To what extent is randomization necessary for $\Delta$-based graph coloring?

**Deterministic multi-pass coloring**

**Multi-pass streaming algorithm:**
- Input is a sequence of elements.
- In each pass, algorithm processes elements one by one.
- Algorithm has limited working space $\ll$ input size.

**$f(\Delta)$-graph coloring task:**
- Stream consists of the edges of a graph with $n$ vertices and maximum degree $\Delta$.
- Output a vertex coloring using $f(\Delta)$ colors.

**Immediate prior work**
- [ACS22]: Cannot color any $(\Delta+n)$-color a graph in 1 pass using $O(n)$ space with deterministic.
- [AK22]: $\Delta+1$ coloring in distributed models.  
- Many papers with 1-pass randomized algorithms [ACK19, A20, ACS22, HKNT22, AMK22] or distributed models [BKM20, Kuh20, HKNT22].

**Our results**
- A deterministic algorithm for $(\Delta+1)$-coloring using $O(\log \Delta \log \log \log n)$ space and $O(n \log n^2)$ space.
- Inspired by techniques of [GK21] and [ACS22].
- This generalizes to degree 1 list coloring, if the vertex color lists are provided in a certain way.

**Multi-pass streaming algorithm, details**
- Use $O(\log \Delta)$ epochs: in each epoch, fix colors for a constant fraction of the uncolored vertices.
- First epoch: randomly assigning colors in $[\Delta+1]$ to each vertex, gives $O(n)$ conflicting edges $\rightarrow$ have set of $(\Omega n)$ non-conflicting vertices.
- Derandomization: use method of conditional expectations to propose vertex colors with $O(n)$ conflicting edges.

**Partially committed coloring (PCC):**
- Constrains a color assignment.
- Say $U$ is set of vertices whose color was not fixed.
- If vertex $x \in U$, its color is $\chi(x)$; if $x \not\in U$, it has a subset of colors $P_x$, from a partition of $[\Delta+1]$.
- Bound on number of conflicting edges of PCC: $\Phi(P) = \sum_{x,y \in [\Delta+1]} |P_x \cap P_y|$.  
- Initially, $\Phi(P) = 1$; have a proposed color for $x$.  
- Once $|P_x| = 1$, have a proposed color for $x$.

**Refining PCC:** repeat to reduce $\sum_{x \in \Delta} |P_x|$.  
- Pass 1: Compute $s_v$ for every uncolored vertex $v$.  
- From current PCC $P$, construct family $F$ of $O(n^2)$ “refined” PCCs where $\sum_{Q \in F} \Phi(Q) \leq \Phi(P)$.
- Passes 2-3: Find a $Q \in F$ where $\Phi(Q) \leq \Phi(P)$.
- Set $P \leftarrow Q$, and repeat until all $|P_x| = 1$.

**Adversarially robust graph coloring**

**Streaming algorithm:**
- Input is a sequence of elements $e_1, \ldots, e_m$.
- “tracking” algorithm: emits output $o_i$ after every input $e_i$, and is wrong if any output is.

**Static setting:**
- For all valid input streams, $P_i$ [output wrong] $\leq \delta$.

**Adversarial setting [BJW20]:**
- An adversary adaptively produces an input sequence $e_1, \ldots, e_m$, where it chooses input $e_i$ based on the algorithm outputs $o_1, \ldots, o_{i-1}$ so far.
- “Adversarially robust”: for all adversaries making valid input streams, $P_i$ [output wrong] $\leq \delta$.

**$f(\Delta)$-graph coloring task:**
- Receive sequence of edges in a simple graph on $n$ vertices of max degree $\Delta$.
- Output a vertex coloring of the edges so far using $f(\Delta)$ colors.

**Immediate prior work**
- [CGS22]: adversarial robust algorithm for $f(\Delta)$-coloring using $O(n)$ space, $O(\Delta n)$ random bits.
- [CGS22]: adversarial robust $O(\Delta^2)$ coloring algorithms need $\Omega(n)$ space.
- Many papers for static setting [BG18, ACK19, AA20, ACS22, HKNT22].

**Our results**
- An adversarially robust, $O(\Delta^2)$-coloring algorithm using $O(n)$ space and oracle access to $O(n \Delta)$ random bits.
- An adversarially robust, $O(\Delta^3)$-coloring algorithm using $O(n)$ space.

An $O(\Delta^3)$ color $\tilde{O}(n)$ space algorithm

- Divide stream into “epochs” of $n$ edges each.
- For each epoch $i$, present output of independent sub-sketch.
- Pick random $h_i : [n] \rightarrow [\Delta^3]$.
- Before epoch $i$, record edges $\{x, y\}$ with $h_i(x) = h_i(y)$ into set $D_i$.
- During epoch $i$, record edges $\{x, y\}$ with $h_i(x) = h_i(y)$ into set $B_i$.
- Compute $(\Delta+1)$-coloring $\chi$ of edge set $D_i \cup B_i$.
- For each vertex $x$, output $\chi(x), h_i(x) \in [\Delta+1] \times [\Delta^3]$.
- Discard sets when no longer needed.
- $|D_i| = O(n \Delta)$ w.h.p. because adversary doesn’t see $h_i$ until epoch 1; and $|B_i| \leq n$.

$O(\Delta^{5/2})$ coloring algorithm with $\tilde{O}(n)$ space algorithm

- Different handling for fast vertices (those which receive $\geq \sqrt{n}$ edges per epoch of $n$ edges).
- Remaining slow vertices form bounded degree graphs $D_i \cup B_i \Rightarrow$ only $O(\Delta^{5/2})$ colors used.

**Fast vertices:**
- Partition vertices by degree into $\sqrt{n}$ levels $[1, \sqrt{n}, \sqrt{n}+1, 2\sqrt{n}, \ldots, \Delta - \sqrt{n} + 1, \Delta]$.
- Color each level like an epoch of the example alg.
- Edges may cross levels, and levels coexist, unlike epochs.
- Have $\sqrt{n}$ levels using $\Delta^{5/2} \times O(\sqrt{n})$ colors each.