitive algorithms for $(\Delta{+}1)$-edge coloring. *CoRR*, abs/2307.02415, 2023. To appear in ESA'24.

[BCPS24a] Sayan Bhattacharya, Martín Costa, Nadav Panski, and Shay Solomon. Arboricity-dependent algorithms for edge coloring. In *19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, volume 294 of *LIPIcs*, pages 12:1–12:15, 2024.

[BCPS24b] Sayan Bhattacharya, Martín Costa, Nadav Panski, and Shay Solomon. Nibbling at long cycles: Dynamic (and static) edge coloring in optimal time. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2024.

[BDH$^+$19] Soheil Behnezhad, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Marina Knittel, and Hamed Saleh. Streaming and massively parallel algorithms for edge coloring. In *27th Annual European Symposium on Algorithms (ESA)*, volume 144 of *LIPIcs*, pages 15:1–15:14, 2019.

[Ber22] Anton Bernshteyn. A fast distributed algorithm for $(\delta+1)$-edge-coloring. *J. Comb. Theory, Ser. B*, 152:319–352, 2022.

[BGW21] Sayan Bhattacharya, Fabrizio Grandoni, and David Wajc. Online edge coloring algorithms via the nibble method. In *Proceedings of theACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2830–2842. SIAM, 2021.

[BM17] Leonid Barenboim and Tzalik Maimon. Fully-dynamic graph algorithms with sublinear time inspired by distributed computing. In *International Conference on Computational Science (ICCS)*, volume 108 of *Procedia Computer Science*, pages 89–98. Elsevier, 2017.

[BS23] Soheil Behnezhad and Mohammad Saneian. Streaming edge coloring with asymptotically optimal colors. *arXiv preprint arXiv:2305.01714*, 2023.

[BSVW24] Joakim Blikstad, Ola Svensson, Radu Vintan, and David Wajc. Online edge coloring is (nearly) as easy as offline. In *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 2024.

[CH82] Richard Cole and John Hopcroft. On edge coloring bipartite graphs. *SIAM Journal on Computing*, 11(3):540–546, 1982.

[CHL$^+$20] Yi-Jun Chang, Qizheng He, Wenzheng Li, Seth Pettie, and Jara Uitto. Distributed edge coloring and a special case of the constructive lovász local lemma. *ACM Trans. Algorithms*, 16(1):8:1–8:51, 2020.

[Chr23] Aleksander Bjørn Grodt Christiansen. The power of multi-step vizing chains. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1013–1026. ACM, 2023.

[Chr24] Aleksander B. G. Christiansen. Deterministic dynamic edge-colouring. *CoRR*, abs/2402.13139, 2024.

[CK08] Richard Cole and Łukasz Kowalik. New linear-time algorithms for edge-coloring planar graphs. *Algorithmica*, 50(3):351–368, 2008.

[CMZ23] Shiri Chechik, Doron Mukhtar, and Tianyi Zhang. Streaming edge coloring with subquadratic palette size. *arXiv preprint arXiv:2305.07090*, 2023.

[CN90] Marek Chrobak and Takao Nishizeki. Improved edge-coloring algorithms for planar graphs. *Journal of Algorithms*, 11(1):102–116, 1990.

[COS01] Richard Cole, Kirstin Ost, and Stefan Schirra. Edge-coloring bipartite multigraphs in O(E log D) time. *Comb.*, 21(1):5–12, 2001.

[CPW19] Ilan Reuven Cohen, Binghui Peng, and David Wajc. Tight bounds for online edge coloring. In *60th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1–25. IEEE Computer Society, 2019.

[CRV24] Aleksander B. G. Christiansen, Eva Rotenberg, and Juliette Vlieghe. Sparsity-parameterised dynamic edge colouring. In *19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, volume 294 of *LIPIcs*, pages 20:1–20:18, 2024.

[CY89] Marek Chrobak and Moti Yung. Fast algorithms for edge-coloring planar graphs. *Journal of Algorithms*, 10(1):35–51, 1989.

[Dav23] Peter Davies. Improved distributed algorithms for the lovász local lemma and edge coloring. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4273–4295. SIAM, 2023.

[DGS24] Aditi Dudeja, Rashmika Goswami, and Michael Saks. Randomized greedy online edge coloring succeeds for dense and randomly-ordered graphs, 2024.

[DHZ19] Ran Duan, Haoqing He, and Tianyi Zhang. Dynamic edge coloring with improved approximation. In *30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2019.

[EK24] Michael Elkin and Ariel Khuzman. Deterministic simple $(1+\epsilon)\Delta$-edge-coloring in near-linear time. *CoRR*, abs/2401.10538, 2024.

[EPS14] Michael Elkin, Seth Pettie, and Hsin-Hao Su. $(2\Delta|1)$-Edge-Coloring is Much Easier than Maximal Matching in the Distributed Setting. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 355–370. SIAM, 2014.

[FGK17] Manuela Fischer, Mohsen Ghaffari, and Fabian Kuhn. Deterministic distributed edge-coloring via hypergraph maximal matching. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 180–191. IEEE, 2017.

[GKMU18] Mohsen Ghaffari, Fabian Kuhn, Yannic Maus, and Jara Uitto. Deterministic distributed edge-coloring with fewer colors. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 418–430, 2018.

[GNK$^+$85] Harold N Gabow, Takao Nishizeki, Oded Kariv, Daneil Leven, and Osamu Terada. Algorithms for edge coloring. *Technical Rport*, 1985.

[GP20] Jan Grebik and Oleg Pikhurko. Measurable versions of vizing's theorem. *Advances in Mathematics*, 374(107378), 2020.

[GS23] Prantar Ghosh and Manuel Stoeckl. Low-memory algorithms for online and w-streaming edge coloring. *arXiv preprint arXiv:2304.12285*, 2023.

[KLS$^+$22] Janardhan Kulkarni, Yang P. Liu, Ashwin Sah, Mehtaab Sawhney, and Jakub Tarnawski. Online edge coloring via tree recurrences and correlation decay. In *54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 104–116. ACM, 2022.

[Kow24] Lukasz Kowalik. Edge-coloring sparse graphs with $\Delta$ colors in quasilinear time. *CoRR*, abs/2401.13839, 2024. To appear in ESA'24.

[KS87] Howard J Karloff and David B Shmoys. Efficient parallel algorithms for edge coloring problems. *Journal of Algorithms*, 8(1):39–52, 1987.

[PR01] Alessandro Panconesi and Romeo Rizzi. Some simple distributed algorithms for sparse networks. *Distributed computing*, 14(2):97–100, 2001.

[Sin19] Corwin Sinnamon. Fast and simple edge-coloring algorithms. *arXiv preprint arXiv:1907.03201*, 2019.

[SV19] Hsin-Hao Su and Hoa T. Vu. Towards the locality of vizing's theorem. In Moses Charikar and Edith Cohen, editors, *51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2019.

[SW21] Amin Saberi and David Wajc. The greedy algorithm is not optimal for on-line edge coloring. In *48th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 198 of *LIPIcs*, pages 109:1–109:18, 2021.

[Viz64] V. G. Vizing. On an estimate of the chromatic class of a p-graph. *Discret Analiz*, 3:25–30, 1964.