Palette Sparsification and Beyond

Maxime Flin
Aalto University

AMG 2026

This Talk

This Talk

1. Palette Sparsification trust me, it's cool

This Talk

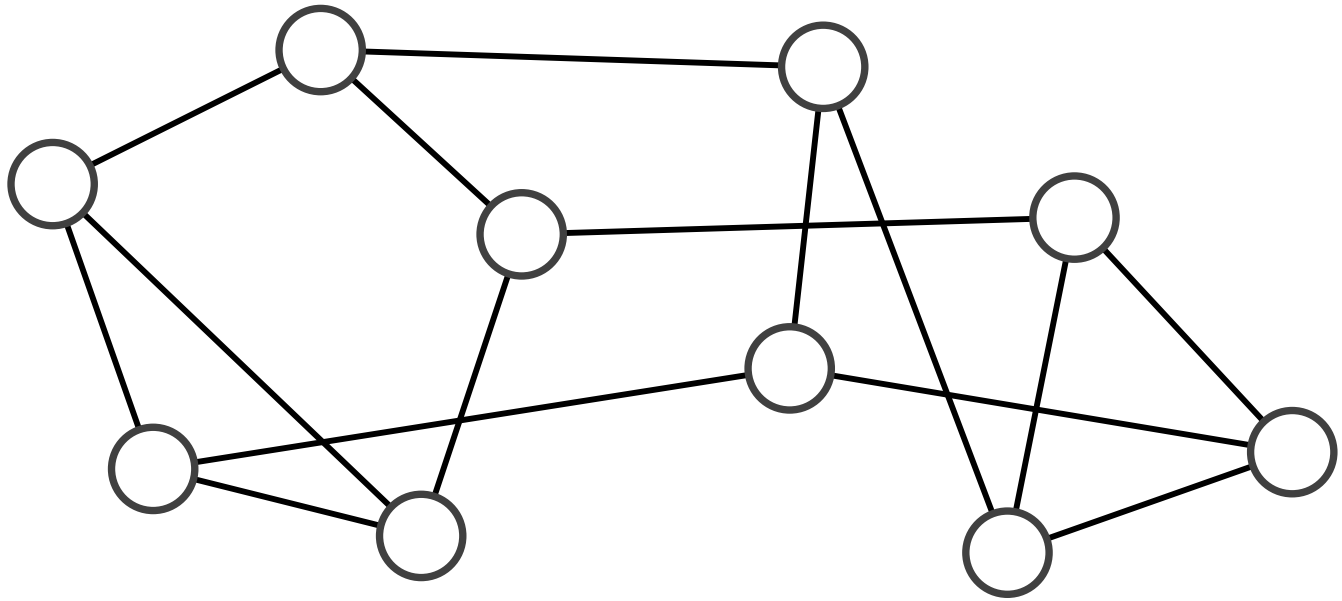
1. Palette Sparsification — trust me, it's cool
2. Sparse Recovery — trust me, it's useful

This Talk

1. Palette Sparsification — trust me, it's cool
2. Sparse Recovery — trust me, it's useful
3. Meta-Algorithm for Coloring — trust me

Palette Sparsification Theorem

Assadi, Chen, Khanna (SODA, 2019)

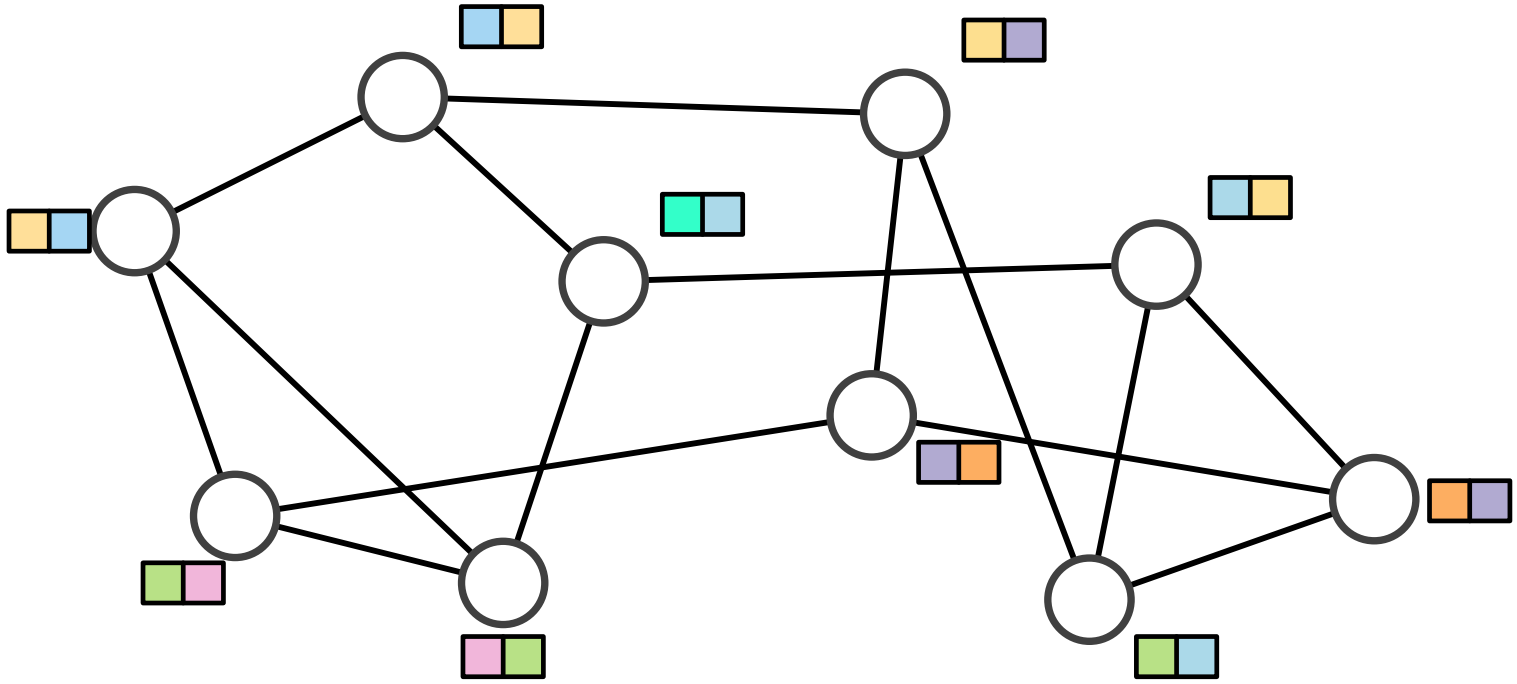


Palette Sparsification Theorem.
Independently sample $O(\log n)$ random colors in $[\Delta + 1]$ for every vertex; whp, there exists a coloring where each vertex uses one of its random colors.

$\Delta =$ the maximum degree

Palette Sparsification Theorem

Assadi, Chen, Khanna (SODA, 2019)

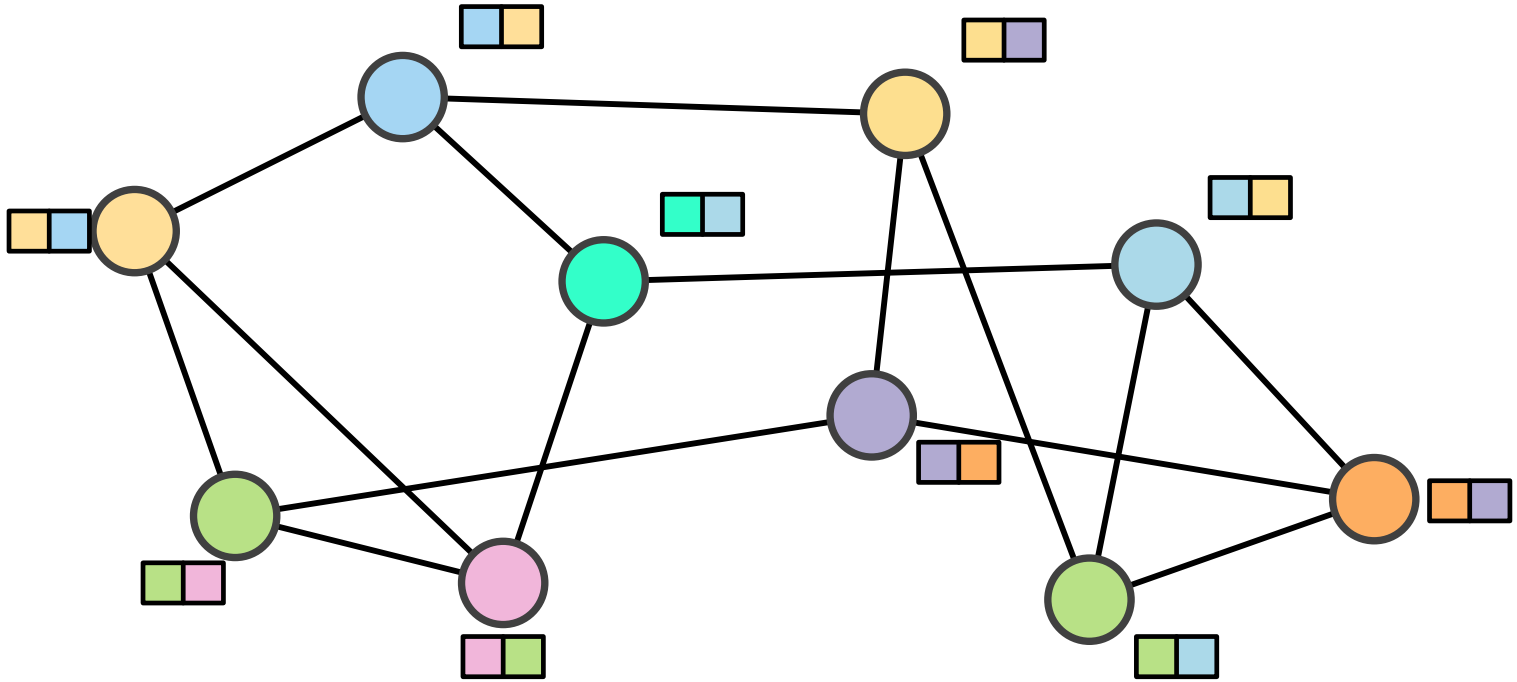


Palette Sparsification Theorem.
Independently sample $O(\log n)$ random colors in $[\Delta + 1]$ for every vertex; whp, there exists a coloring where each vertex uses one of its random colors.

$\Delta =$ the maximum degree

Palette Sparsification Theorem

Assadi, Chen, Khanna (SODA, 2019)

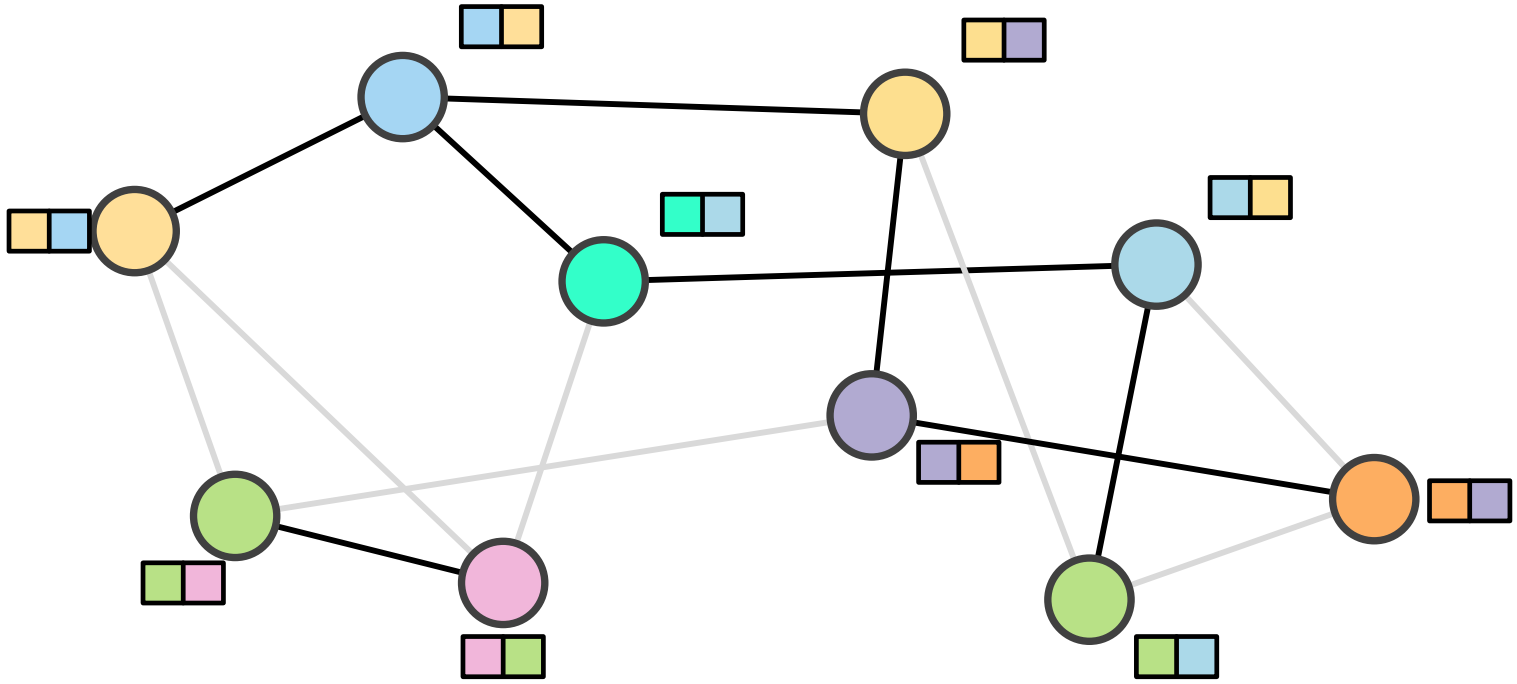


Palette Sparsification Theorem.
Independently sample $O(\log n)$ random colors in $[\Delta + 1]$ for every vertex; whp, there exists a coloring where each vertex uses one of its random colors.

$\Delta =$ the maximum degree

Palette Sparsification Theorem

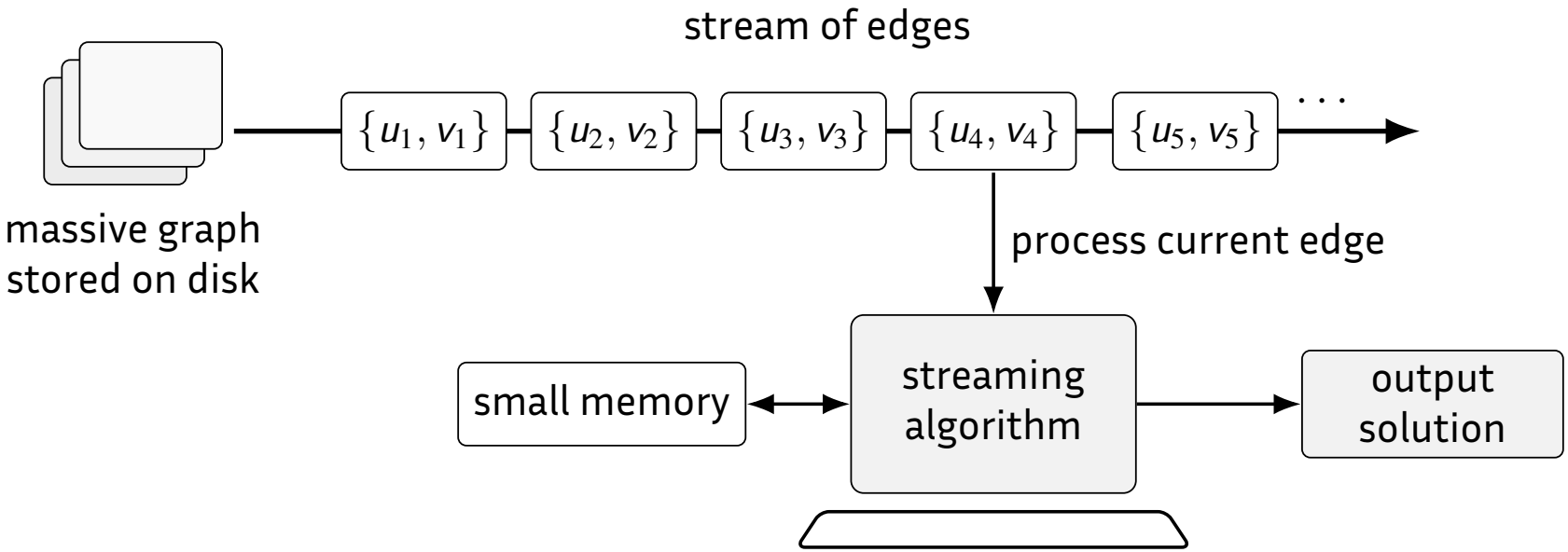
Assadi, Chen, Khanna (SODA, 2019)



Palette Sparsification Theorem.
Independently sample $O(\log n)$ random colors in $[\Delta + 1]$ for every vertex; whp, there exists a coloring where each vertex uses one of its random colors.

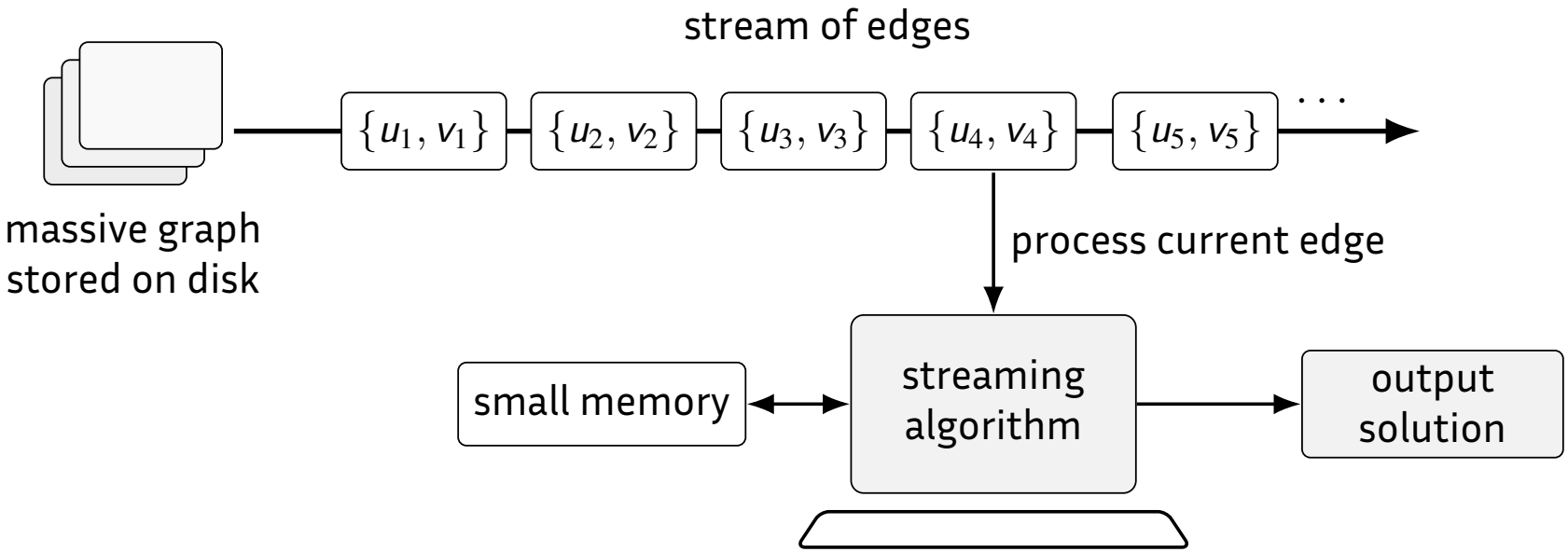
$\Delta =$ the maximum degree

Semi-Streaming Model



$n =$ number of nodes

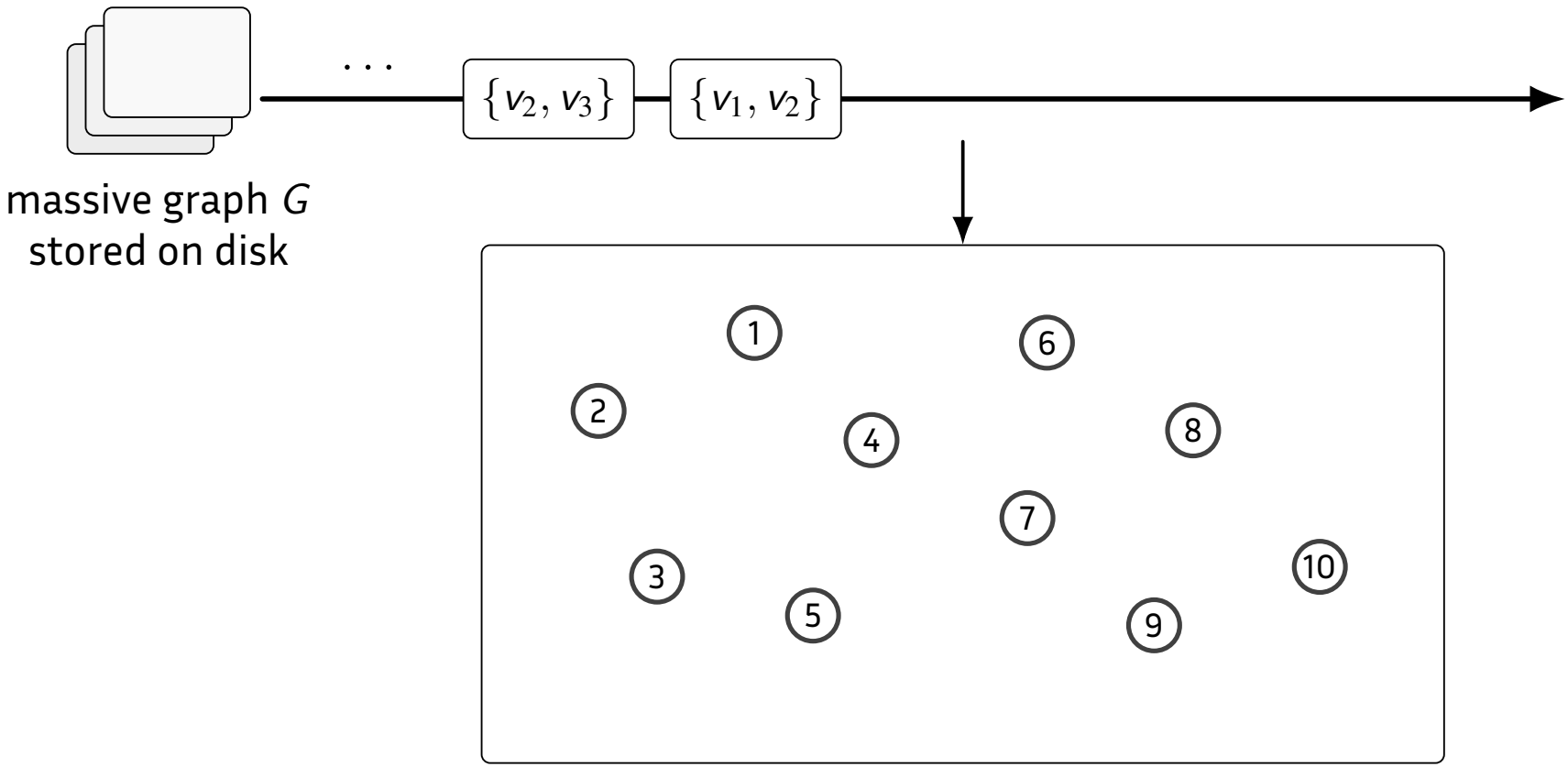
Semi-Streaming Model



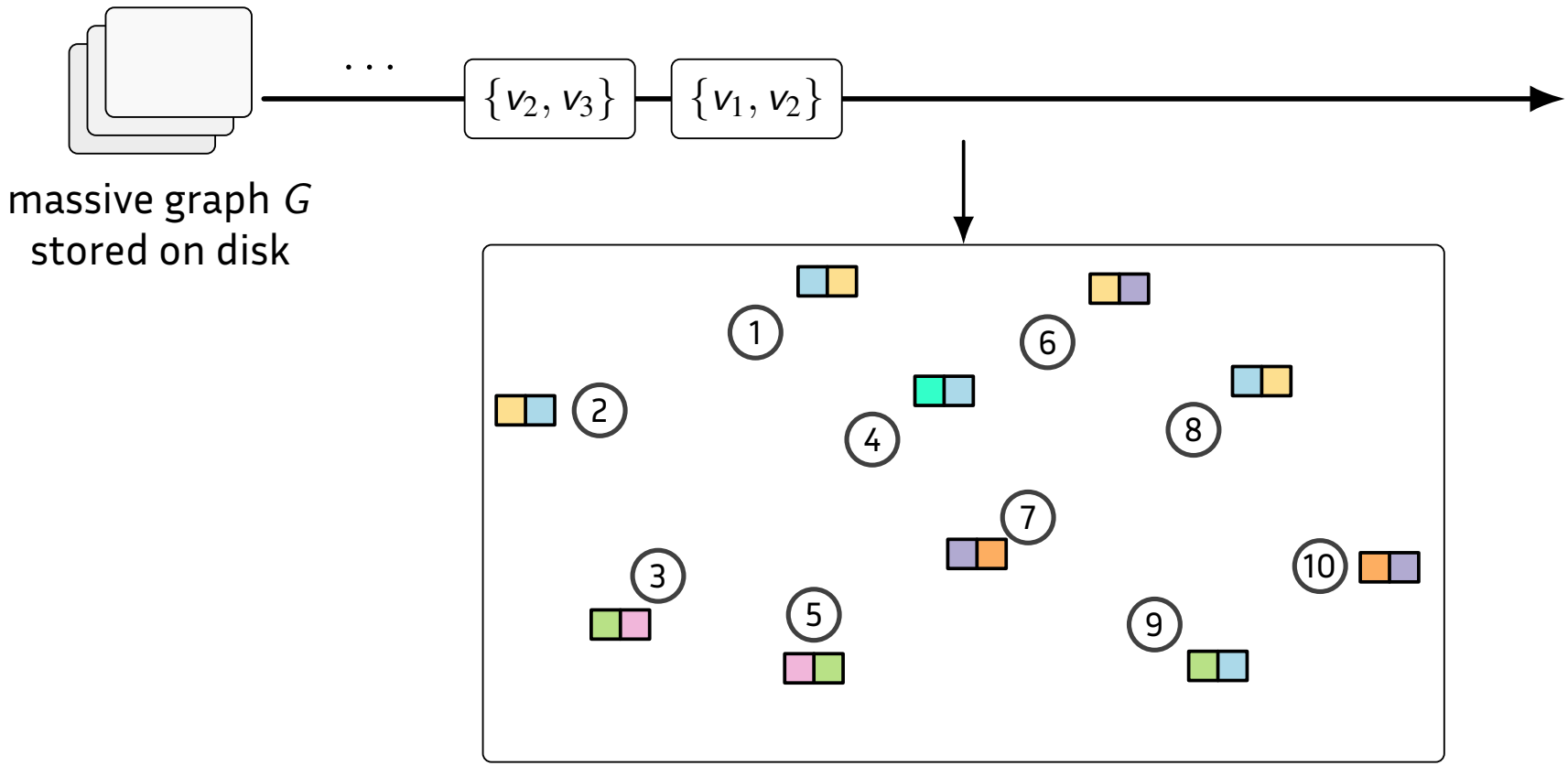
Ideally **one-pass** algorithm using $n \cdot \text{poly}(\log n)$ **space**

$n =$ number of nodes

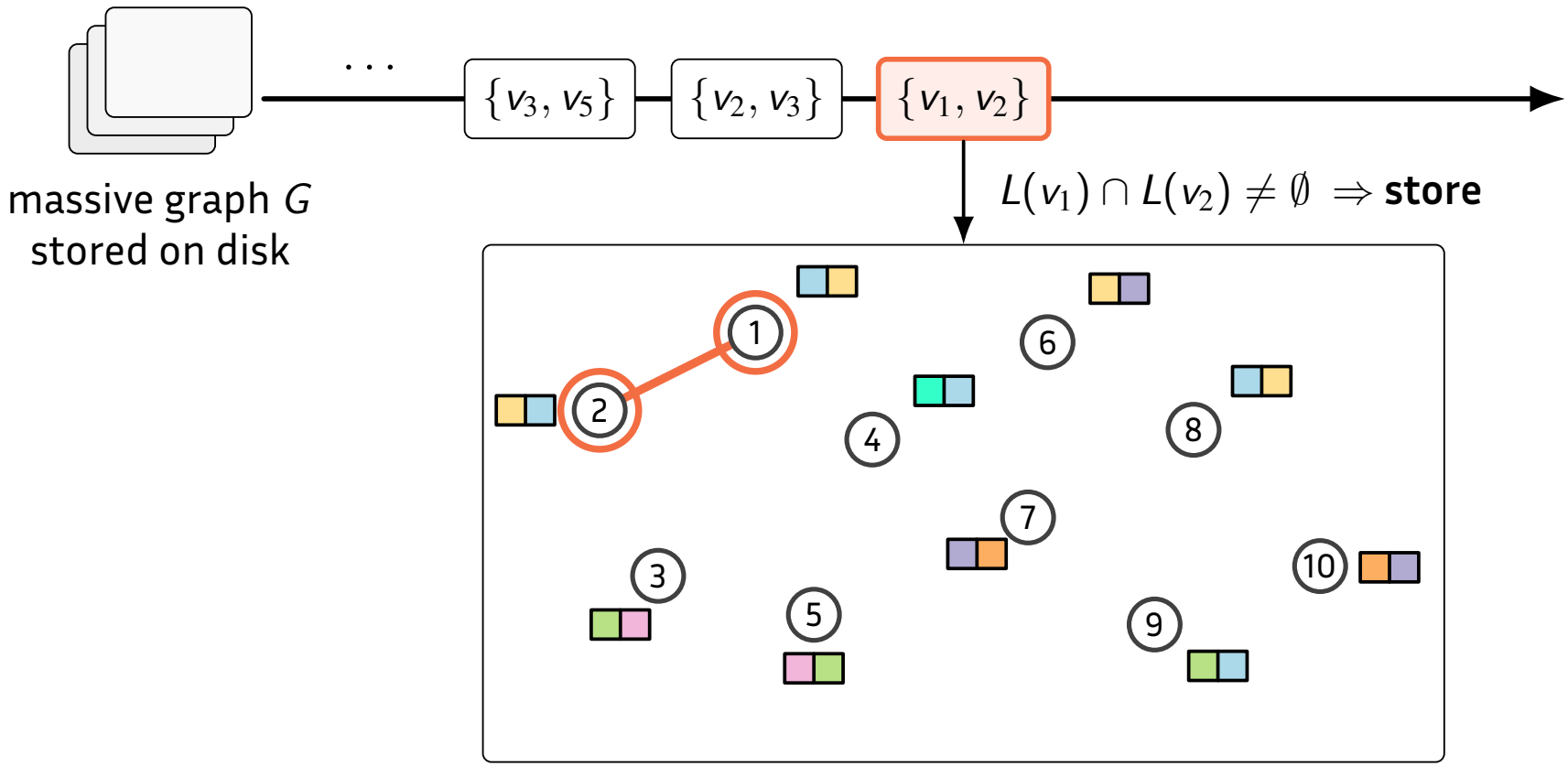
Semi-Streaming $(\Delta + 1)$-Coloring



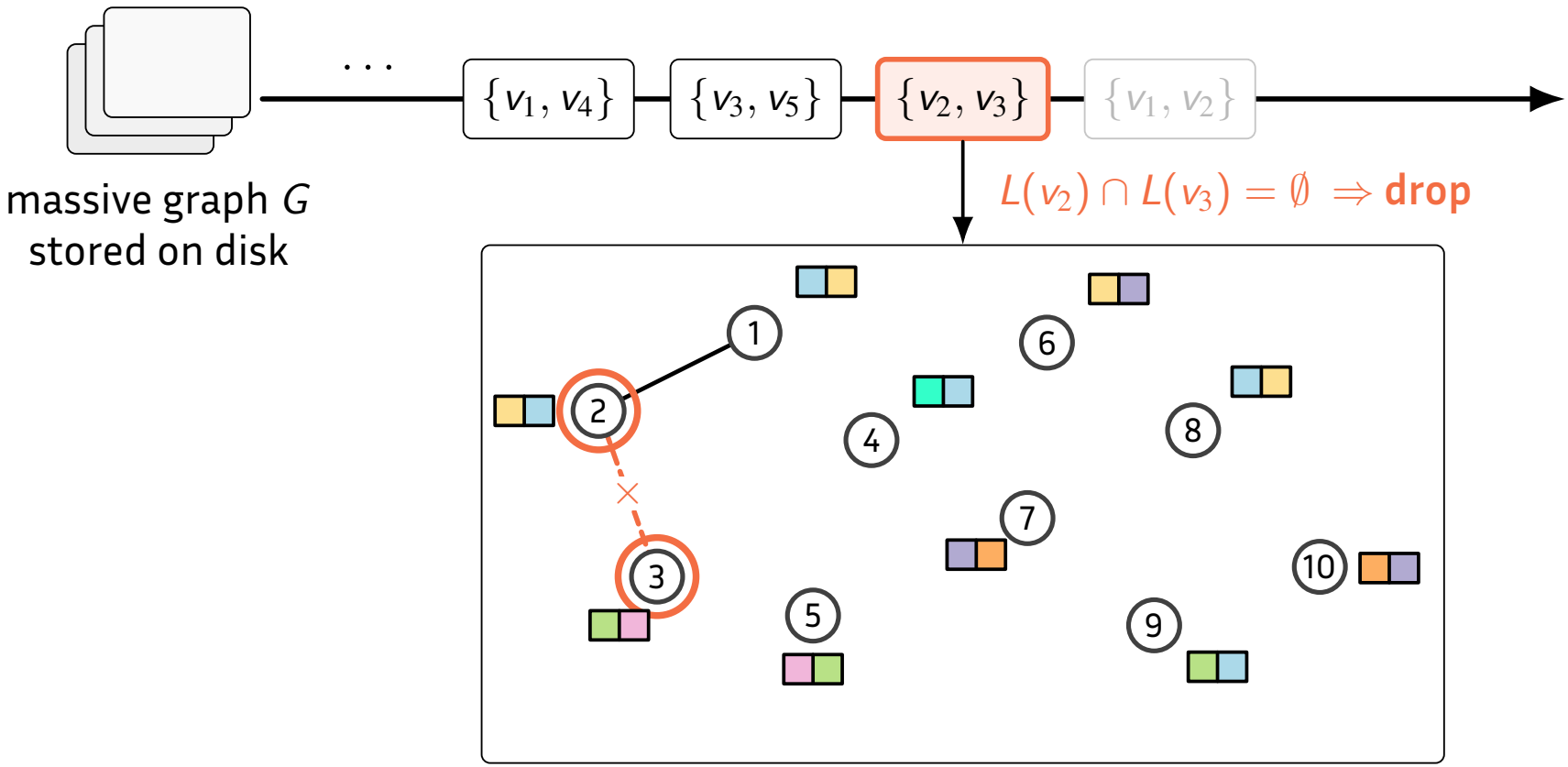
Semi-Streaming $(\Delta + 1)$-Coloring



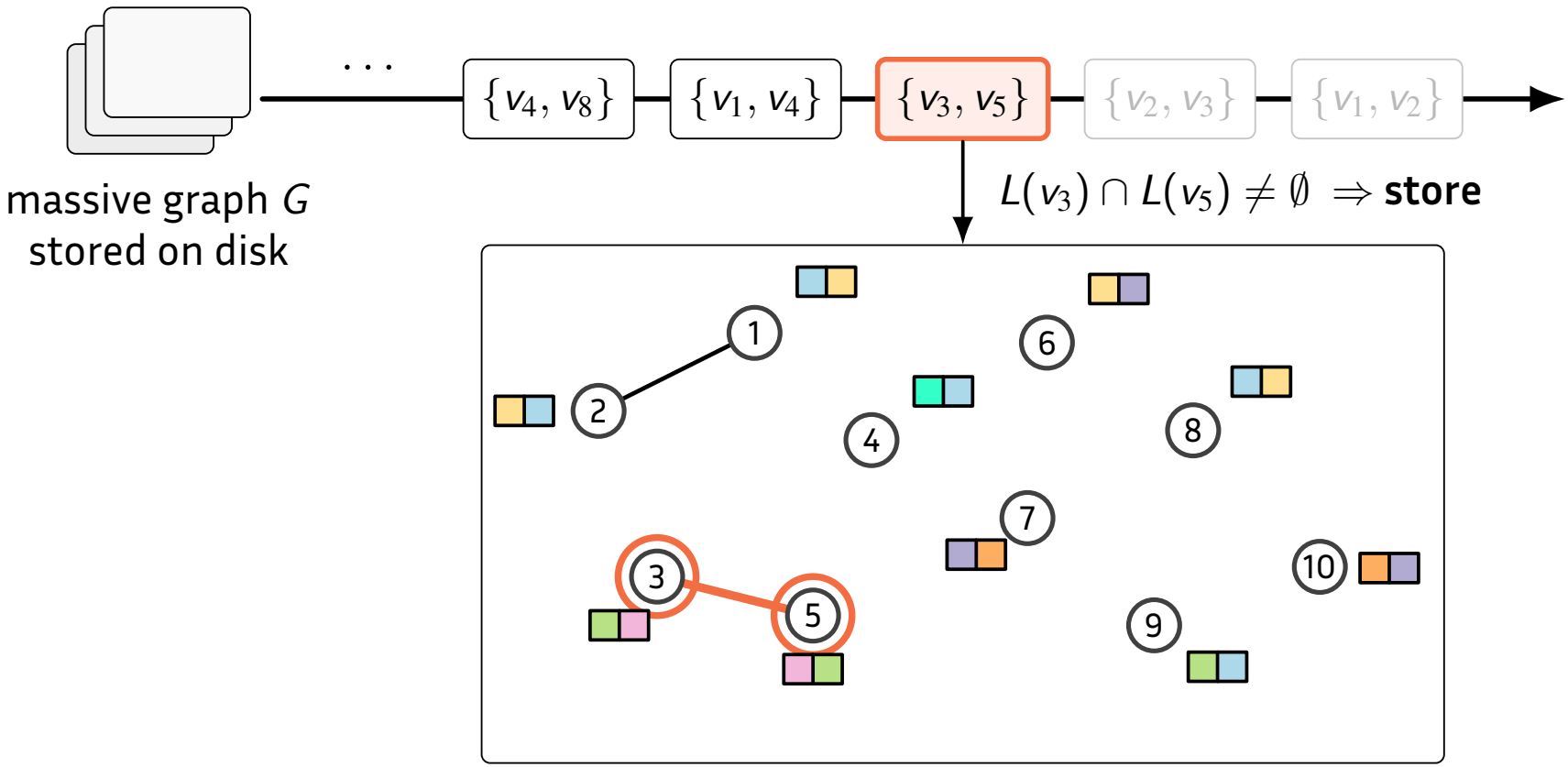
Semi-Streaming $(\Delta + 1)$-Coloring



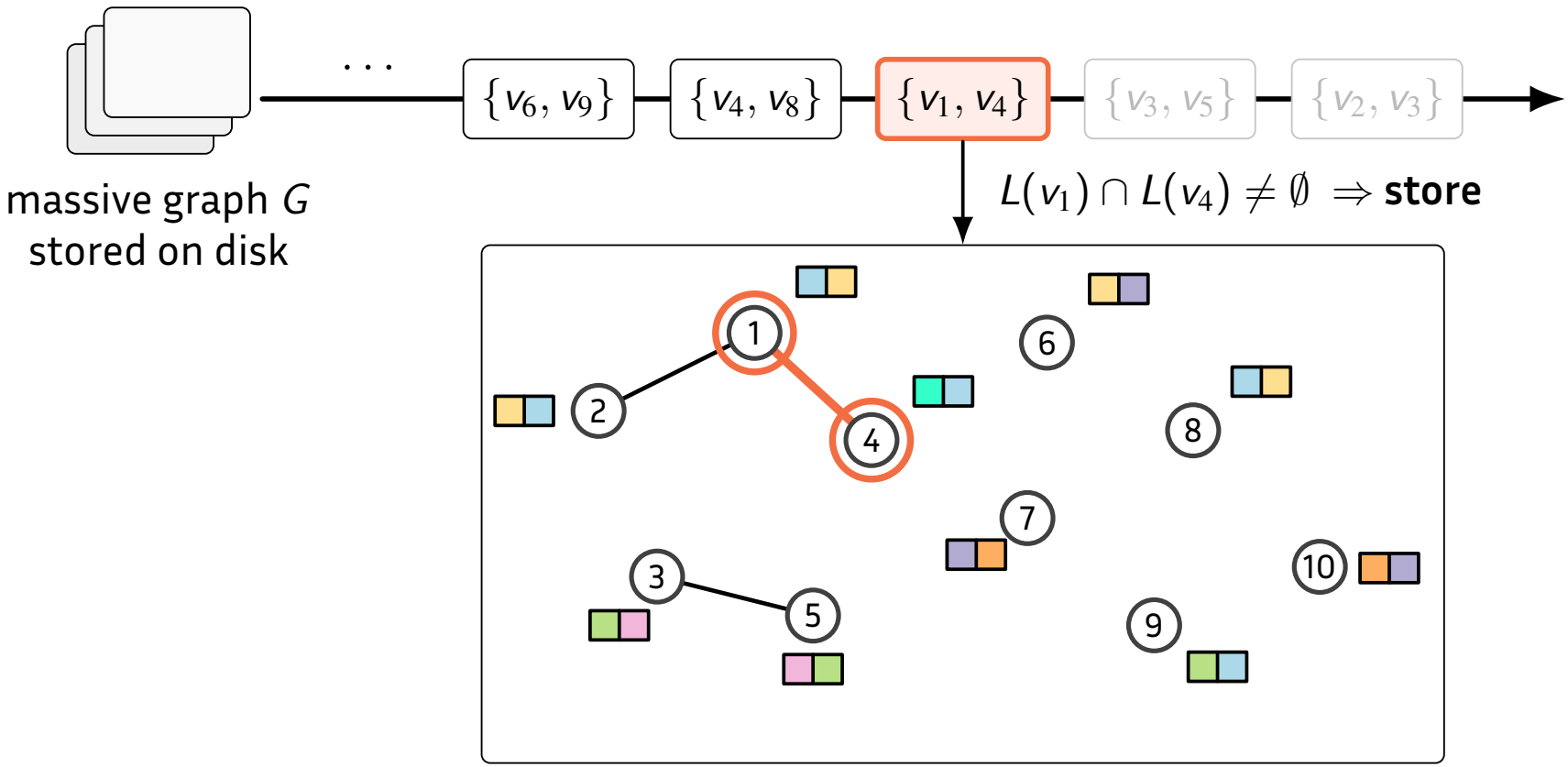
Semi-Streaming $(\Delta + 1)$-Coloring



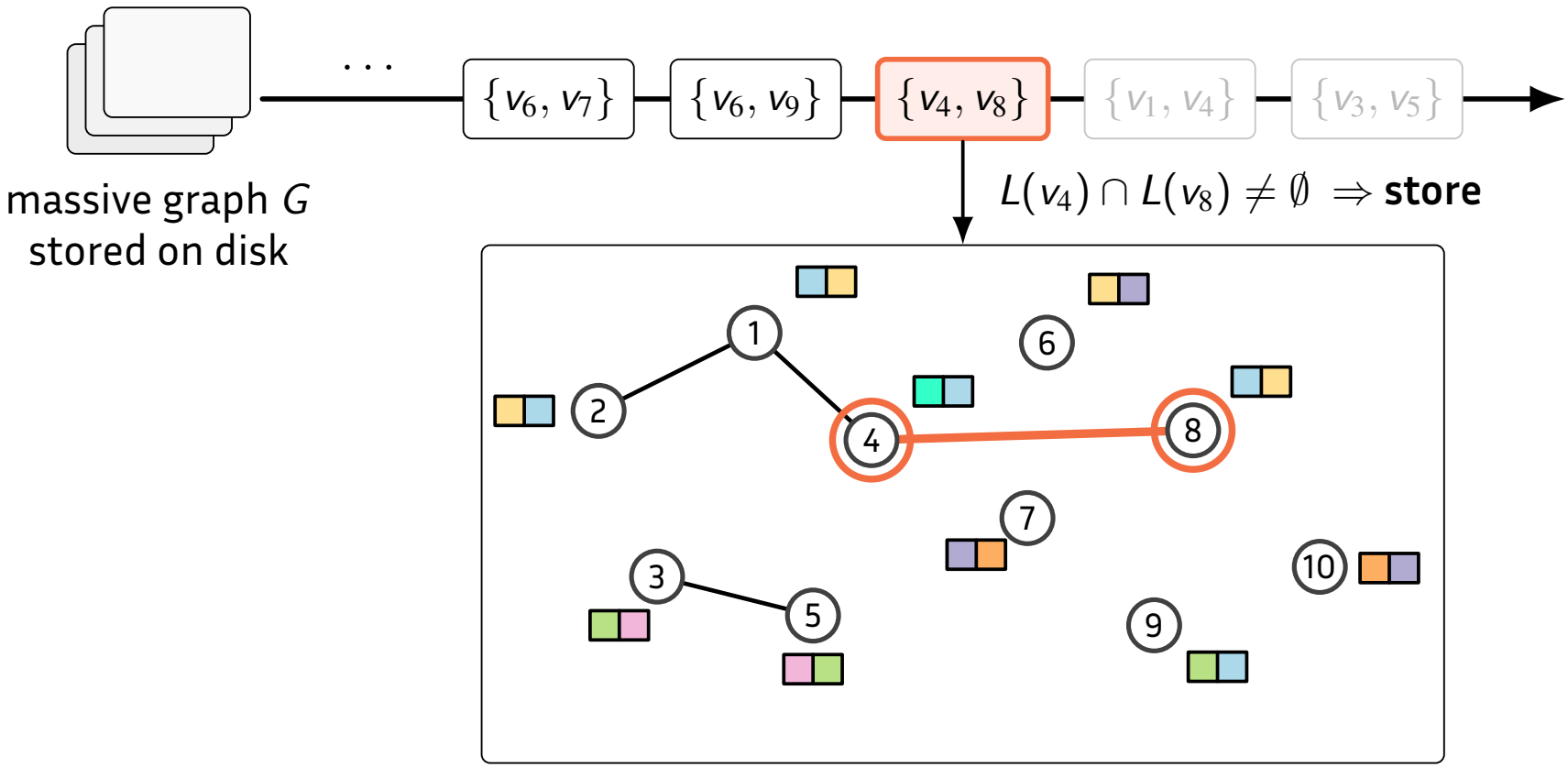
Semi-Streaming $(\Delta + 1)$-Coloring



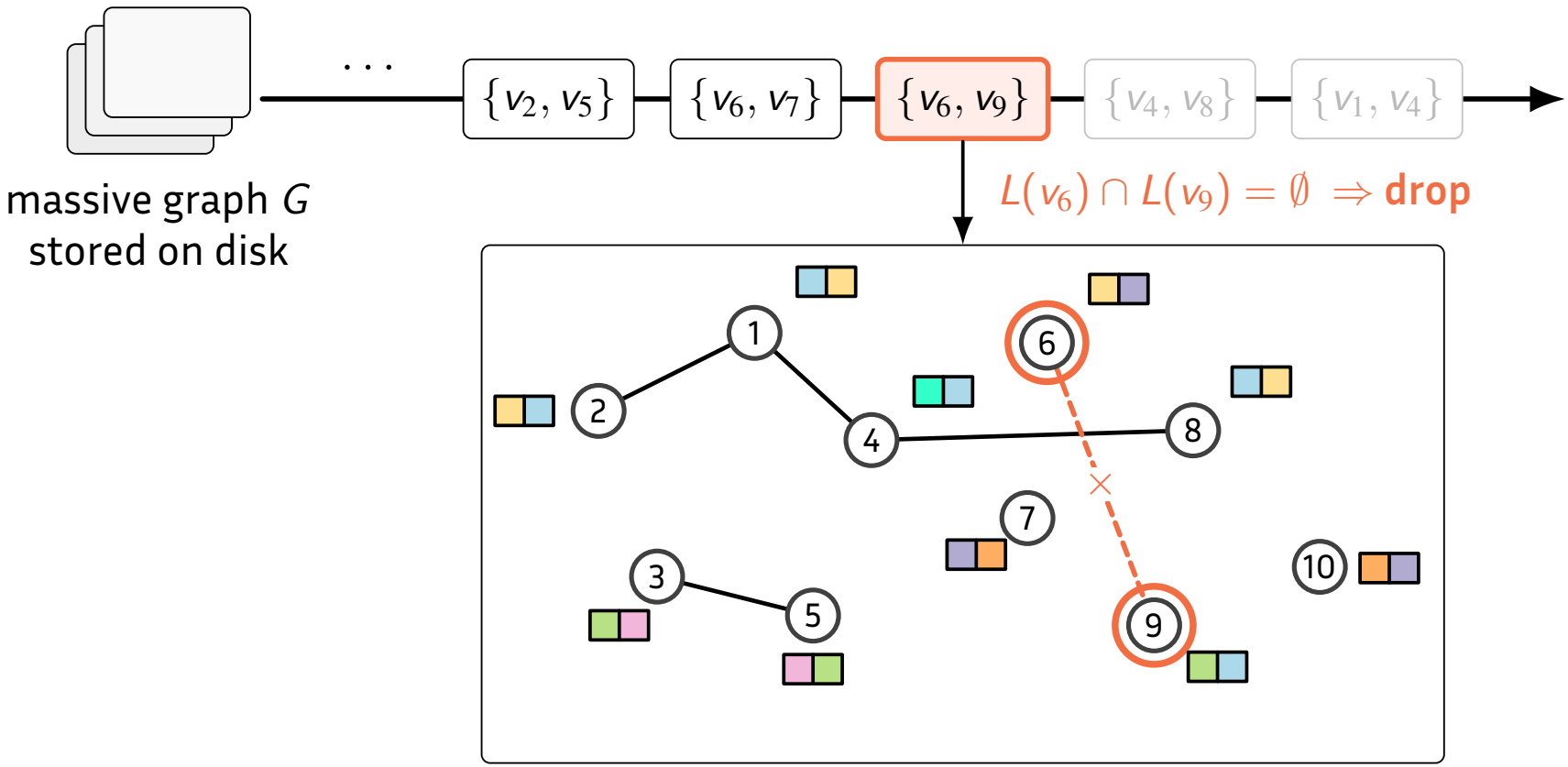
Semi-Streaming $(\Delta + 1)$-Coloring



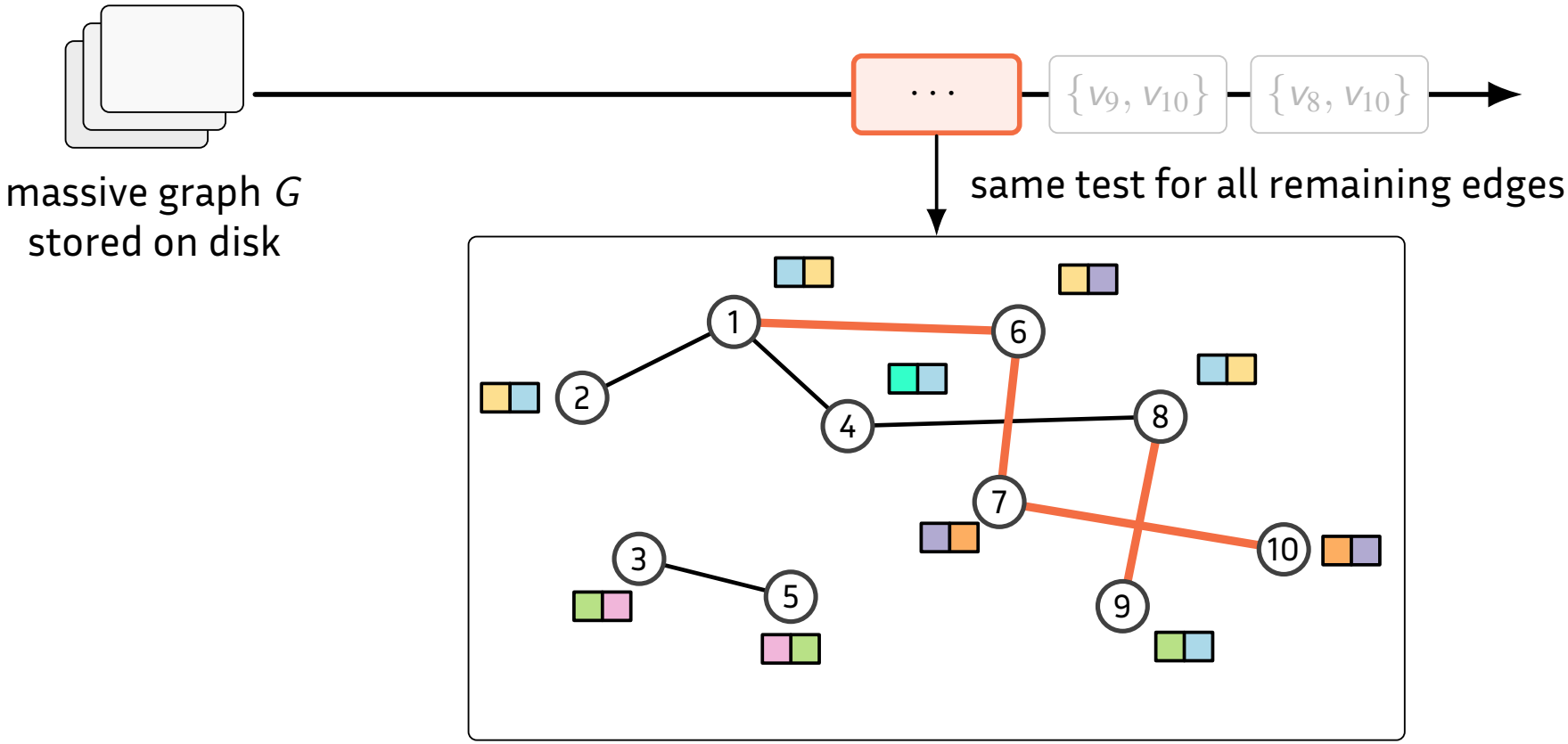
Semi-Streaming $(\Delta + 1)$-Coloring



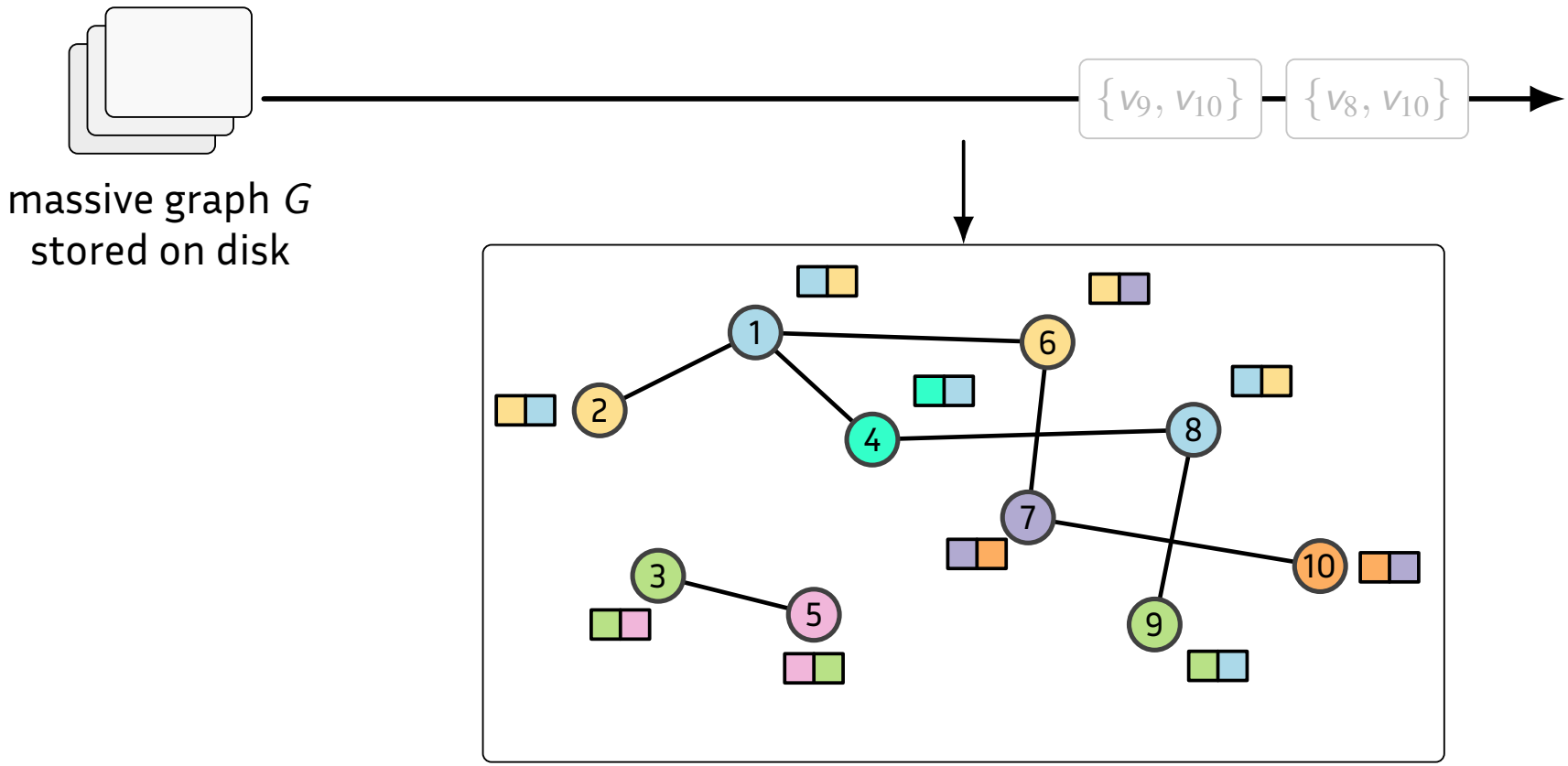
Semi-Streaming $(\Delta + 1)$-Coloring



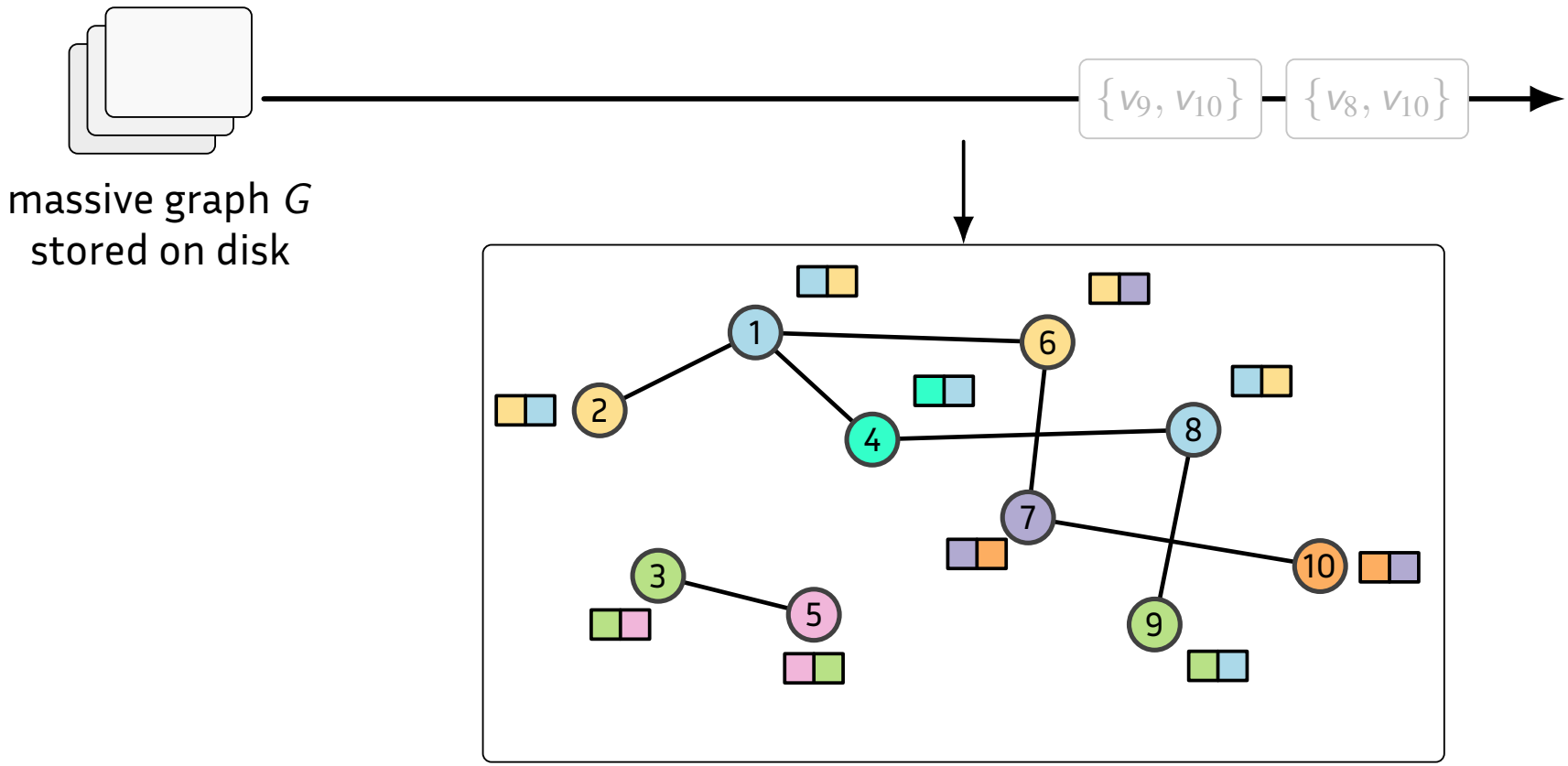
Semi-Streaming $(\Delta + 1)$-Coloring



Semi-Streaming $(\Delta + 1)$-Coloring



Semi-Streaming $(\Delta + 1)$-Coloring



Also works in other models: query, CONGESTED-CLIQUE, MPC with $\widetilde{O}(n)$ space

This Talk

1. Palette Sparsification

   why does it work?

2. Coloring in Semi-Streaming with Fewer Colors

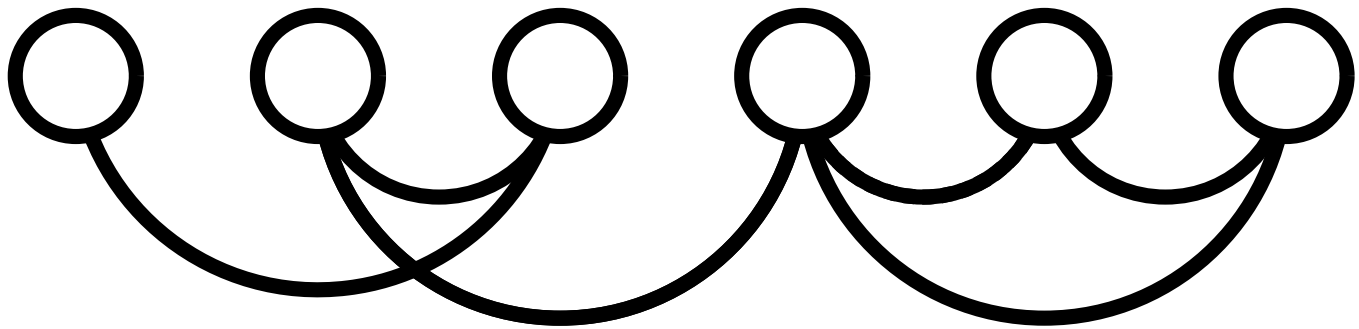
   need more tools

3. Meta-Algorithm

   connections between distributed and sub-linear alg.

Chapitre 1: Palette Sparsification

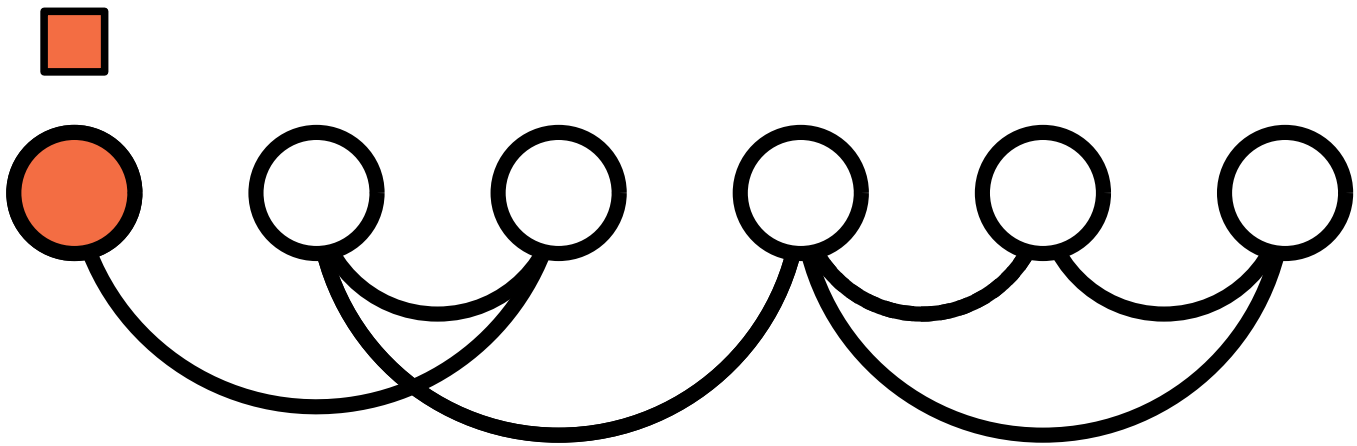
Warm-up: 2Δ-Coloring



$L(v) =$ list of colors sampled by this process

Claim. Whp $|L(v)| \leq O(\log n)$ for all v

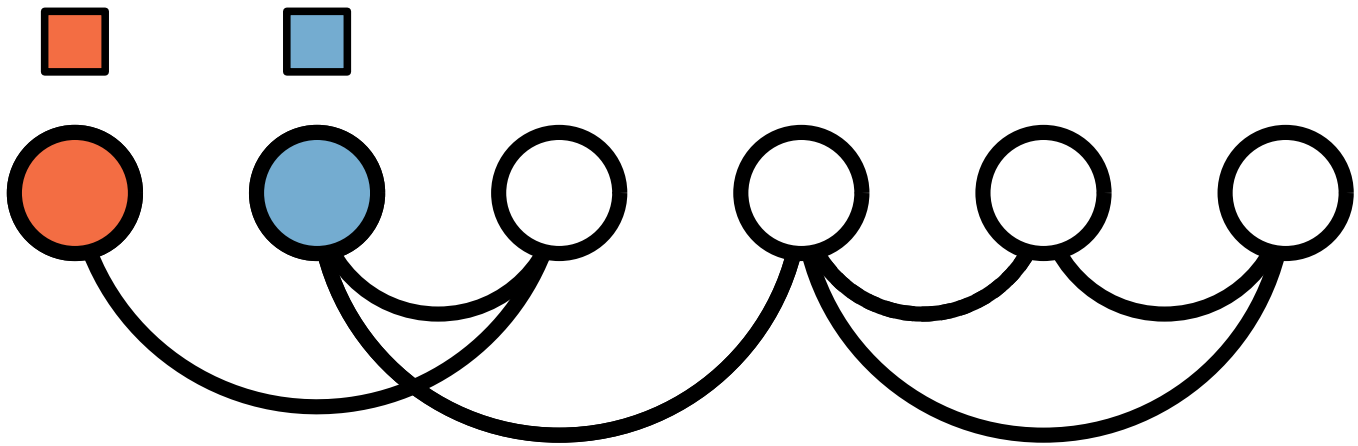
Warm-up: 2Δ-Coloring



$L(v) =$ list of colors sampled by this process

Claim. Whp $|L(v)| \leq O(\log n)$ for all v

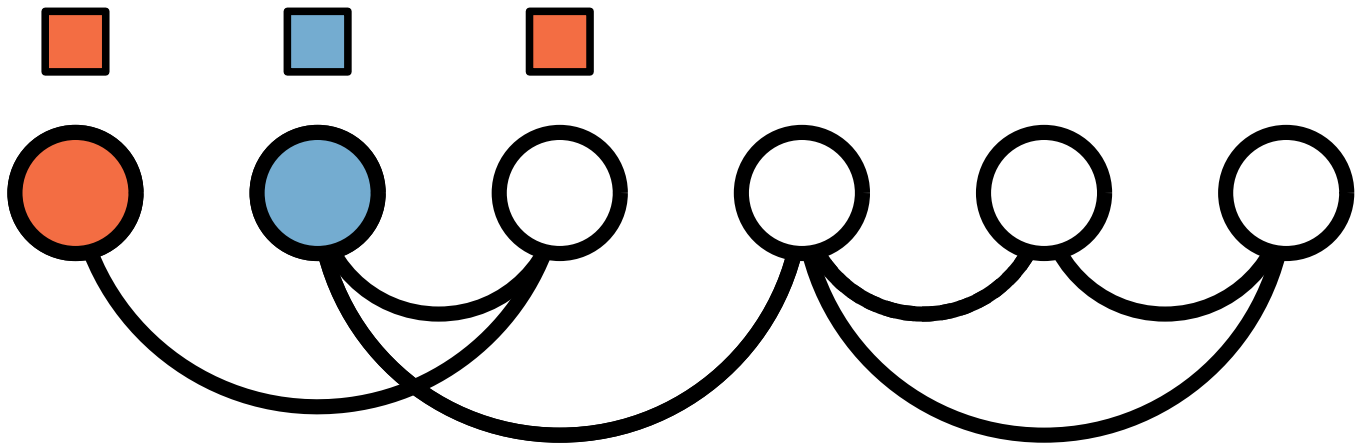
Warm-up: 2Δ-Coloring



$L(v) =$ list of colors sampled by this process

Claim. Whp $|L(v)| \leq O(\log n)$ for all v

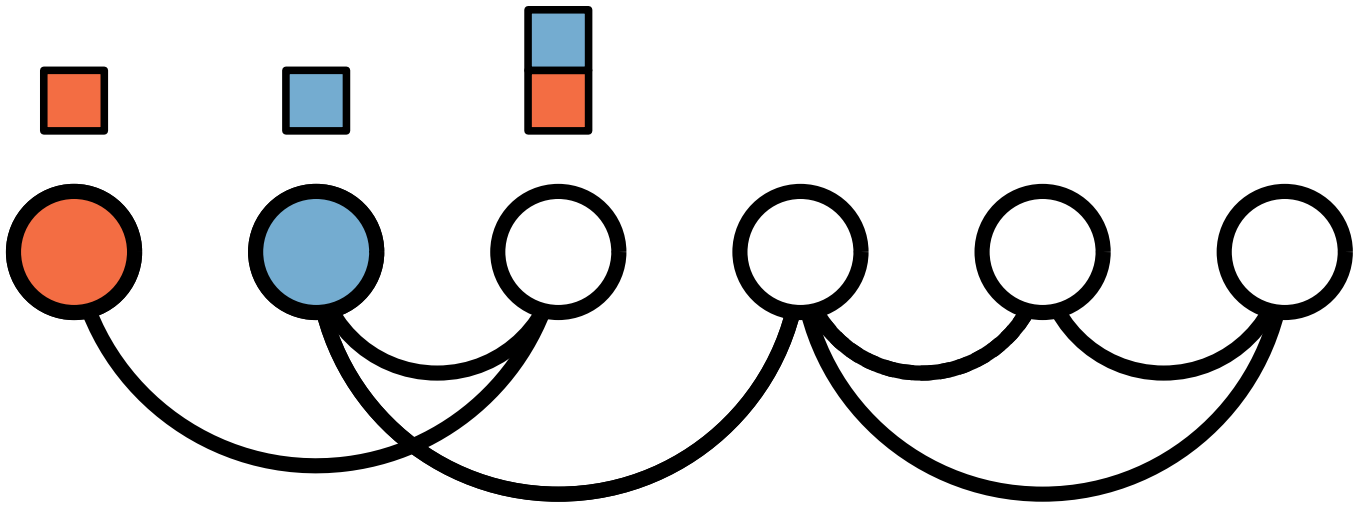
Warm-up: 2Δ-Coloring



$L(v) =$ list of colors sampled by this process

Claim. Whp $|L(v)| \leq O(\log n)$ for all v

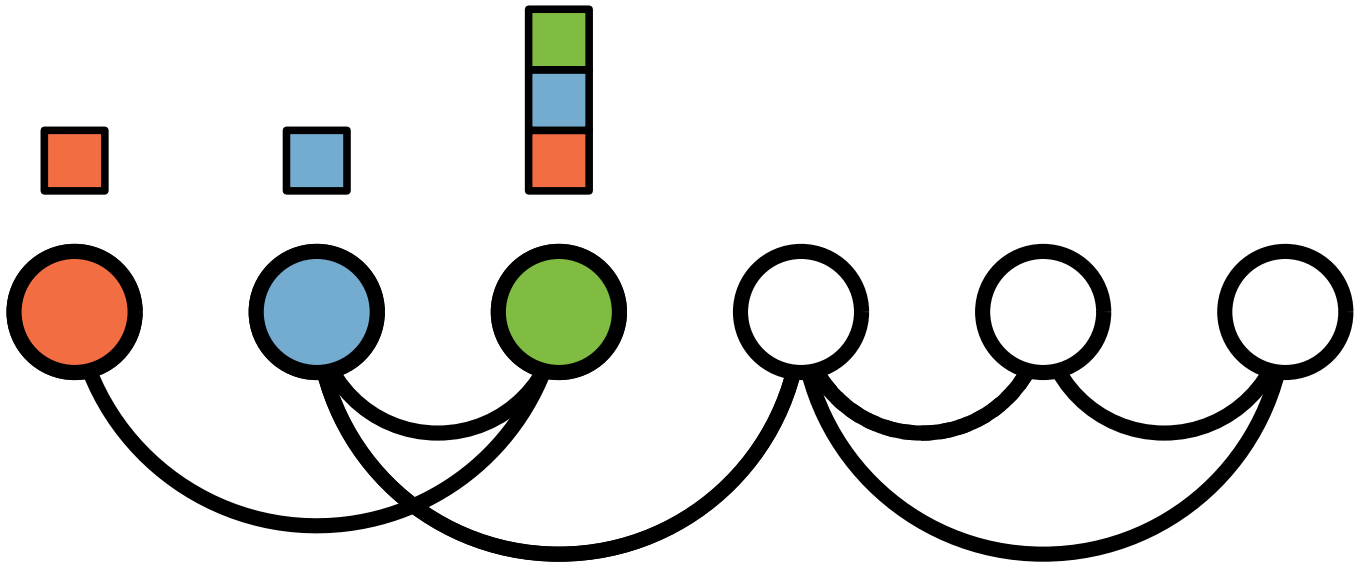
Warm-up: 2Δ-Coloring



$L(v) =$ list of colors sampled by this process

Claim. Whp $|L(v)| \leq O(\log n)$ for all v

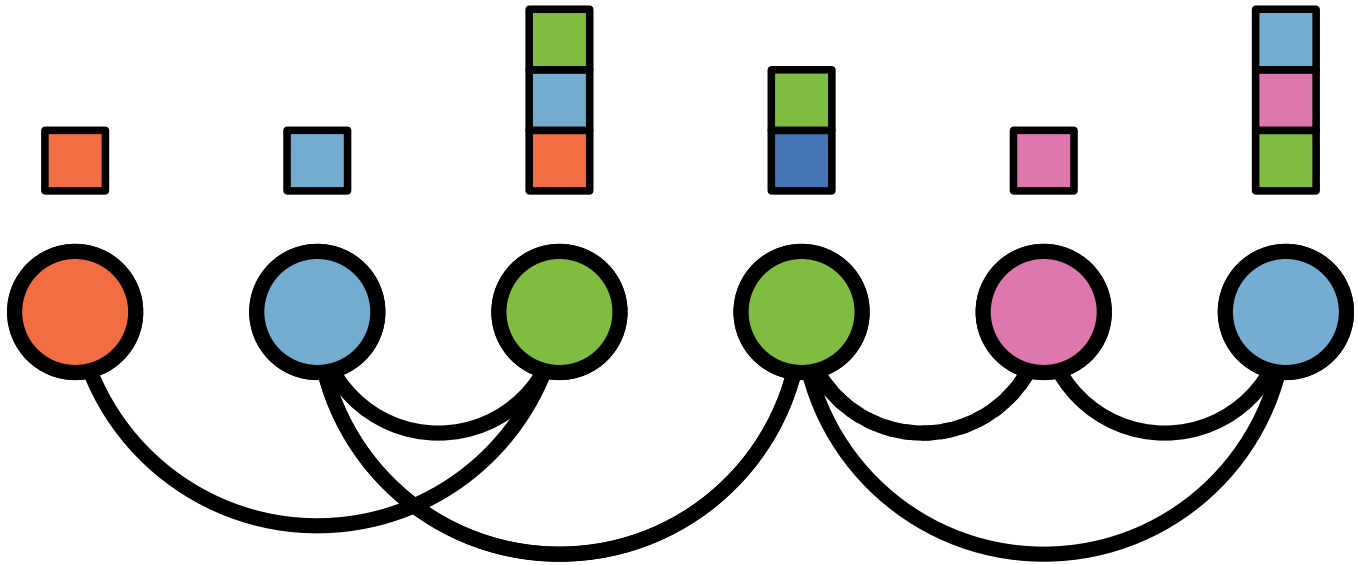
Warm-up: 2Δ-Coloring



$L(v) =$ list of colors sampled by this process

Claim. Whp $|L(v)| \leq O(\log n)$ for all v

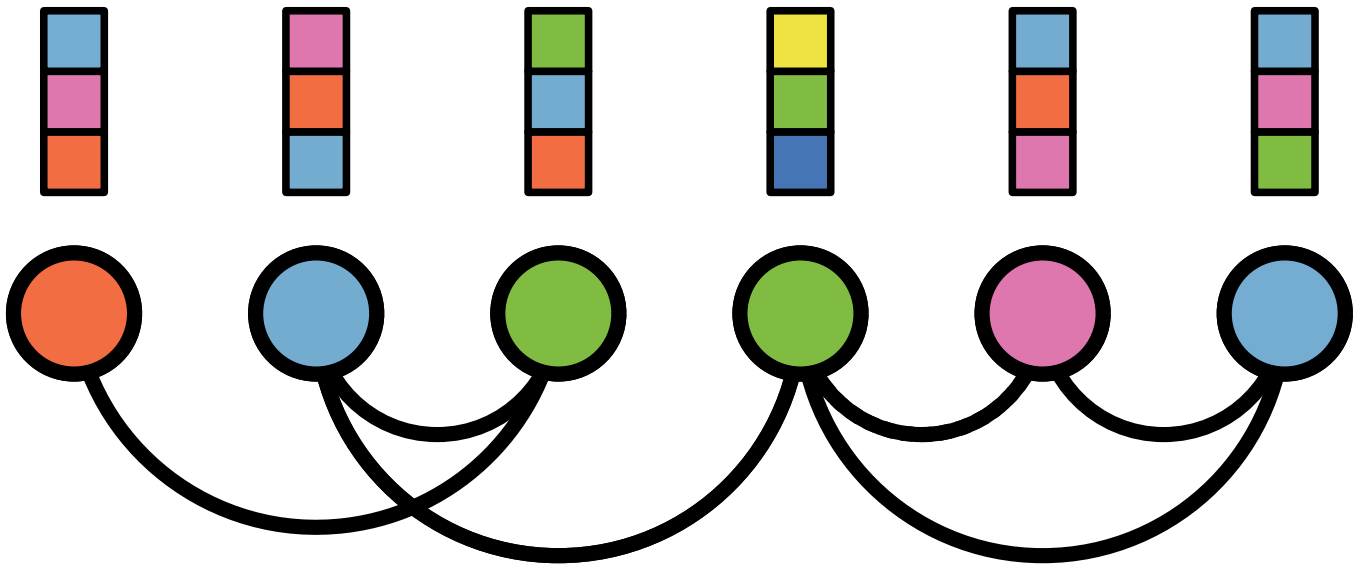
Warm-up: 2Δ-Coloring



$L(v) =$ list of colors sampled by this process

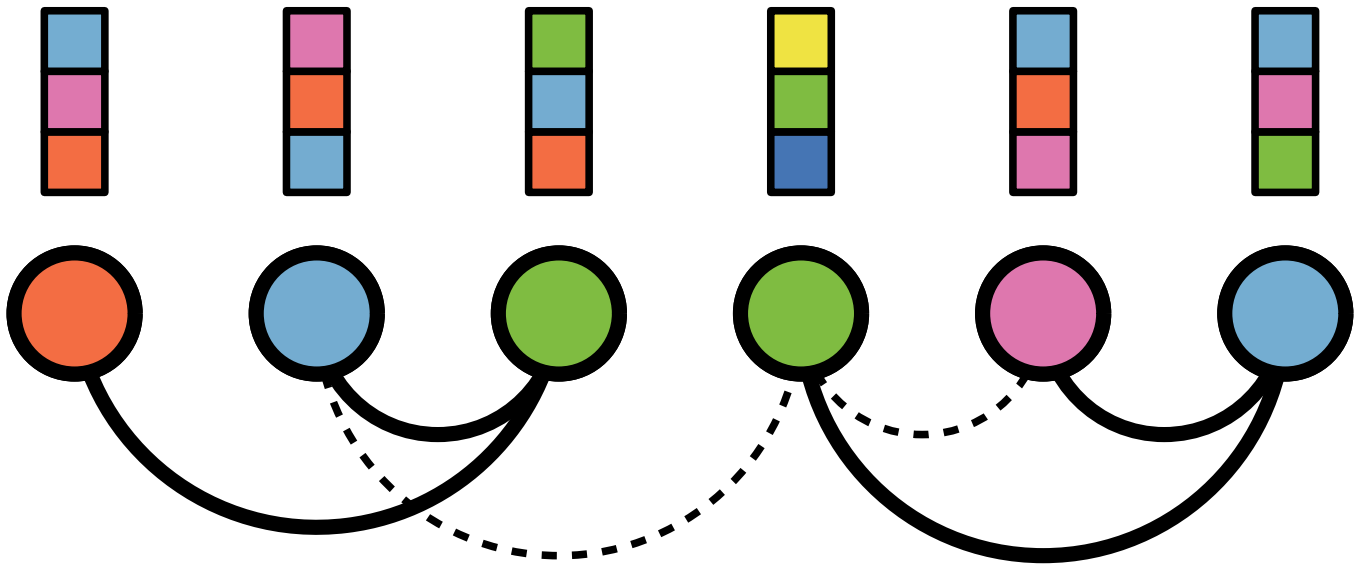
Claim. Whp $|L(v)| \leq O(\log n)$ for all v

The Sparsified Graph



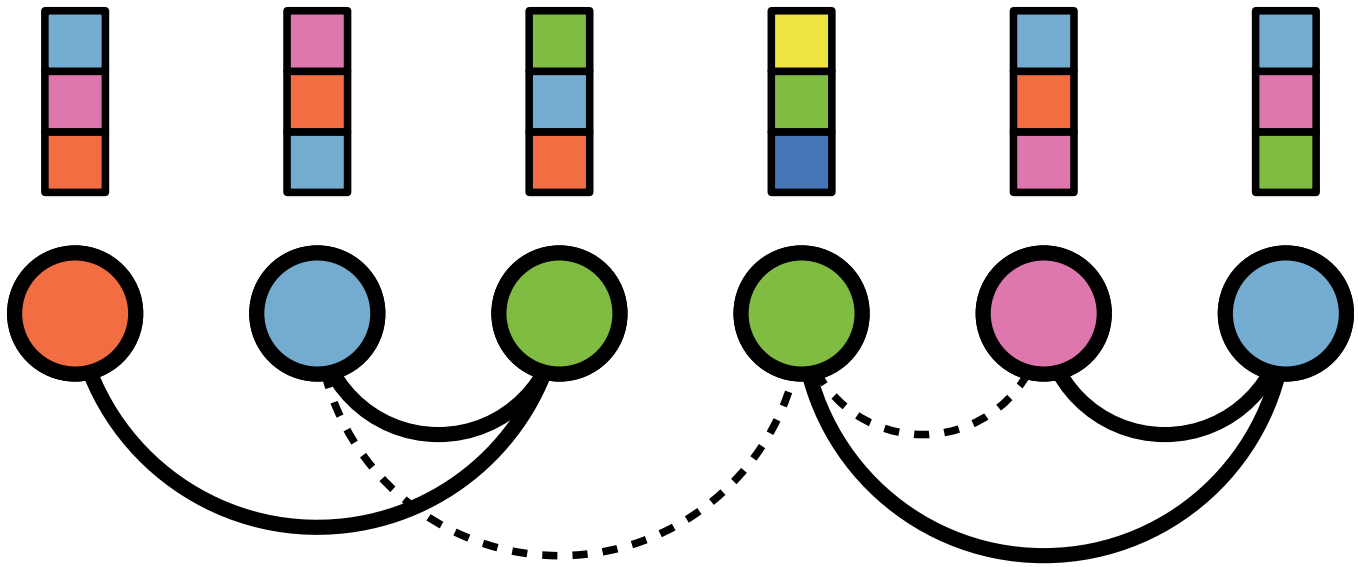
Claim. The Sparsified Graph has $O(n \log^2 n)$ edges whp.

The Sparsified Graph



Claim. The Sparsified Graph has $O(n \log^2 n)$ edges whp.

The Sparsified Graph



Claim. The Sparsified Graph has $O(n \log^2 n)$ edges whp.

Sparsified Graph: Analysis

Claim. Each v has $O(\log^2 n)$ incident edges in the sparsified graph

Sparsified Graph: Analysis

Claim. Each v has $O(\log^2 n)$ incident edges in the sparsified graph

Fix the list $L(v)$ of $O(\log n)$ colors.

$$\Pr[L(u) \cap L(v) \neq \emptyset]$$

Sparsified Graph: Analysis

Claim. Each v has $O(\log^2 n)$ incident edges in the sparsified graph

Fix the list $L(v)$ of $O(\log n)$ colors.

$$\Pr[L(u) \cap L(v) \neq \emptyset] \leq |L(v)| \cdot O\left(\frac{\log n}{\Delta}\right)$$

Sparsified Graph: Analysis

Claim. Each v has $O(\log^2 n)$ incident edges in the sparsified graph

Fix the list $L(v)$ of $O(\log n)$ colors.

$$\Pr[L(u) \cap L(v) \neq \emptyset] \leq |L(v)| \cdot O\left(\frac{\log n}{\Delta}\right) = O\left(\frac{\log^2 n}{\Delta}\right)$$

Sparsified Graph: Analysis

Claim. Each v has $O(\log^2 n)$ incident edges in the sparsified graph

Fix the list $L(v)$ of $O(\log n)$ colors.

$$\Pr[L(u) \cap L(v) \neq \emptyset] \leq |L(v)| \cdot O\left(\frac{\log n}{\Delta}\right) = O\left(\frac{\log^2 n}{\Delta}\right)$$

$\mathbb{E}[\deg(v) \text{ in sparsified graph}]$

Sparsified Graph: Analysis

Claim. Each v has $O(\log^2 n)$ incident edges in the sparsified graph

Fix the list $L(v)$ of $O(\log n)$ colors.

$$\Pr[L(u) \cap L(v) \neq \emptyset] \leq |L(v)| \cdot O\left(\frac{\log n}{\Delta}\right) = O\left(\frac{\log^2 n}{\Delta}\right)$$

$$\mathbb{E}[\deg(v) \text{ in sparsified graph}] = \sum_{u \in N(v)} \Pr[L(u) \cap L(v) \neq \emptyset]$$

Sparsified Graph: Analysis

Claim. Each v has $O(\log^2 n)$ incident edges in the sparsified graph

Fix the list $L(v)$ of $O(\log n)$ colors.

$$\Pr[L(u) \cap L(v) \neq \emptyset] \leq |L(v)| \cdot O\left(\frac{\log n}{\Delta}\right) = O\left(\frac{\log^2 n}{\Delta}\right)$$

$$\begin{aligned}\mathbb{E}[\deg(v) \text{ in sparsified graph}] &= \sum_{u \in N(v)} \Pr[L(u) \cap L(v) \neq \emptyset] \\ &\leq \Delta \cdot O\left(\frac{\log^2 n}{\Delta}\right)\end{aligned}$$

Sparsified Graph: Analysis

Claim. Each v has $O(\log^2 n)$ incident edges in the sparsified graph

Fix the list $L(v)$ of $O(\log n)$ colors.

$$\Pr[L(u) \cap L(v) \neq \emptyset] \leq |L(v)| \cdot O\left(\frac{\log n}{\Delta}\right) = O\left(\frac{\log^2 n}{\Delta}\right)$$

$$\mathbb{E}[\deg(v) \text{ in sparsified graph}] = \sum_{u \in N(v)} \Pr[L(u) \cap L(v) \neq \emptyset]$$

$$\leq \Delta \cdot O\left(\frac{\log^2 n}{\Delta}\right) \leq O(\log^2 n)$$

Sparsified Graph: Analysis

Claim. Each v has $O(\log^2 n)$ incident edges in the sparsified graph

Fix the list $L(v)$ of $O(\log n)$ colors.

$$\Pr[L(u) \cap L(v) \neq \emptyset] \leq |L(v)| \cdot O\left(\frac{\log n}{\Delta}\right) = O\left(\frac{\log^2 n}{\Delta}\right)$$

$$\mathbb{E}[\deg(v) \text{ in sparsified graph}] = \sum_{u \in N(v)} \Pr[L(u) \cap L(v) \neq \emptyset]$$

$$\leq \Delta \cdot O\left(\frac{\log^2 n}{\Delta}\right) \leq O(\log^2 n)$$

but what about $\Delta + 1$ colors?

Asymmetric Palette Sparsification (Assadi, Yazdanyar, SOSA 2025)

Algorithm.

1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size

$$\min\left\{\Delta + 1, \frac{40n \ln n}{i}\right\}$$

3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

Theorem. It colors all nodes (whp).
Claim. Average List Size $O(\log^2 n)$
Claim. Number of edges in the sparsified graph is $O(n \log^4 n)$

Why Greedy Coloring Works (whp)

Algorithm.

1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size **with** $40n \ln n/i$ **colors**
3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

$N_<(v) =$ neighbors of v colored after v

$d_<(v) = |N_<(v)| =$ nb of neighbors of v colored after v

$$\mathbb{E}[\text{colors available to } v_{i+1}\,] = \mathbb{E}[d_<(v_{i+1}) + 1] = \frac{i}{n-1} \cdot \Delta + 1$$

Why Greedy Coloring Works (whp)

Algorithm.

1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size **with** $40n \ln n/i$ **colors**
3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

$N_<(v) =$ neighbors of v colored after v

$d_<(v) = |N_<(v)| =$ nb of neighbors of v colored after v

$$\mathbb{E}[\text{colors available to } v_{i+1}\,] = \mathbb{E}[d_<(v_{i+1}) + 1] = \frac{i}{n-1} \cdot \Delta + 1$$

$$\Pr[L(v_{i+1}) \text{ has no avail color}]$$

Why Greedy Coloring Works (whp)

Algorithm.

1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size **with $40n \ln n/i$ colors**
3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

$N_<(v) =$ neighbors of v colored after v

$d_<(v) = |N_<(v)| =$ nb of neighbors of v colored after v

$$\mathbb{E}[\text{colors available to } v_{i+1}] = \mathbb{E}[d_<(v_{i+1}) + 1] = \frac{i}{n-1} \cdot \Delta + 1$$

$$\Pr[L(v_{i+1}) \text{ has no avail color}] = \left(\frac{\Delta + 1 - d_<(v_{i+1})}{\Delta + 1}\right)^{\frac{40n \ln n}{i}}$$

Why Greedy Coloring Works (whp)

Algorithm.

1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size **with $40n \ln n/i$ colors**
3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

$N_<(v) =$ neighbors of v colored after v

$d_<(v) = |N_<(v)| =$ nb of neighbors of v colored after v

$$\mathbb{E}[\text{colors available to } v_{i+1}] = \mathbb{E}[d_<(v_{i+1}) + 1] = \frac{i}{n-1} \cdot \Delta + 1$$

$$\Pr[L(v_{i+1}) \text{ has no avail color}] = \left(1 - \frac{d_<(v_{i+1})}{\Delta + 1}\right)^{\frac{40n \ln n}{i}}$$

Why Greedy Coloring Works (whp)

Algorithm.

1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size **with $40n \ln n/i$ colors**
3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

$N_<(v) =$ neighbors of v colored after v

$d_<(v) = |N_<(v)| =$ nb of neighbors of v colored after v

$$\mathbb{E}[\text{colors available to } v_{i+1}\,] = \mathbb{E}[d_<(v_{i+1}) + 1] = \frac{i}{n-1} \cdot \Delta + 1$$

$$\Pr[L(v_{i+1}) \text{ has no avail color}] \leq \exp\left(-\frac{d_<(v_{i+1})}{\Delta + 1}\right)^{\frac{40n \ln n}{i}}$$

Why Greedy Coloring Works (whp)

Algorithm.

1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size **with** $40n \ln n/i$ **colors**
3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

$N_<(v) =$ neighbors of v colored after v

$d_<(v) = |N_<(v)| =$ nb of neighbors of v colored after v

$$\mathbb{E}[\text{colors available to } v_{i+1}] = \mathbb{E}[d_<(v_{i+1}) + 1] = \frac{i}{n-1} \cdot \Delta + 1$$

$$\Pr[L(v_{i+1}) \text{ has no avail color}] \leq \exp\left(-\frac{d_<(v_{i+1})}{\Delta + 1} \cdot \frac{40n \ln n}{i}\right)$$

Why Greedy Coloring Works (whp)

Algorithm.

1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size **with $40n \ln n/i$ colors**
3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

$N_<(v) =$ neighbors of v colored after v

$d_<(v) = |N_<(v)| =$ nb of neighbors of v colored after v

$$\mathbb{E}[\text{colors available to } v_{i+1}] = \mathbb{E}[d_<(v_{i+1}) + 1] = \frac{i}{n-1} \cdot \Delta + 1$$

$$\begin{aligned}\Pr[L(v_{i+1}) \text{ has no avail color}] &\leq \exp\left(-\frac{d_<(v_{i+1})}{\Delta+1} \cdot \frac{40n \ln n}{i}\right) \\ &\leq \exp(-40 \ln n) = n^{-40}\end{aligned}$$

Average List Size

Algorithm.

1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size **with** $40n \ln n/i$ **colors**
3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

$$\sum_{i=1}^{n} |L(v_i)|$$

Average List Size

Algorithm.

1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size **with** $40n \ln n/i$ **colors**
3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

$$\sum_{i=1}^{n} |L(v_i)| = 40n \ln n \cdot \sum_{i=1}^{n} \frac{1}{i}$$

Average List Size

Algorithm.

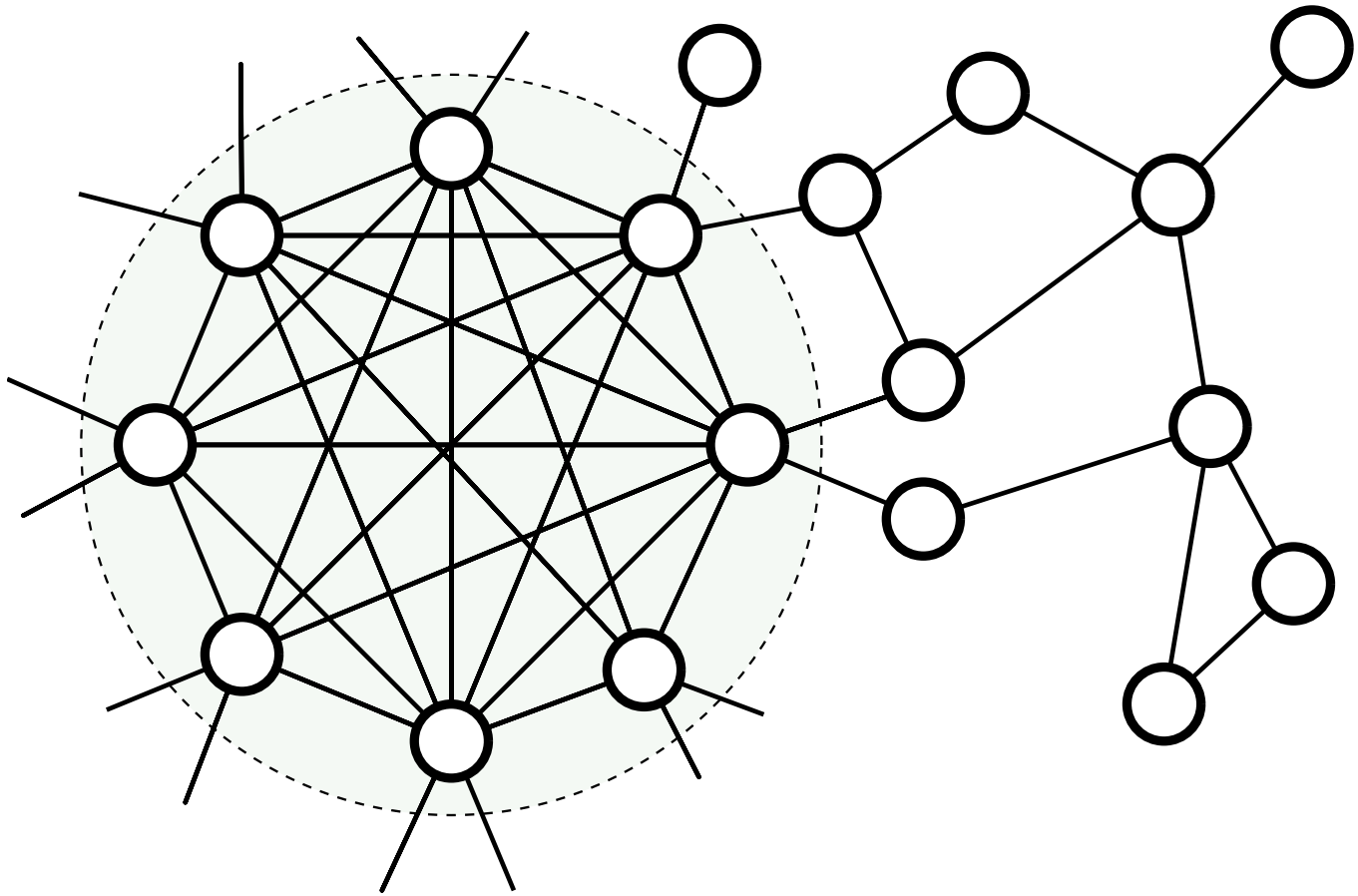
1. Sample a **permutation** of the nodes $\pi = [v_1, v_2, \ldots, v_n]$
2. Sample $L(v_i) \subseteq [\Delta + 1]$ of size **with** $40n \ln n/i$ **colors**
3. Color vertices greedily in the reverse order $v_n, v_{n-1}, \ldots$

$$\sum_{i=1}^{n} |L(v_i)| = 40n \ln n \cdot \sum_{i=1}^{n} \frac{1}{i} = O(n \ln^2 n)$$

Palette Sparsification (Assadi, Chen, Khanna, SODA 2019)

Theorem. If we sample lists of $O(\log n)$ **colors** in $[\Delta + 1]$ for each vertex, w.h.p., each vertex can be colored with a color from its list.

Sparse-Dense Decomposition (Harris, Schneider, Su, J. ACM 2018)



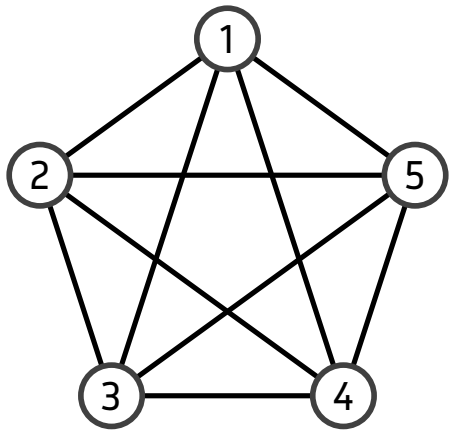
ε-Almost-Clique:

- $|C| \leq (1+\varepsilon)\Delta$,
- $|N(v) \cap C| \geq (1-\varepsilon)\Delta$ for every $v \in C$

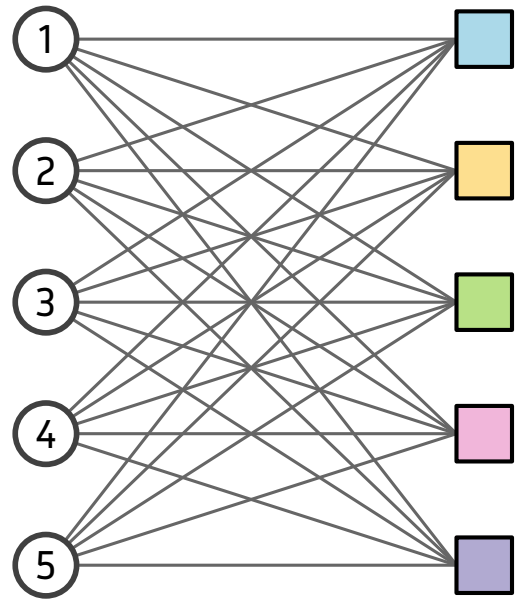
$\Omega(\Delta)$-sparse nodes

Palette Graph

clique K

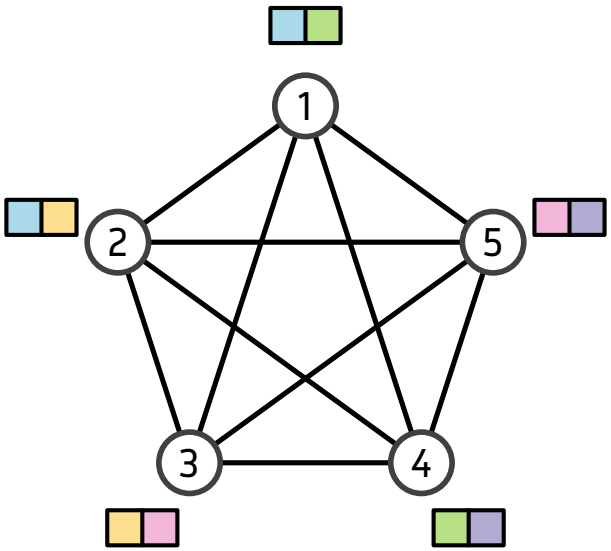


vertices of K palette $[\Delta + 1]$

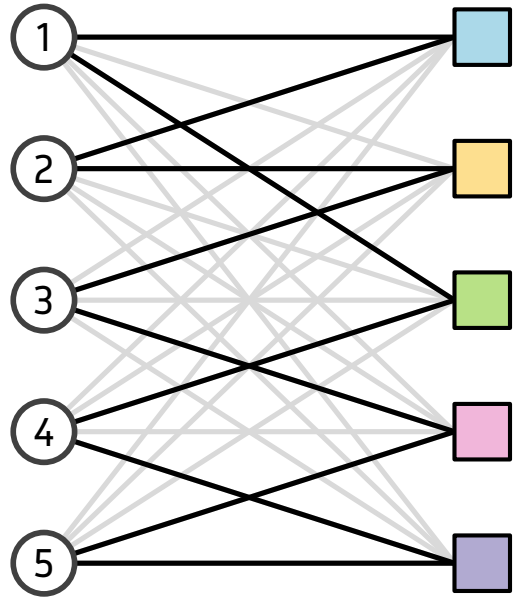


Palette Graph

clique K

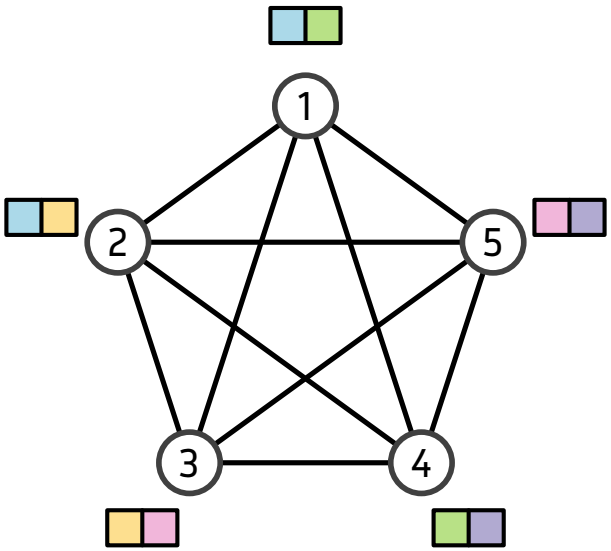


vertices of K palette $[\Delta + 1]$

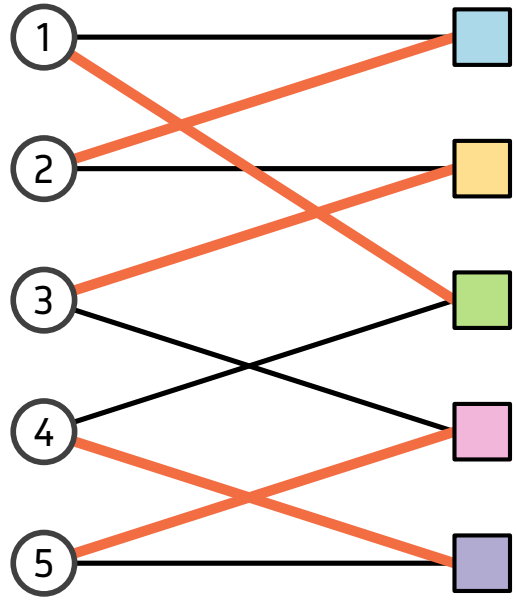


Palette Graph

clique K

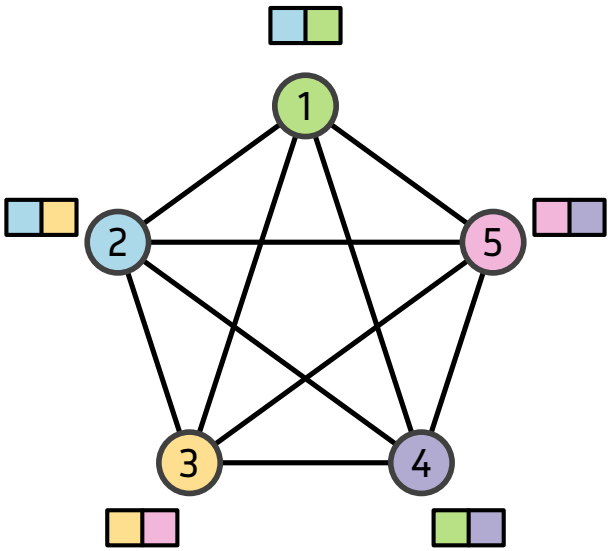


vertices of K palette $[\Delta + 1]$

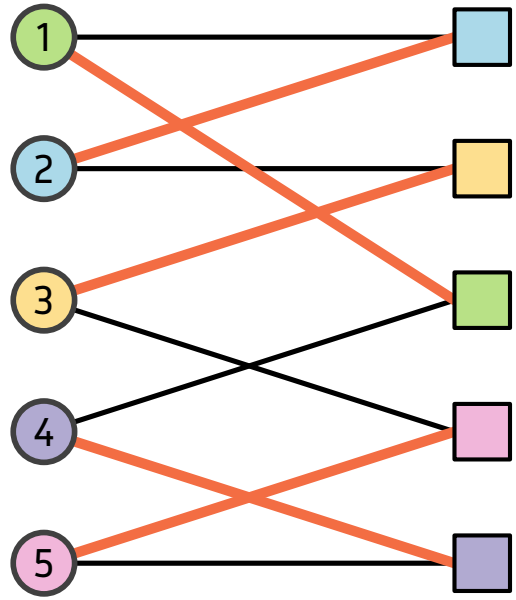


Palette Graph

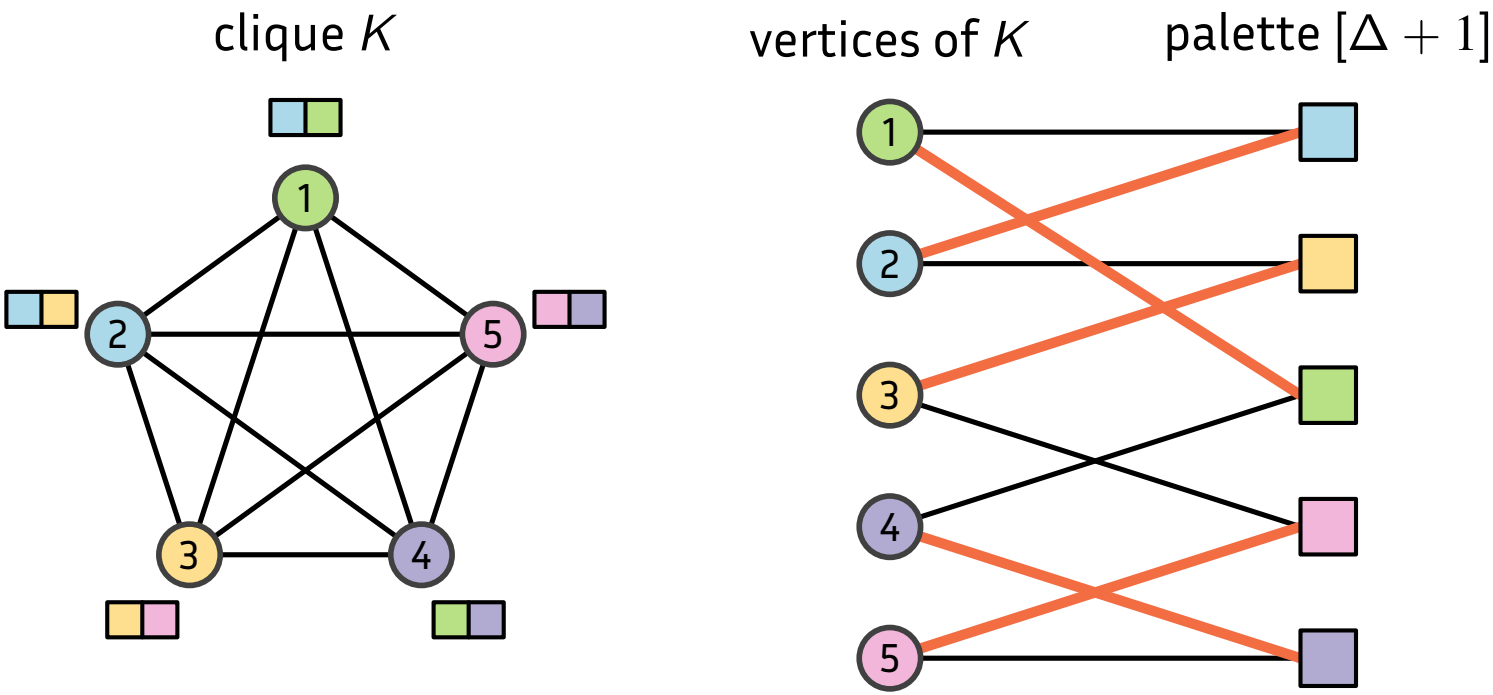
clique K



vertices of K palette $[\Delta + 1]$

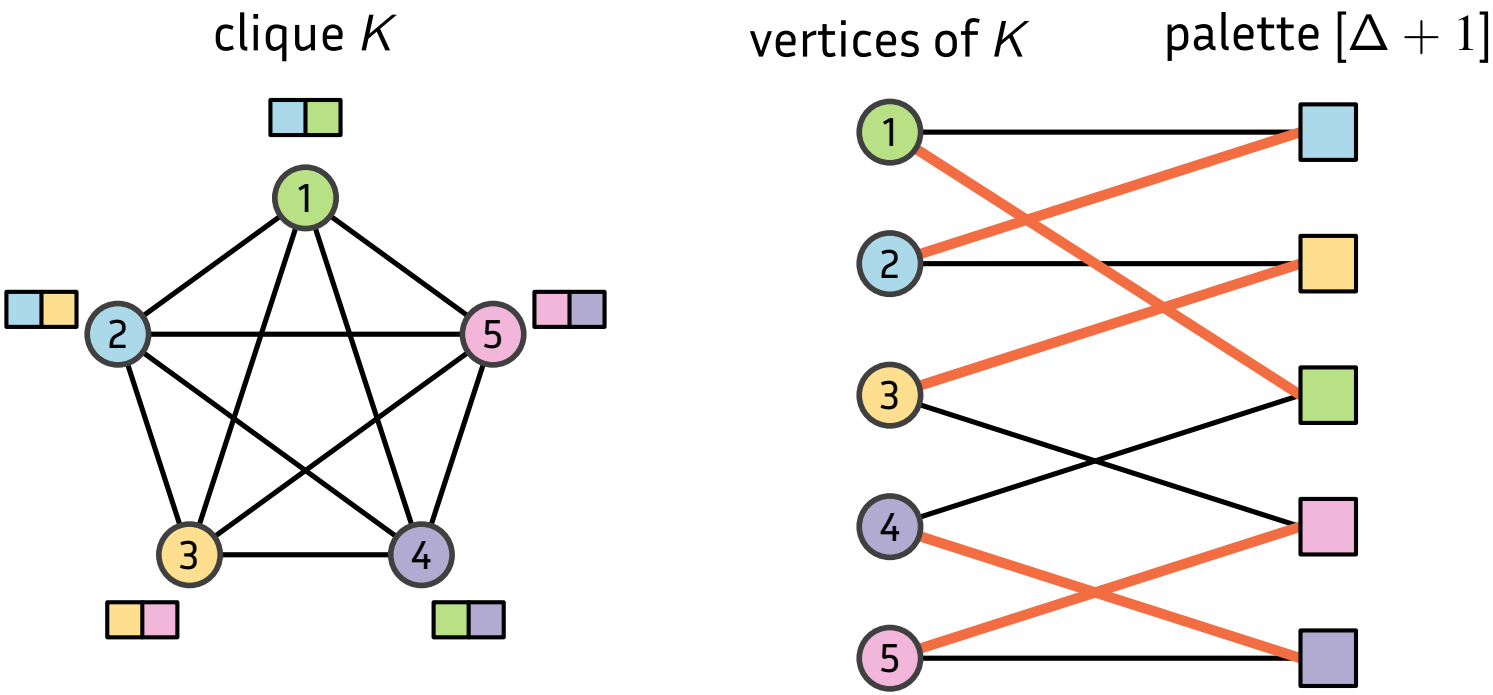


Palette Graph



Claim. Whp the Hall condition holds, thus a coloring exists.

Palette Graph



Claim. Whp the Hall condition holds, thus a coloring exists.

It is technical, so let's not talk about this

Chapitre 2: Fewer Colors?

Existential Results

Theorem. (Brooks, 1941) A graph with maximum degree $\Delta \geq 3$ is **Δ-colorable** unless it contains a **$(\Delta + 1)$-clique**.

Existential Results

Theorem. (Brooks, 1941) A graph with maximum degree $\Delta \geq 3$ is **Δ-colorable** unless it contains a **$(\Delta + 1)$-clique**.

Theorem. (Reed, 1999) A graph with maximum degree $\Delta \geq 10^{14}$ is **$(\Delta - 1)$-colorable** unless it contains a **Δ-clique**.

Existential Results

Theorem. (Brooks, 1941) A graph with maximum degree $\Delta \geq 3$ is Δ-**colorable** unless it contains a $(\Delta + 1)$-**clique**.

Theorem. (Reed, 1999) A graph with maximum degree $\Delta \geq 10^{14}$ is $(\Delta - 1)$-**colorable** unless it contains a Δ-**clique**.

Theorem. (Molloy & Reed, 2014) For all $k \leq k_\Delta \approx \sqrt{\Delta}$, a graph of maximum degree Δ either

- is $(\Delta - k)$-**colorable**, unless
- it has a **"local" subgraph** that is ***not*** $(\Delta - k)$-colorable.

Existential Results

Theorem. (Brooks, 1941) A graph with maximum degree $\Delta \geq 3$ is Δ-**colorable** unless it contains a $(\Delta + 1)$-**clique**.

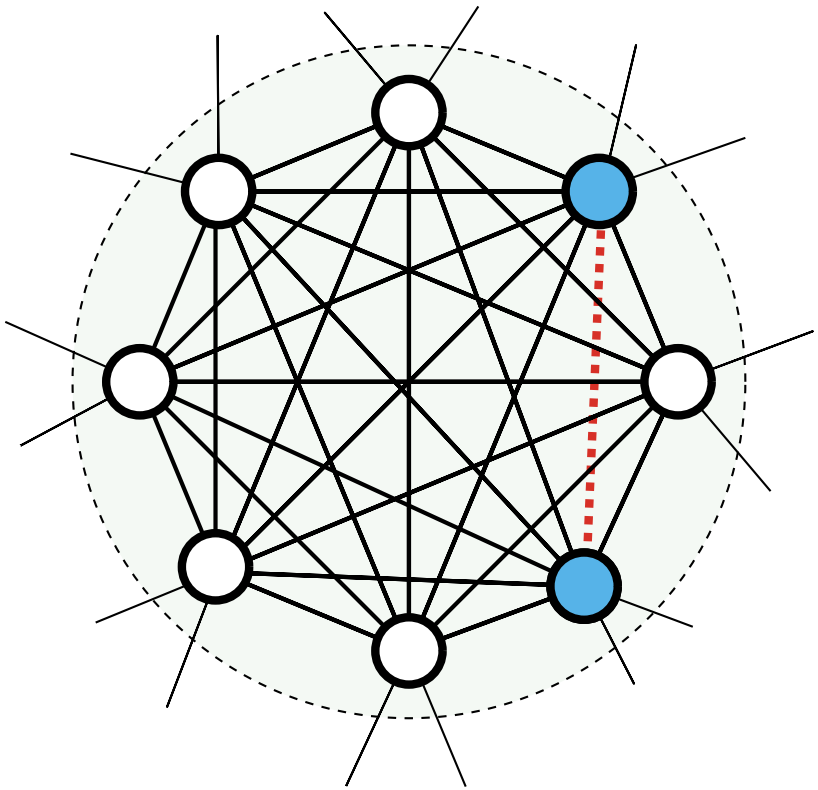
Theorem. (Reed, 1999) A graph with maximum degree $\Delta \geq 10^{14}$ is $(\Delta - 1)$-**colorable** unless it contains a Δ-**clique**.

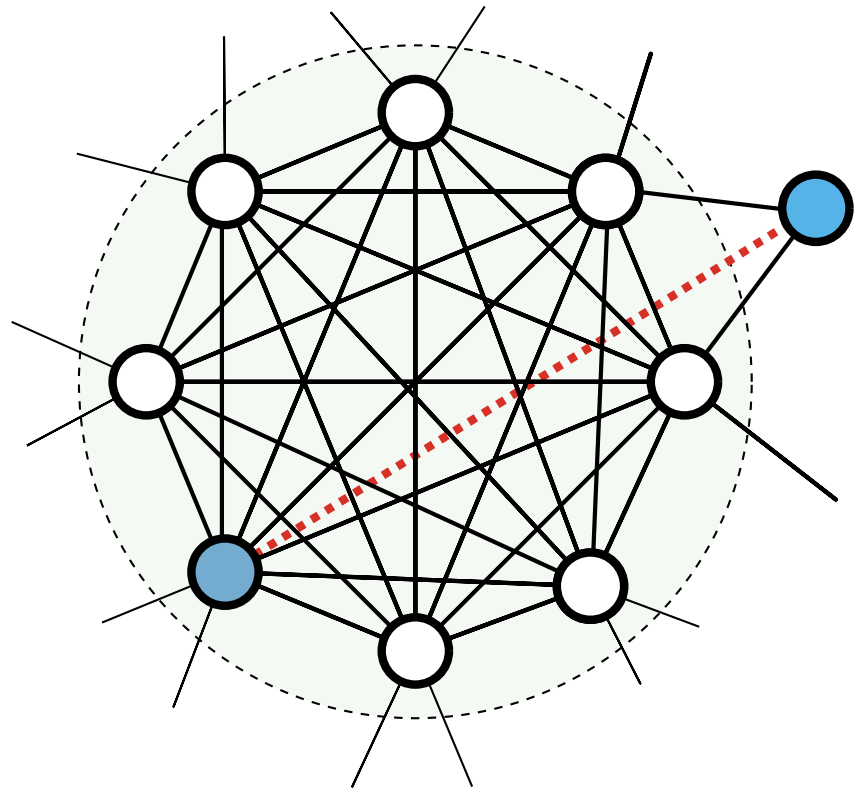
Theorem. (Molloy & Reed, 2014) For all $k \leq k_\Delta \approx \sqrt{\Delta}$, a graph of maximum degree Δ either

- is $(\Delta - k)$-**colorable**, unless
- it has a **"local" subgraph** that is ***not*** $(\Delta - k)$-colorable.

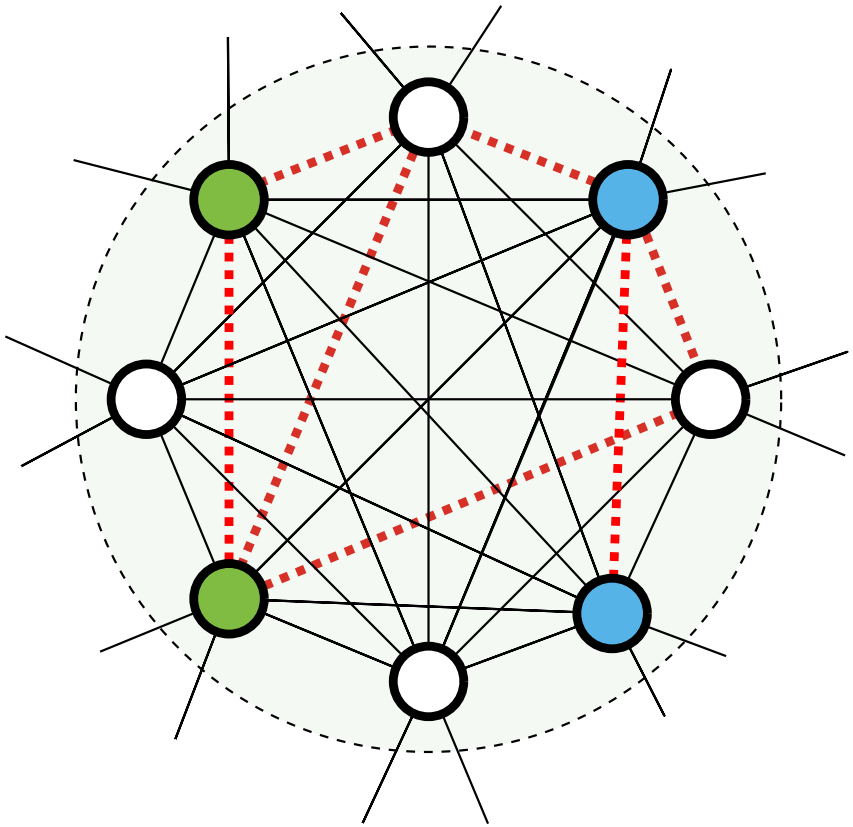
Can we do any of this in semi-streaming?

Obstructions to Palette Sparsification for Δ-Coloring

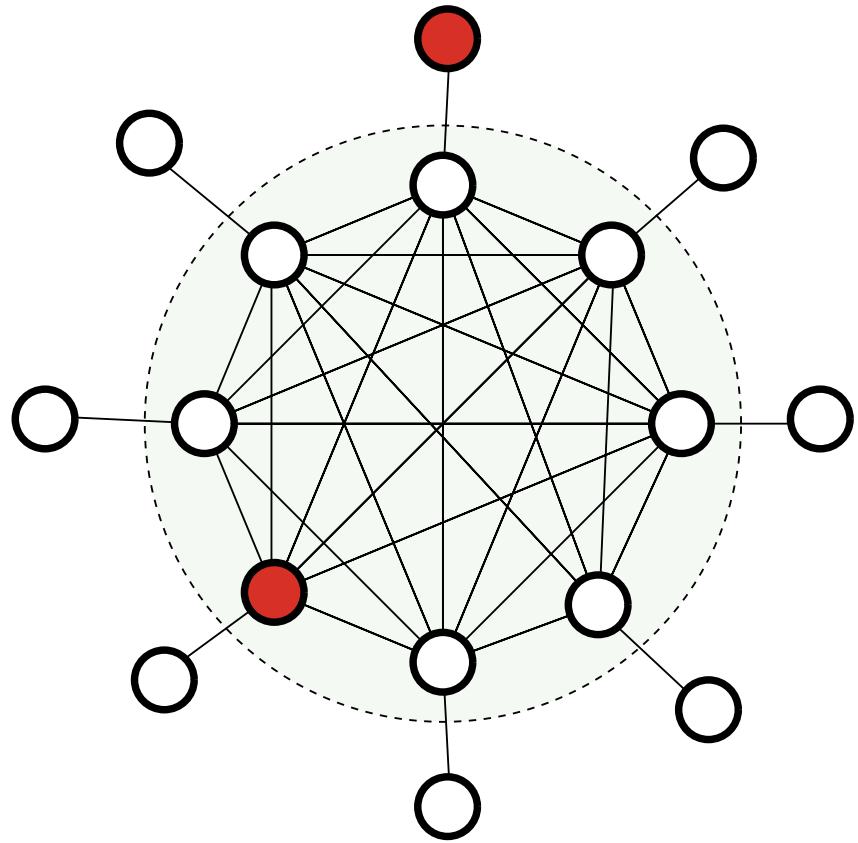




Otherwise Palette Sparsification Works

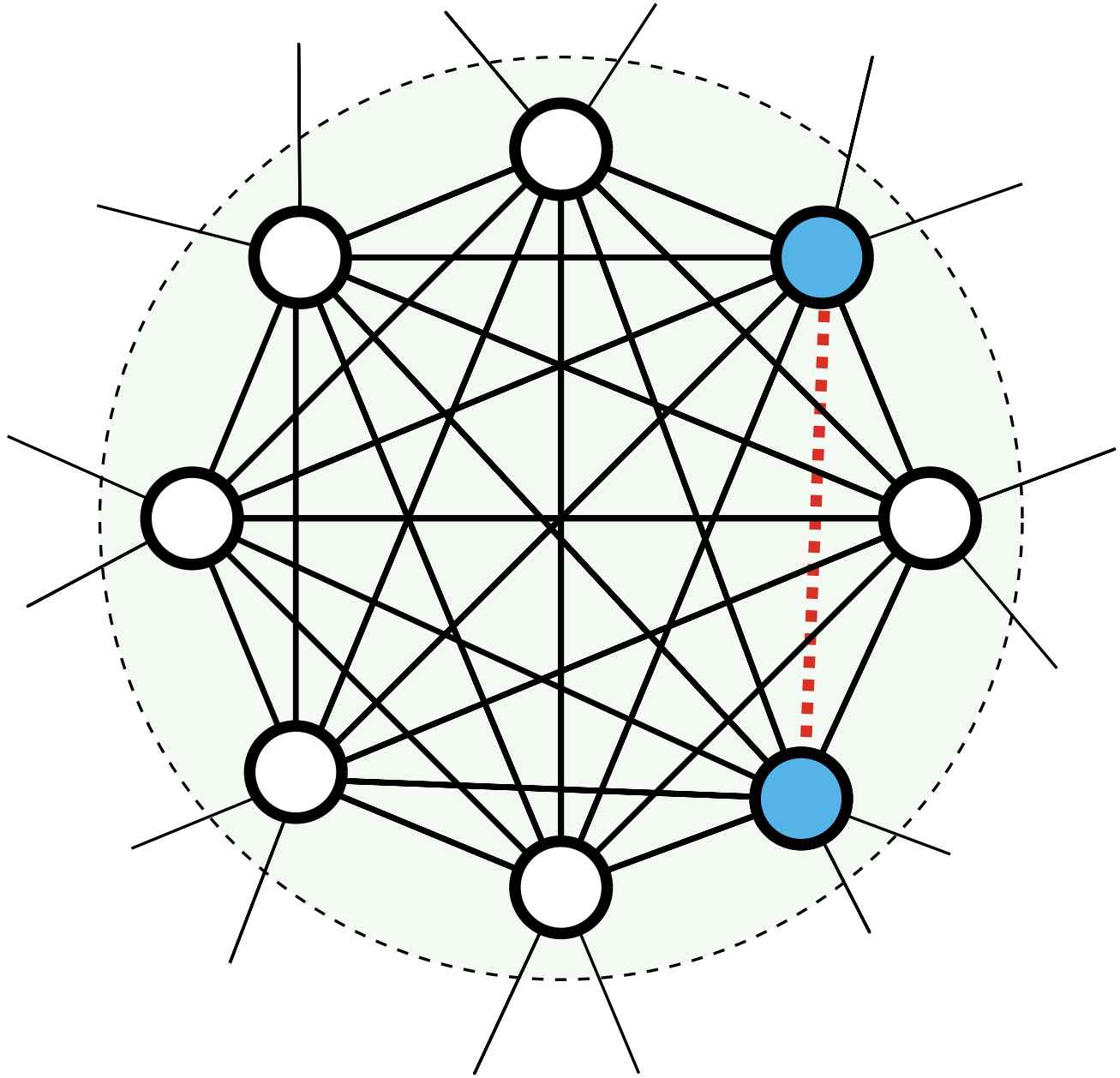


"Holey" Almost-Cliques



"Expanding" Almost-Cliques

How to Find the Missing Edge?

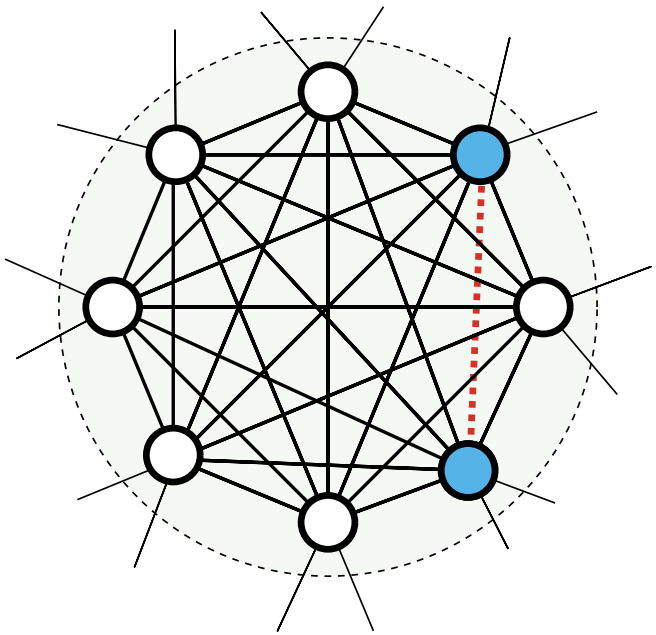


Sparse Recovery

a vector is k-sparse if it has $\leq k$ non-zero values

$$x = [0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, -1, 0] \text{ is 3-sparse}$$

storing a k-sparse vector takes $O(k \log n)$ bits

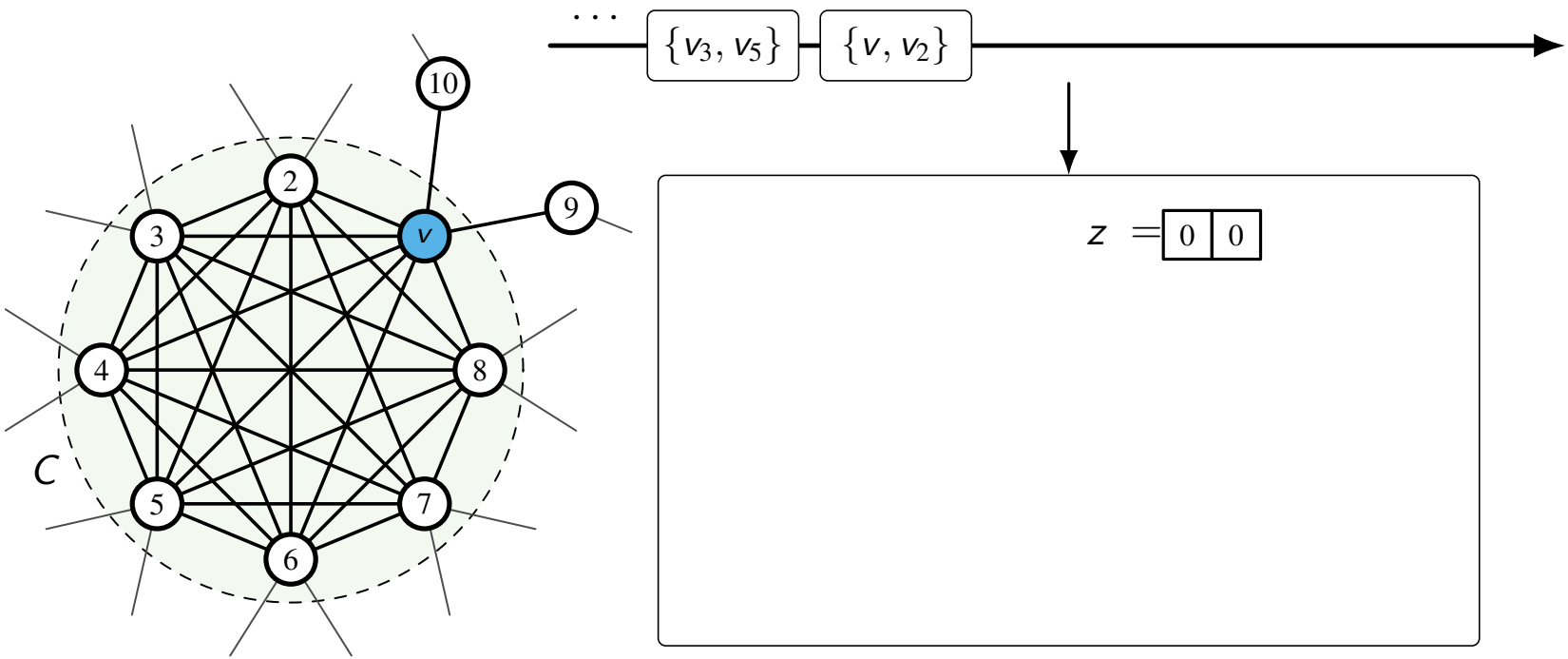


$$x = (1_C - 1_{N(v)})_u = \begin{cases} 1 & \text{if } u \in C \setminus N(v) \\ -1 & \text{if } u \in N(v) \setminus C \\ 0 & \text{otherwise} \end{cases}$$

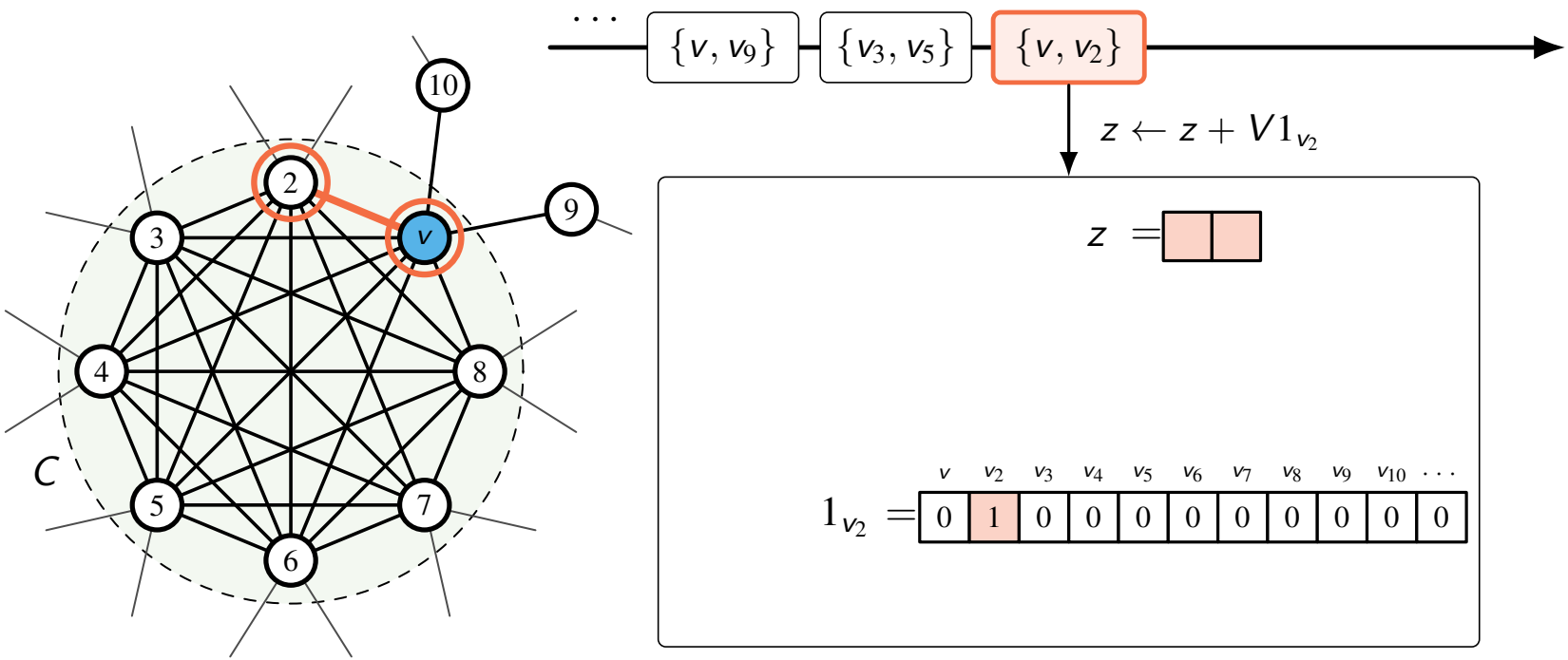
Problem. 1_C and $1_{N(v)}$ are not sparse.

Claim. There is a matrix $V : \mathbb{F}_p^n \to \mathbb{F}_p^{2k}$ such that x can be recovered from Vx if x is k-sparse.

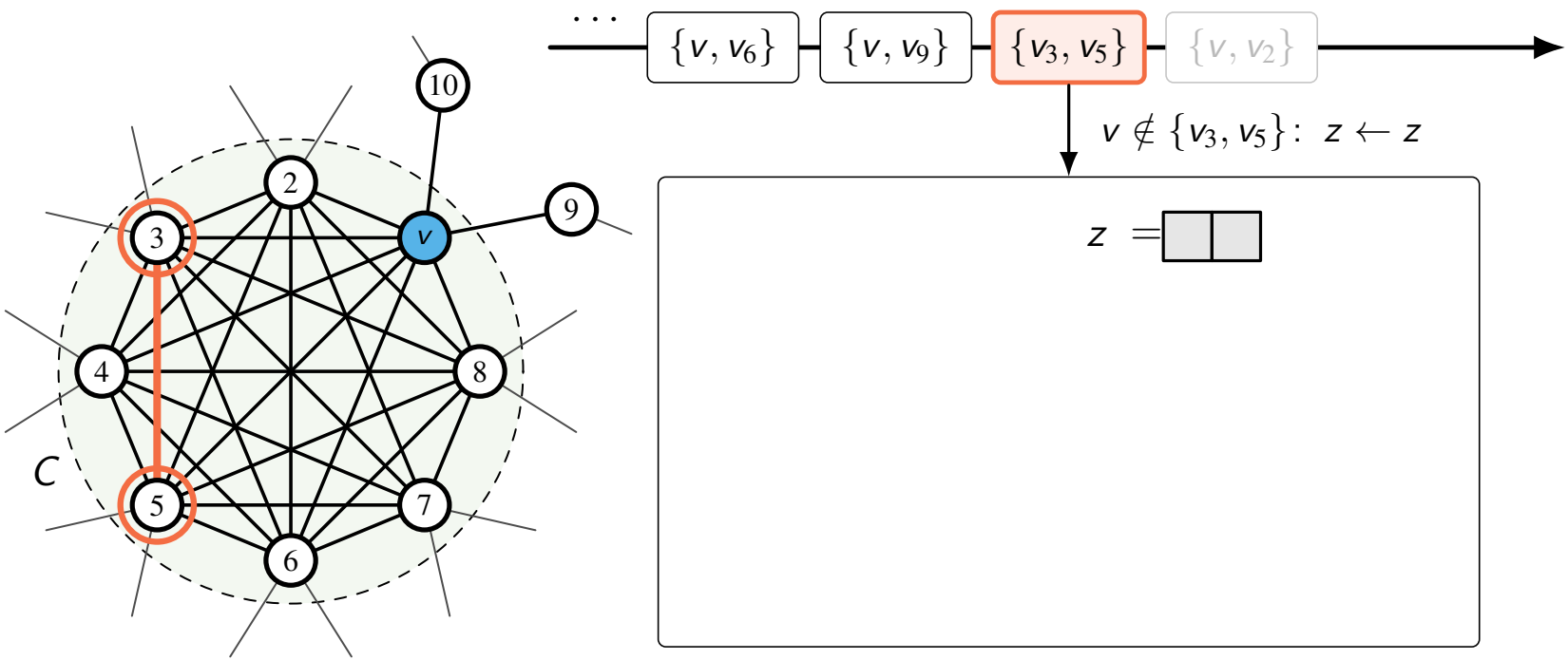
Finding the Missing Edge in Semi-Streaming (Assadi, Kumar, Mittal, STOC 2022)



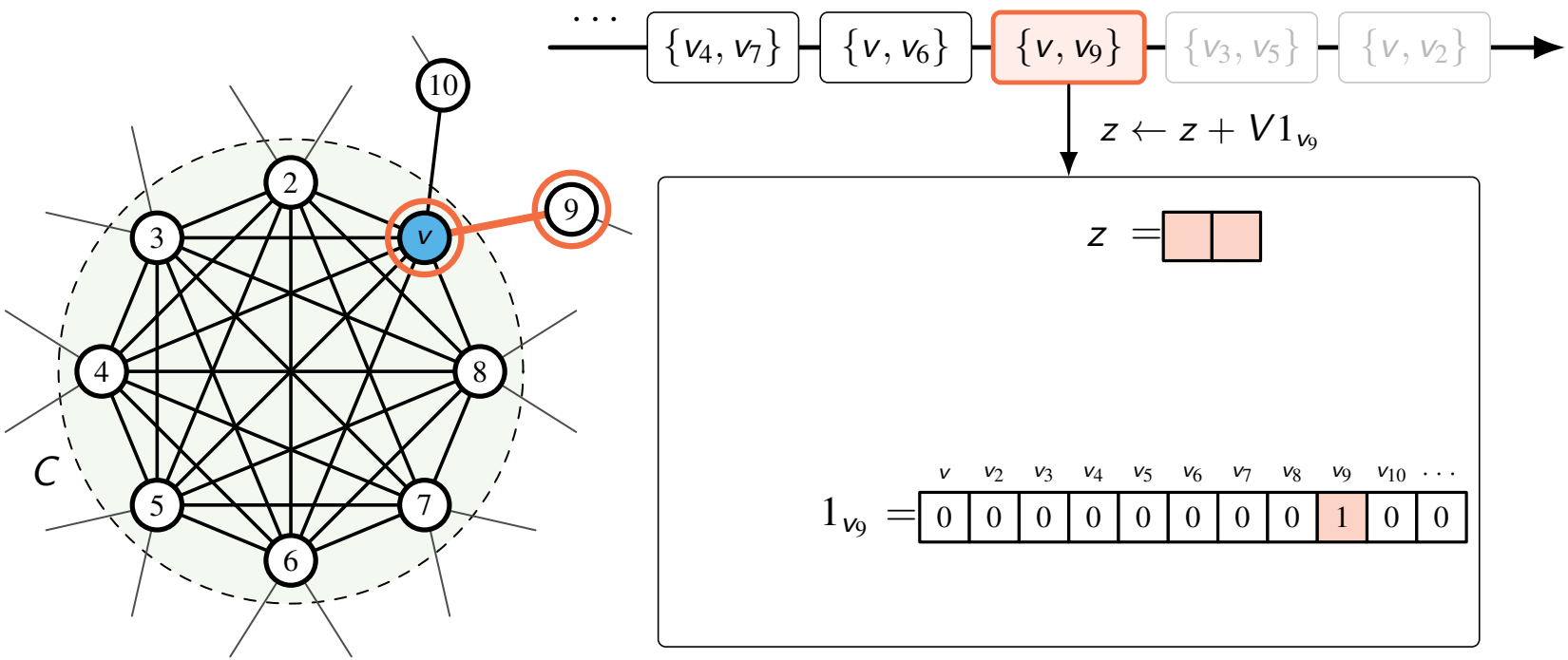
Finding the Missing Edge in Semi-Streaming (Assadi, Kumar, Mittal, STOC 2022)



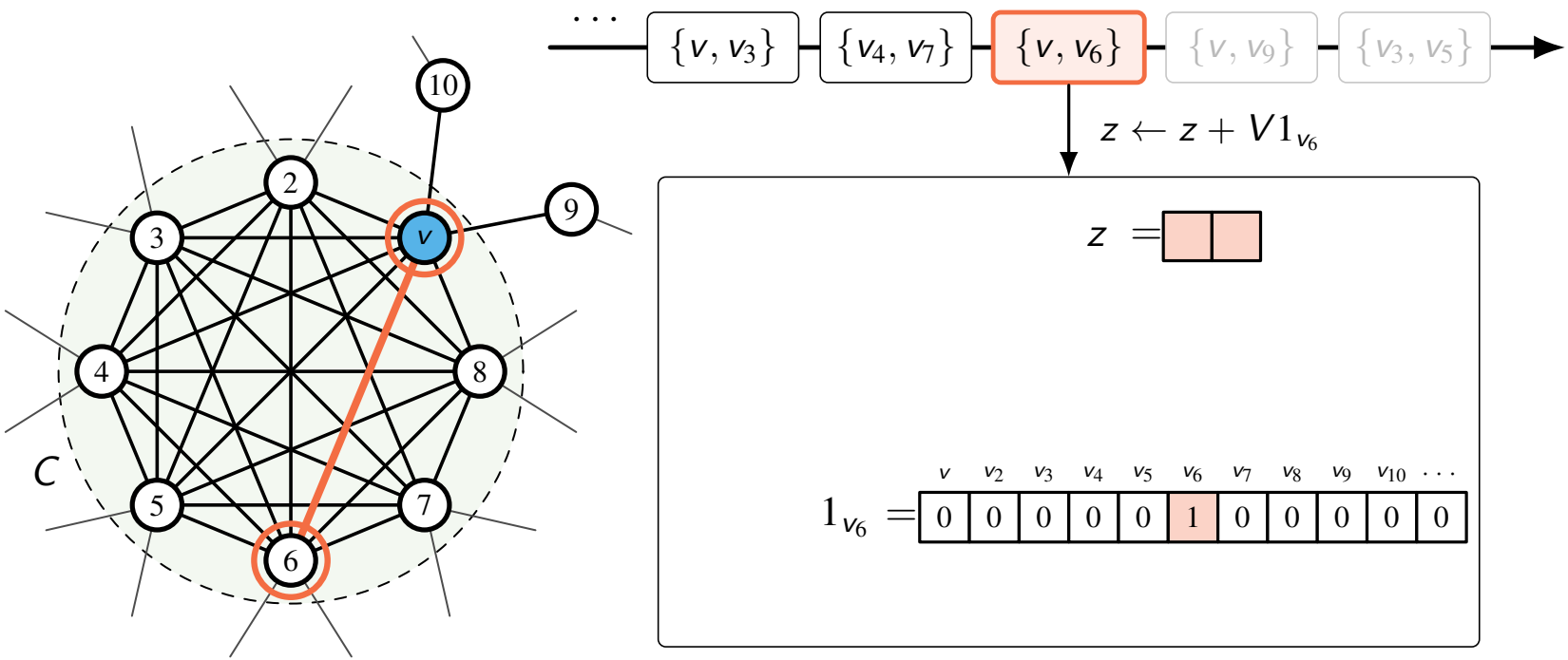
Finding the Missing Edge in Semi-Streaming (Assadi, Kumar, Mittal, STOC 2022)



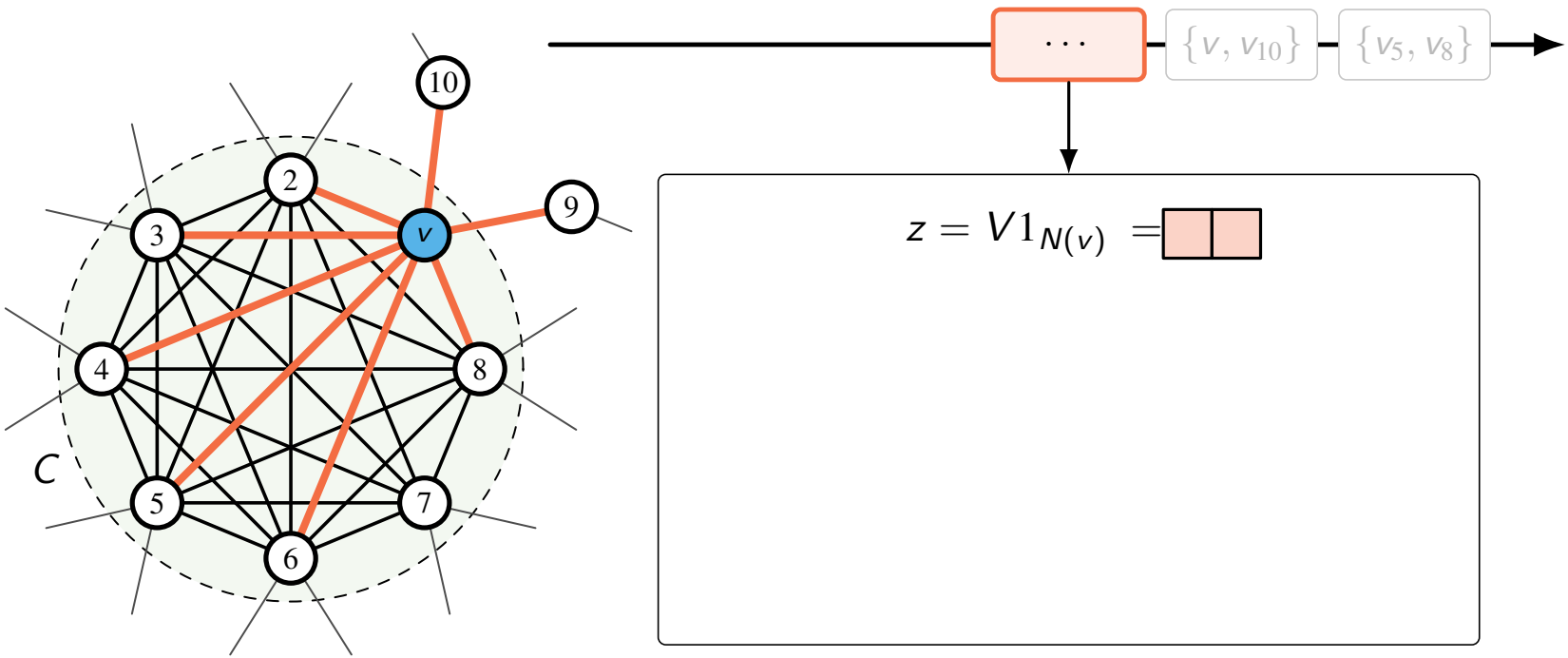
Finding the Missing Edge in Semi-Streaming (Assadi, Kumar, Mittal, STOC 2022)



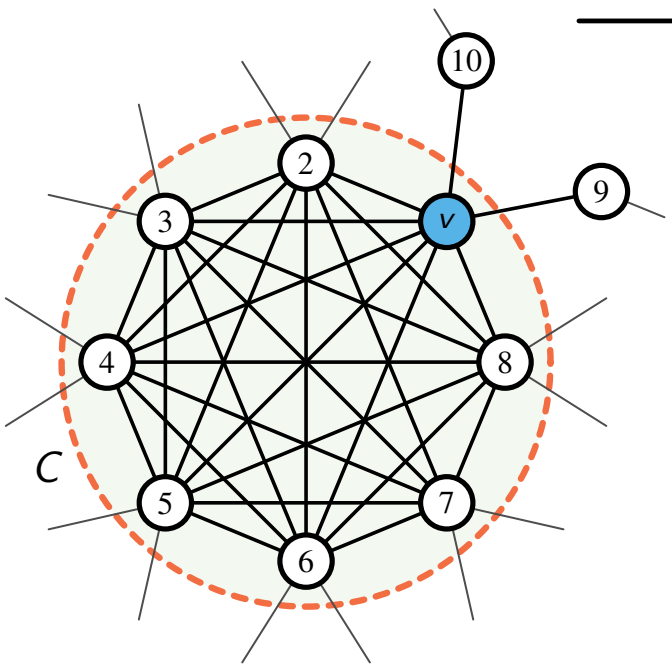
Finding the Missing Edge in Semi-Streaming (Assadi, Kumar, Mittal, STOC 2022)

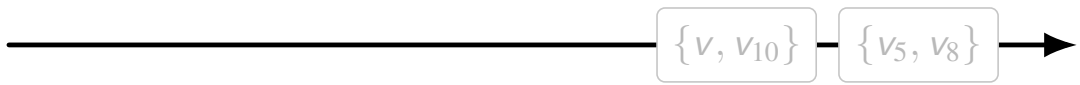


Finding the Missing Edge in Semi-Streaming (Assadi, Kumar, Mittal, STOC 2022)



Finding the Missing Edge in Semi-Streaming (Assadi, Kumar, Mittal, STOC 2022)

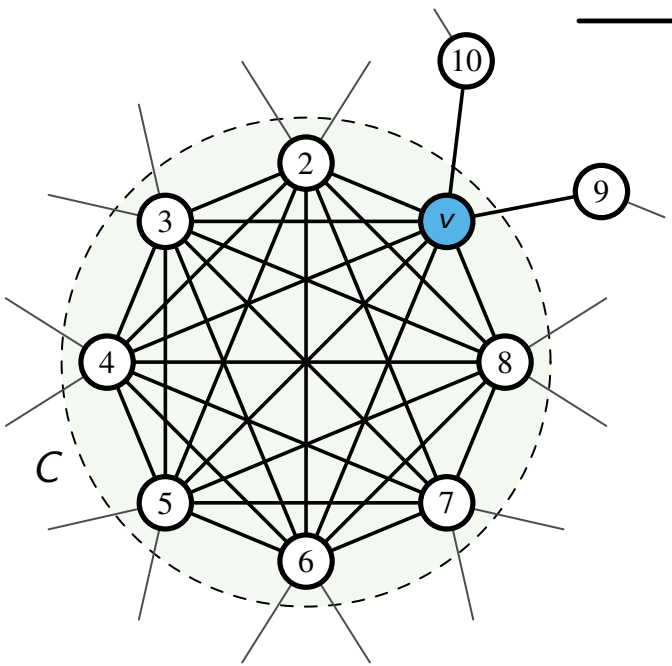


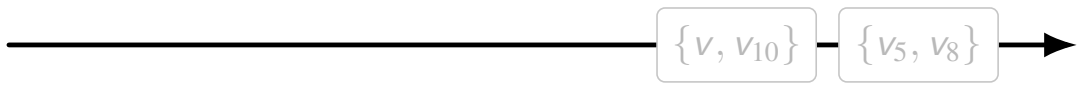


$z = V1_{N(v)}$ =

$V1_C$ =

Finding the Missing Edge in Semi-Streaming (Assadi, Kumar, Mittal, STOC 2022)



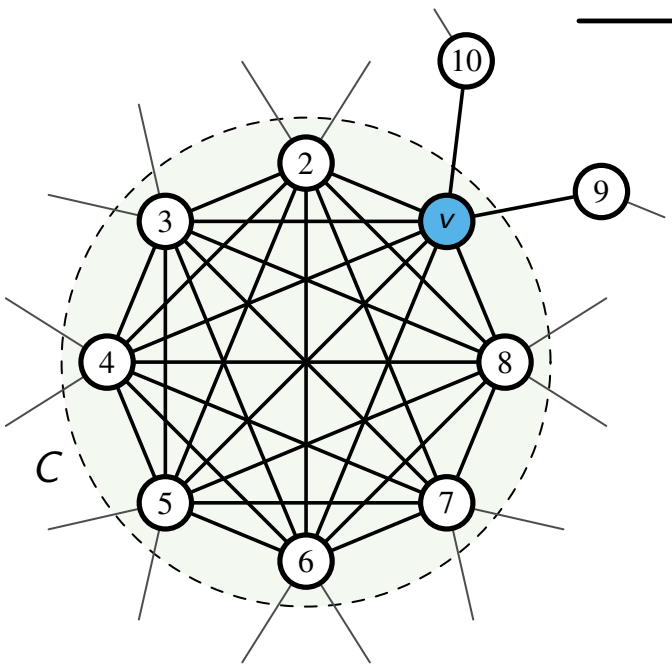


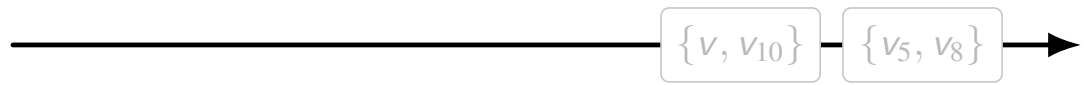
$z = V1_{N(v)}$ =

$V1_C$ =

$V(1_C - 1_{N(v)})$ =

Finding the Missing Edge in Semi-Streaming (Assadi, Kumar, Mittal, STOC 2022)





$z = V1_{N(v)} =$ 

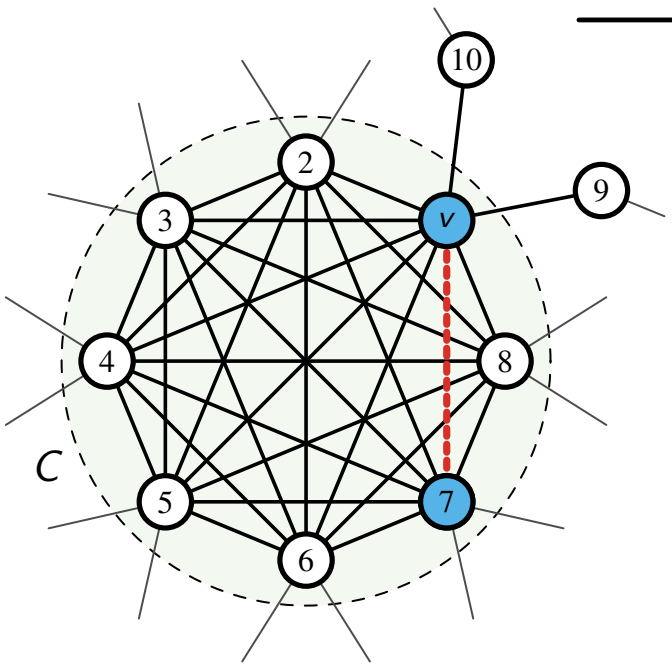
$V1_C =$ 

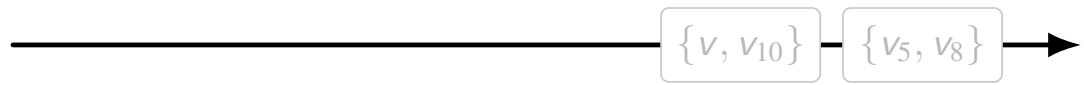
$V(1_C - 1_{N(v)}) =$ 

$1_C - 1_{N(v)} =$

v	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	$\cdots$
1	0	0	0	0	0	1	0	−1	−1	0

Finding the Missing Edge in Semi-Streaming (Assadi, Kumar, Mittal, STOC 2022)





$$z = V1_{N(v)} =$$

$$V1_C =$$

$$V(1_C - 1_{N(v)}) =$$

$1_C - 1_{N(v)} =$

v	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	$\cdots$
1	0	0	0	0	0	1	0	−1	−1	0

$$\Rightarrow \{v, v_7\} \notin E(G)$$

Sparse Recovery

$p > n$ a prime number

$$V = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ 1 & 2^2 & 3^2 & \dots & n^2 \\ \vdots & \vdots & \dots & \vdots & \\ 1 & 2^{2k-1} & 3^{2k-1} & \dots & n^{2k-1} \end{bmatrix}$$

Claim. $V : \mathbb{F}_p^n \to \mathbb{F}_p^{2k}$ is injective on k-sparse vectors

Sparse Recovery

$p > n$ a prime number

$$V = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ 1 & 2^2 & 3^2 & \dots & n^2 \\ \vdots & \vdots & \dots & \vdots & \\ 1 & 2^{2k-1} & 3^{2k-1} & \dots & n^{2k-1} \end{bmatrix}$$

Claim. $V : \mathbb{F}_p^n \to \mathbb{F}_p^{2k}$ is injective on k-sparse vectors

Let $x \neq y$ that are k-sparse and $Vx = Vy$

Sparse Recovery

$p > n$ a prime number

$$V = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ 1 & 2^2 & 3^2 & \dots & n^2 \\ \vdots & \vdots & \dots & \vdots & \\ 1 & 2^{2k-1} & 3^{2k-1} & \dots & n^{2k-1} \end{bmatrix}$$

Claim. $V : \mathbb{F}_p^n \to \mathbb{F}_p^{2k}$ is injective on k-sparse vectors

Let $x \neq y$ that are k-sparse and $Vx = Vy$
$V(x - y)$ is a linear combination of at most $2k$ columns of V.

Sparse Recovery

$p > n$ a prime number

$$V = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ 1 & 2^2 & 3^2 & \dots & n^2 \\ \vdots & \vdots & \dots & \vdots & \\ 1 & 2^{2k-1} & 3^{2k-1} & \dots & n^{2k-1} \end{bmatrix}$$

Claim. $V : \mathbb{F}_p^n \to \mathbb{F}_p^{2k}$ is injective on k-sparse vectors

Let $x \neq y$ that are k-sparse and $Vx = Vy$
$V(x - y)$ is a linear combination of at most $2k$ columns of V.
But any $2k$ cols of V are linearly independent... contradiction!

Sparse Recovery

$p > n$ a prime number

$$V = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ 1 & 2^2 & 3^2 & \dots & n^2 \\ \vdots & \vdots & \dots & \vdots & \\ 1 & 2^{2k-1} & 3^{2k-1} & \dots & n^{2k-1} \end{bmatrix}$$

Claim. $V : \mathbb{F}_p^n \to \mathbb{F}_p^{2k}$ is injective on k-sparse vectors

Let $x \neq y$ that are k-sparse and $Vx = Vy$
$V(x - y)$ is a linear combination of at most $2k$ columns of V.
But any $2k$ cols of V are linearly independent... contradiction!

Could be useful in CONGEST, CONGESTED CLIQUE, NCC, ...

Scheme For Δ-Coloring in Semi-Streaming (Simplified)

(Assadi, Kumar, Mittal, STOC 2022)

Before the streaming pass: sample random colors

During the streaming pass:

- compute the sparse-dense decomposition
- keep edges of the sparsified graph
- Compute $V 1_{N(v)}$ for all $v \in V$

After the streaming pass:

- Color sparse nodes, holey almost-cliques, expanding almost-cliques using palette sparsification
- In the hard cases, use sparse recovery

One Fewer Color

New Theorem. (FH, ICALP 2026)
There exists a **randomized one-pass semi-streaming** algorithm that given any graph with maximum degree $\Delta \geq \Delta_0$ and no Δ-clique, **outputs a $(\Delta - 1)$-coloring** w.h.p.

Even Fewer Colors?

New Theorem? (CCKM, STOC 2027 submission)
There exists a **randomized one-pass streaming** algorithm using $\widetilde{O}(nk)$ space that given any graph with maximum degree Δ **outputs a $(\Delta - k)$-coloring** w.h.p. when such colorings exist.

Chapitre 3: Meta-Algorithm for Δ-Coloring and $(\Delta - 1)$-Coloring

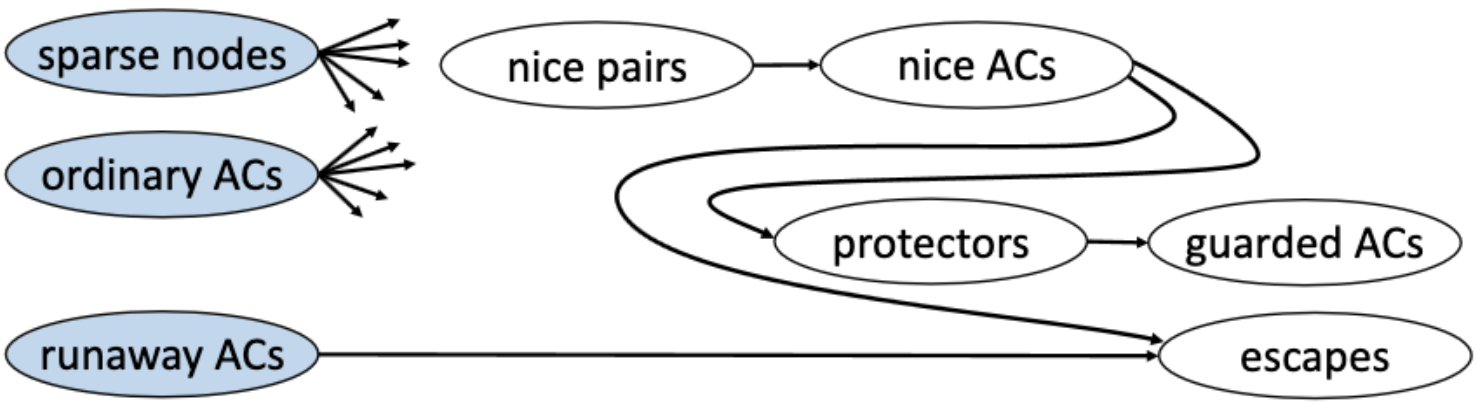
Type	*Holey*			*Unholey*		
	Friendly	*Social*	*Lonely*	*Friendly*	*Social*	*Lonely*
Small	Phase 4	Phases 2 or 4	Phase 2	Phase 6	Phases 2 or 6	Phase 2
Critical	Phase 4	Phase 4	Phase 4	Phase 5	Phase 5	Phase 5
Large	Phase 4	Phase 4	Phase 4	–	–	–

Table 1. A list of all combination of different almost-cliques together with the phase of our coloring procedure that is responsible for handling them. Note that by Observation 4.2, there are no unholey large almost-cliques. The sparse vertices are handled in Phase 1 and Phase 3. Moreover, Phase 1 may color some vertices of lonely (or social) small almost-cliques that we are *not* allowed to recolor (we can recolor all the other remnants of Phase 1 after Phase 3).

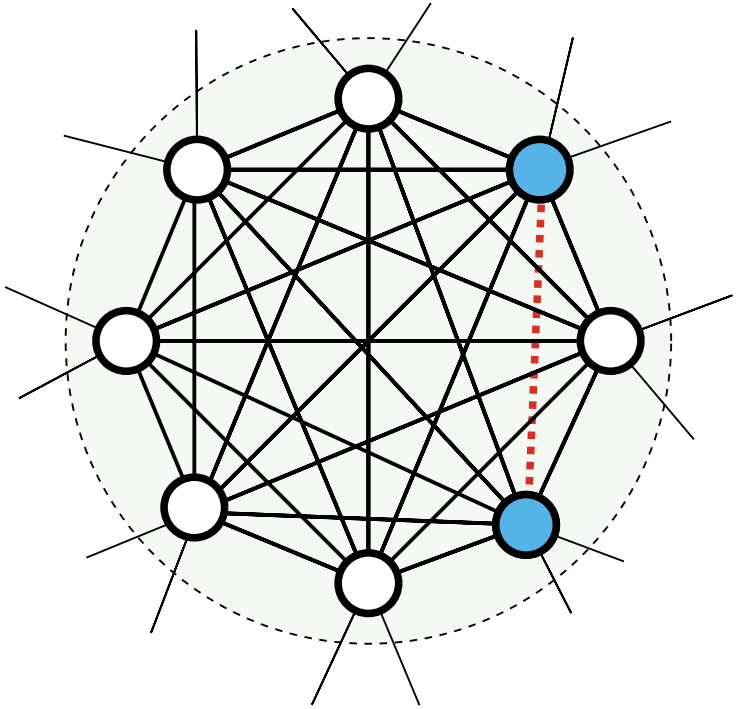
Meanwhile, Δ-Coloring in the LOCAL Model

Fast Distributed Brooks' Theorem

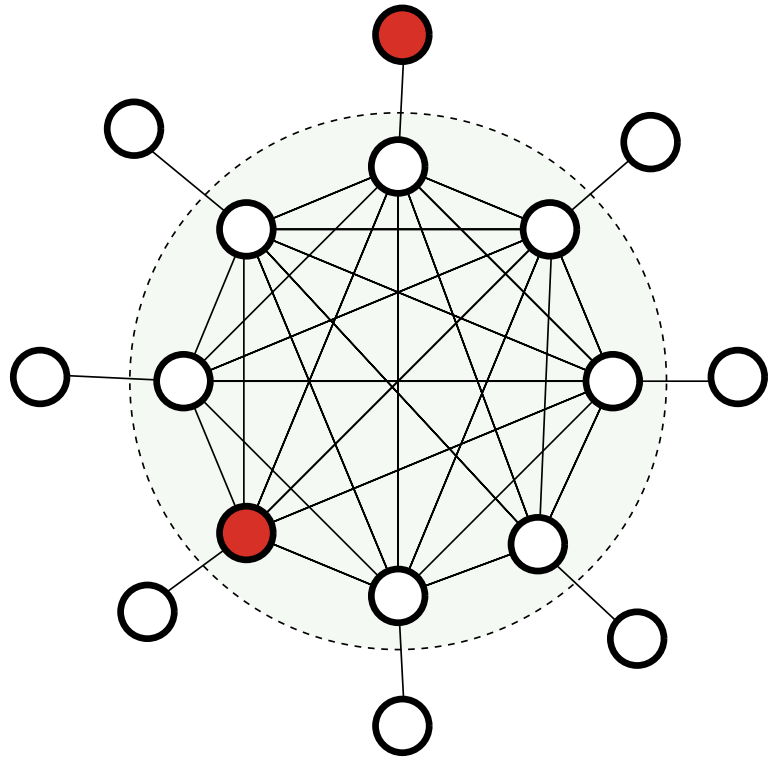
Manuela Fischer · Reykjavik University, Iceland
Magnús M. Halldórsson · Reykjavik University, Iceland
Yannic Maus · TU Graz, Austria



Meta-Algorithm for Δ-Coloring



Structural Arguments work

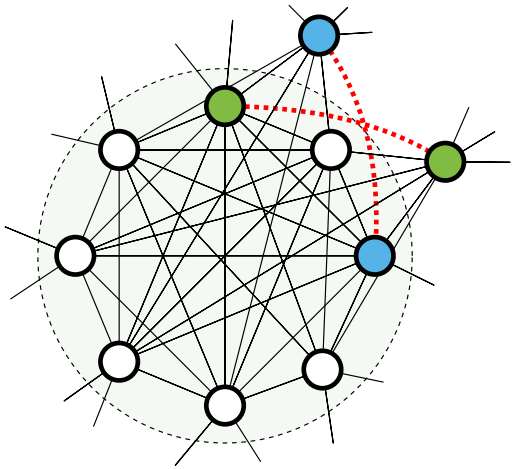


Randomness works

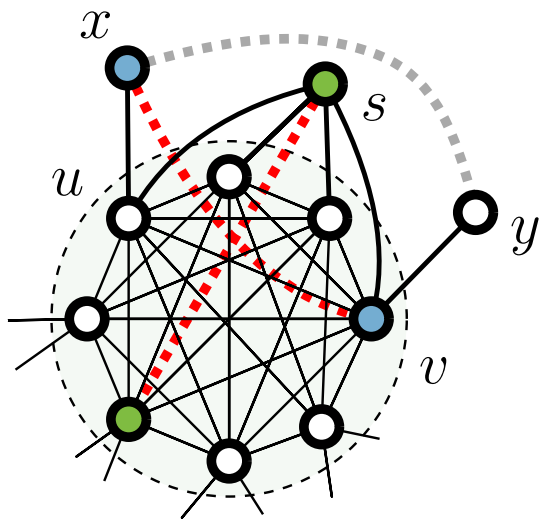
Challenges.

1. In LOCAL: symmetry breaking and LLL
2. In streaming: palette sparsification and sparse recovery

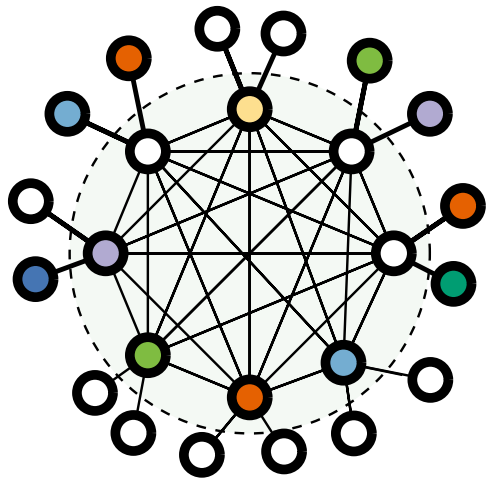
Meta-Algorithm for $(\Delta - 1)$-Coloring



Structural
Arguments work



Local
Transformations



Randomness works

Challenges.

1. In LOCAL: symmetry breaking and LLL
2. In streaming: palette sparsification and sparse recovery

Why is it **not** in the paper?

Two years ago, solving LLL in LOCAL was annoying...

Theorem. (**F**GHMN, 2026) There is an algorithm for solving LLLs with $p \leq d^{-1000}$ in $\mathrm{poly}(\log \log n)$ rounds of LOCAL.

What now?

How about a meta-algorithm for $(\Delta - k)$-coloring?

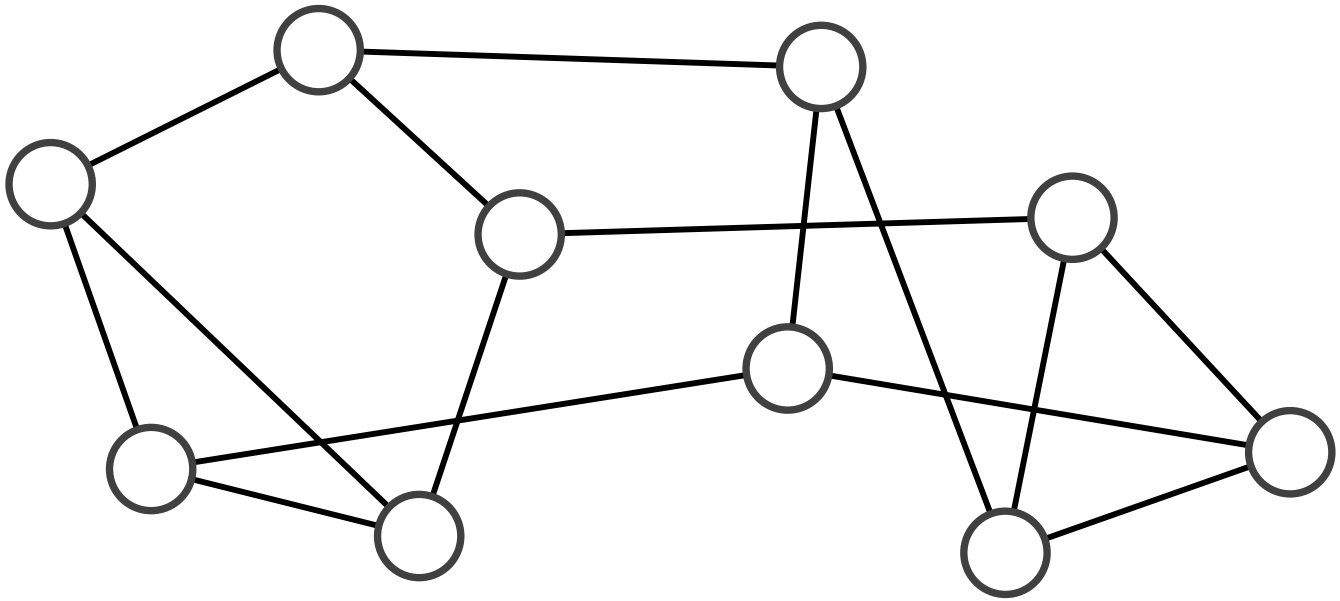
- we have a $\mathrm{poly}(\log \log n)$-round LOCAL algorithm (**F**HJM, STOC 2026)
- we seem to have a streaming algorithm?

Question. Are they related?

C’est la Fin

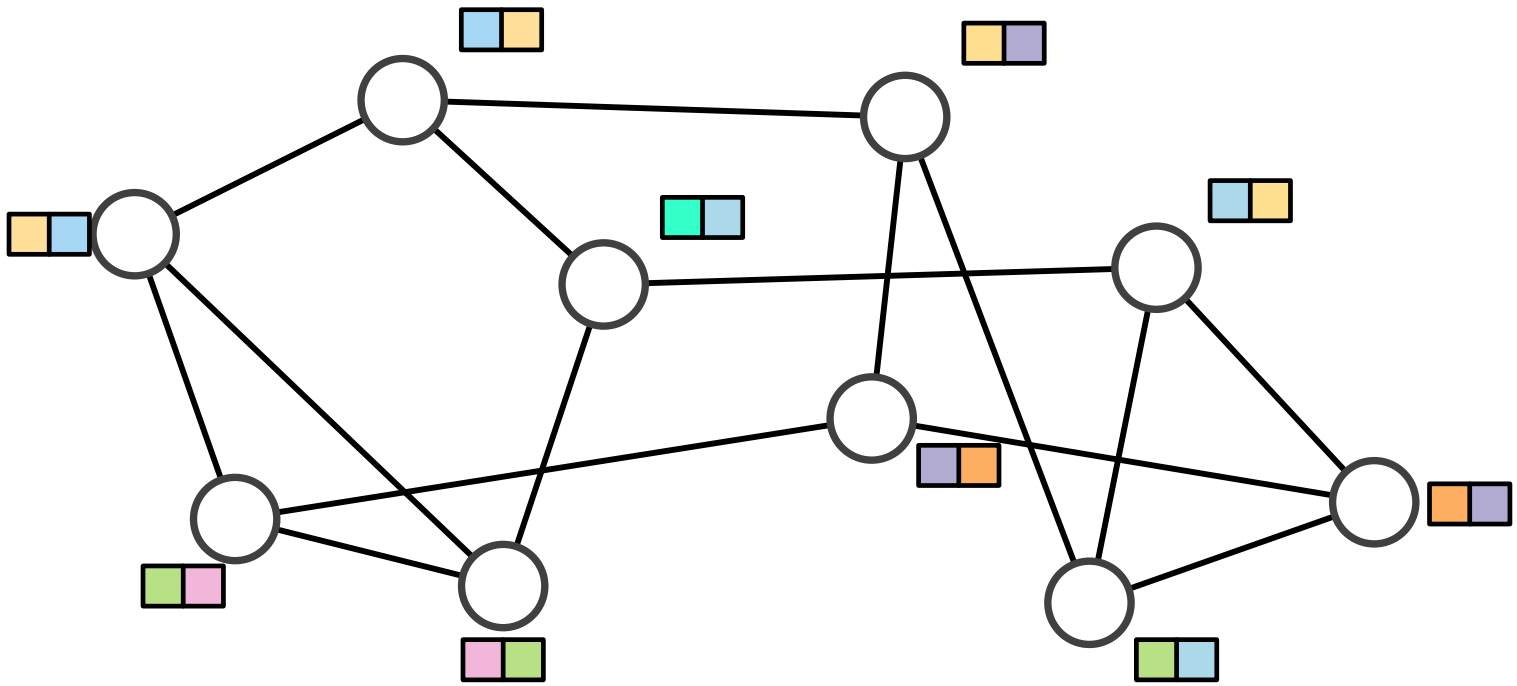
This is the End

Takeaway 1: Palette Sparsification



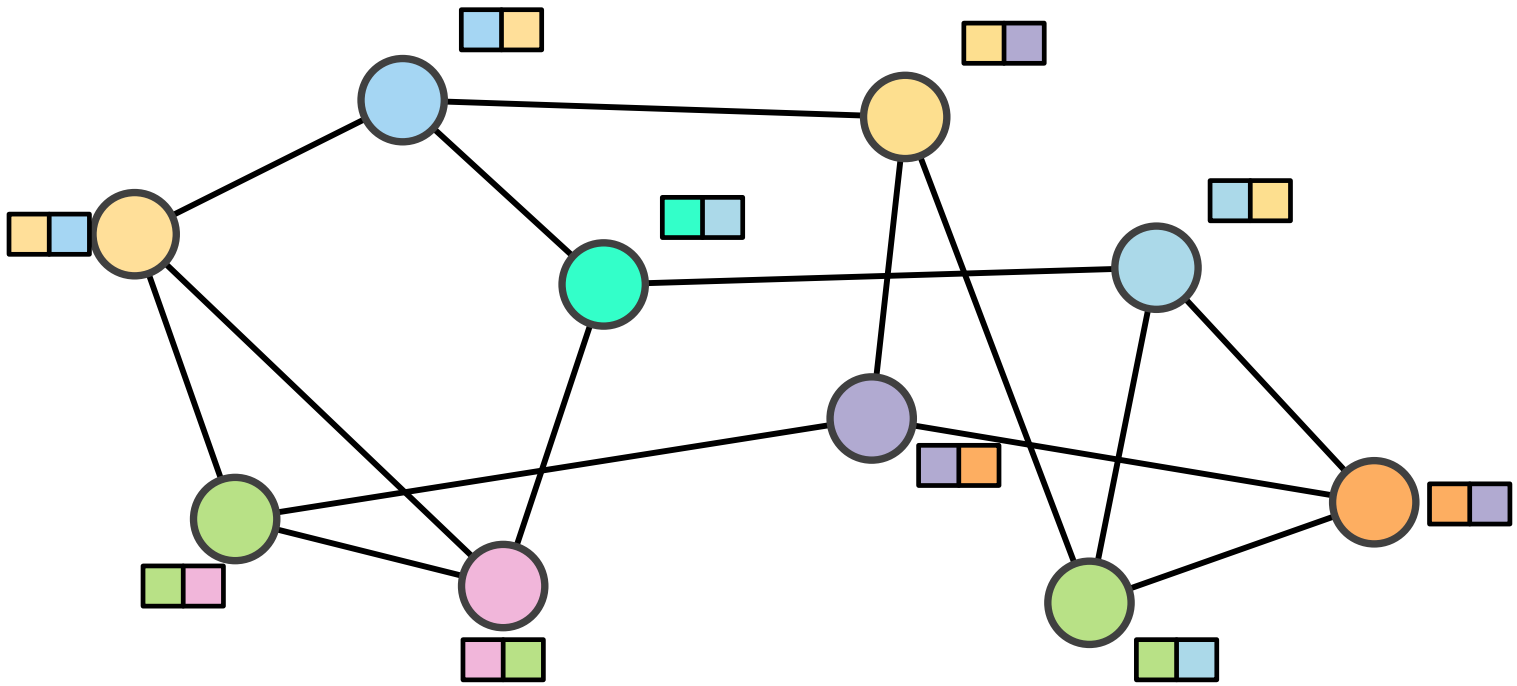
Palette Sparsification Theorem. (ACK19)
Independently sample $O(\log n)$ random colors in $[\Delta + 1]$ for every vertex; whp, there exists a coloring where each vertex uses one of its random colors.

Takeaway 1: Palette Sparsification



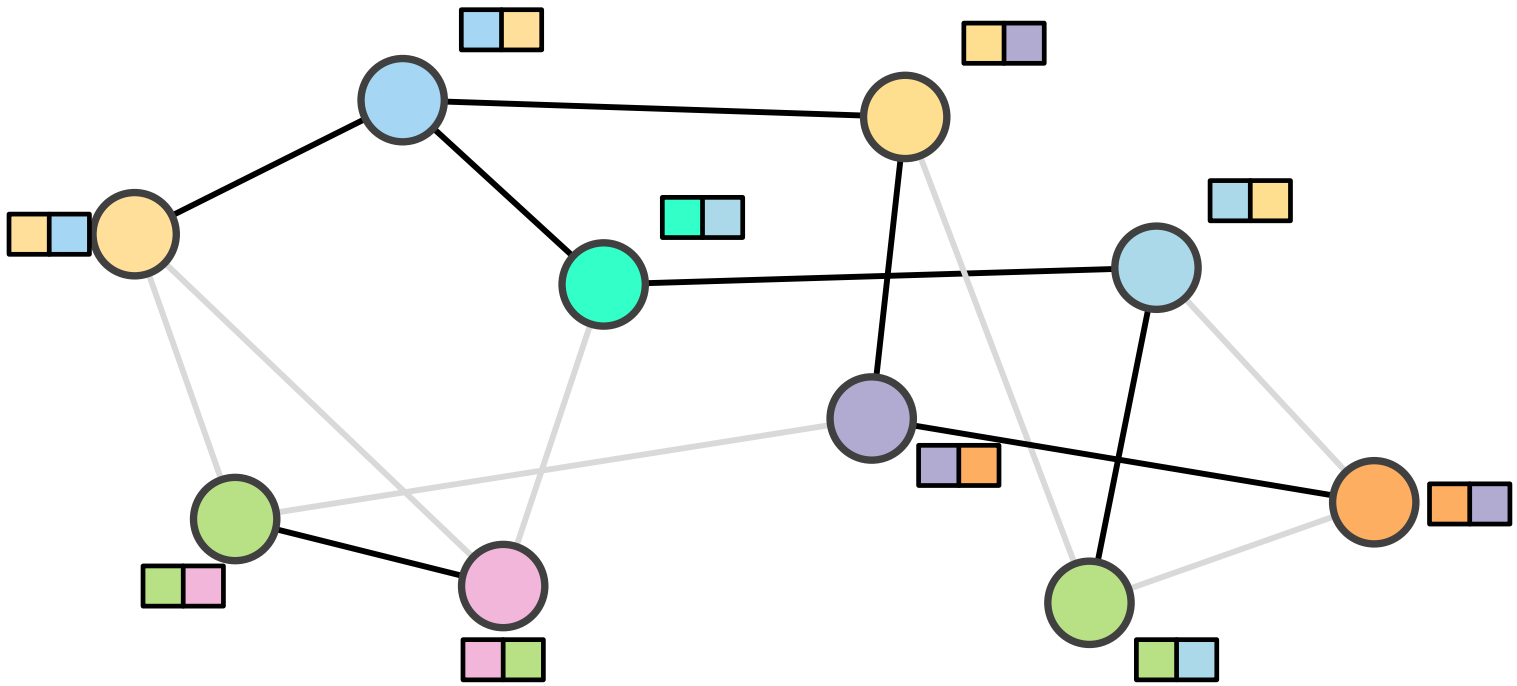
Palette Sparsification Theorem. (ACK19)
Independently sample $O(\log n)$ random colors in $[\Delta + 1]$ for every vertex; whp, there exists a coloring where each vertex uses one of its random colors.

Takeaway 1: Palette Sparsification



Palette Sparsification Theorem. (ACK19)
Independently sample $O(\log n)$ random colors in $[\Delta + 1]$ for every vertex; whp, there exists a coloring where each vertex uses one of its random colors.

Takeaway 1: Palette Sparsification



Palette Sparsification Theorem. (ACK19)
Independently sample $O(\log n)$ random colors in $[\Delta + 1]$ for every vertex; whp, there exists a coloring where each vertex uses one of its random colors.

Takeaway 2: Sparse Recovery

$$V = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ 1 & 2^2 & 3^2 & \dots & n^2 \\ \vdots & \vdots & \dots & \vdots & \\ 1 & 2^{2s-1} & 3^{2s-1} & \dots & n^{2s-1} \end{bmatrix}$$

Claim. $V : \mathbb{F}_p^n \to \mathbb{F}_p^{2s}$ is injective on s-sparse vectors

Could be useful in CONGEST, CONGESTED CLIQUE, NCC, ...

Takeaway 3: Meta-Coloring Algorithm

Δ- and $(\Delta - 1)$-Coloring can be attacked similarly in LOCAL and streaming.

How about a meta-algorithm for $(\Delta - k)$-coloring?

Challenges.

1. In LOCAL: symmetry breaking and LLL
2. In streaming: palette sparsification and sparse recovery
3. other models?

Is coloring special? what about other problems?

Takeaway 3: Meta-Coloring Algorithm

Δ- and $(\Delta - 1)$-Coloring can be attacked similarly in LOCAL and streaming.

How about a meta-algorithm for $(\Delta - k)$-coloring?

Challenges.

1. In LOCAL: symmetry breaking and LLL
2. In streaming: palette sparsification and sparse recovery
3. other models?

Is coloring special? what about other problems?

Thanks!