Dynamic Low Stretch Spanning Trees in Sub-polynomial Time

Shiri Chechik, Tianyi Zhang

Tel Aviv University, Tsinghua University

Sparsification

Approximate dense objects using sparse objects

masonry arch

truss arch

This example is taken from www.cosy.sbg.ac.at/~sk/talks/Salzburg2017.pdf

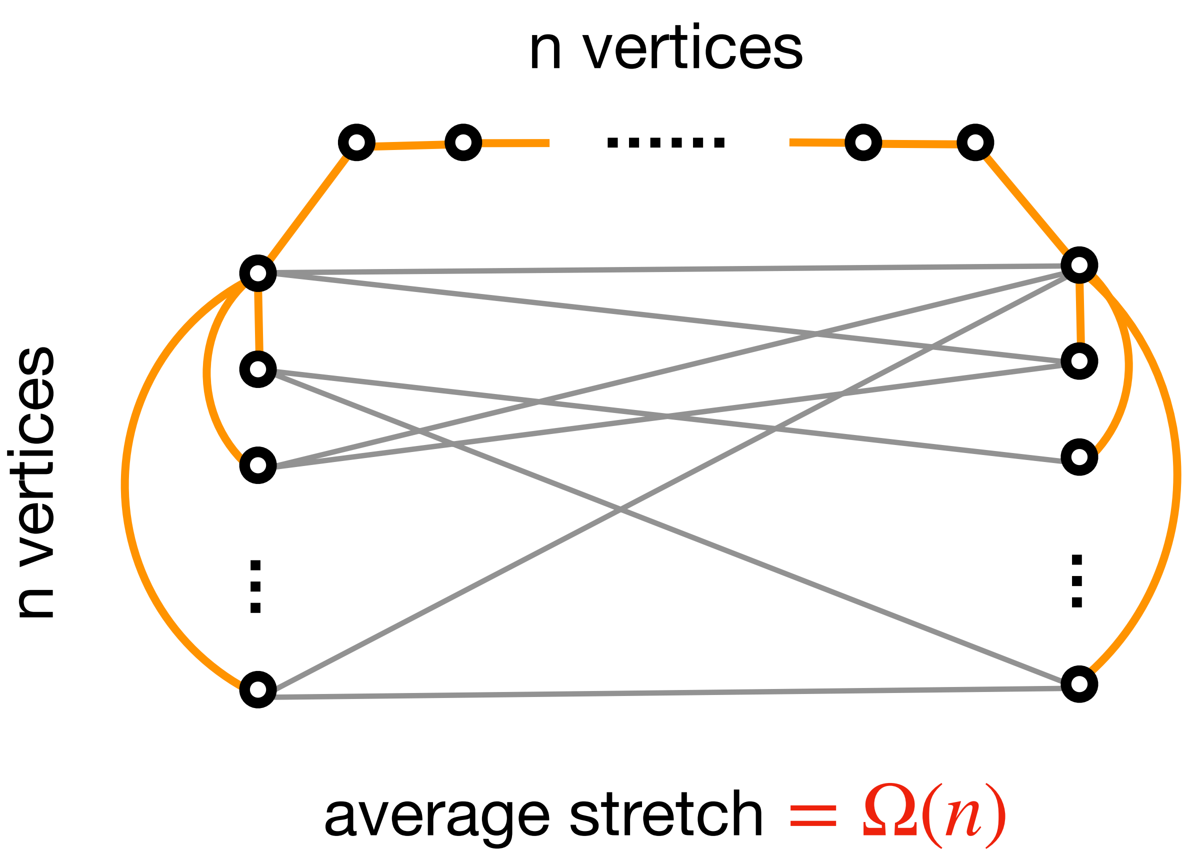
Graph sparsification

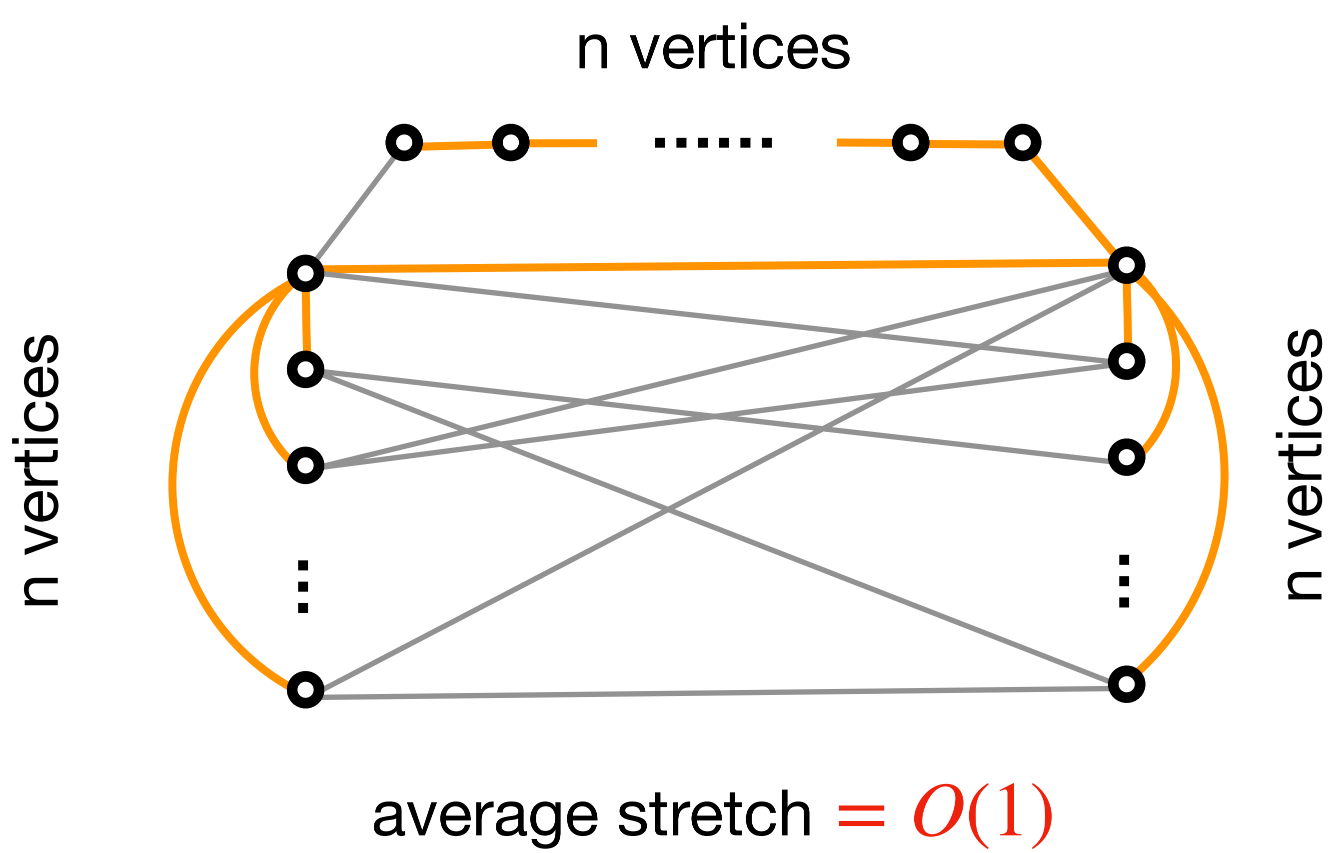
- Let $G = (V, E)$ be an undirected (multi-)graph
- Want to reduce number of edges while preserving certain graph properties
- <u>Example:</u>
  (Spanners) Every graph has a subgraph with $O(n^{1.5})$ edges that 3-approximates pairwise distances
- What if we want to sparsify G as a **spanning tree**?

Definition: low-stretch spanning trees

- Let T be a **spanning tree** of graph G
- Want **low average stretch** of T:

$$\text{stretch of } T = \frac{1}{|E|} \sum_{(u,v) \in E} \text{dist}_T(u, v)$$





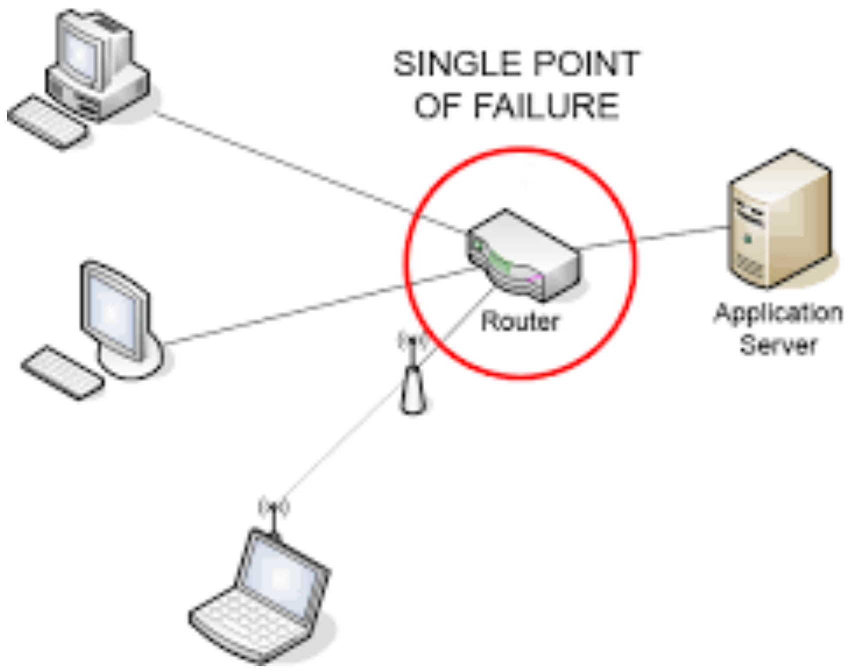
History of low-stretch trees

Reference	Average stretch	Construction time
[Alon+'95]	$\Omega(\log n)$	
[Alon+'95]	$2^{O(\sqrt{\log n \cdot \log\log n})}$	$\tilde{O}(m)$
[Elk+'08]	$O(\log^2 n \log\log n)$	$\tilde{O}(m)$
[ABN'09]	$O(\log n \cdot \log\log n \cdot (\log\log\log n)^3)$	$\tilde{O}(m)$
[AN'12]	$O(\log n \cdot \log\log n)$	$\tilde{O}(m)$

n = # vertices, m = # edges

The dynamic setting

Our world keeps changing



change of network topologies

change of traffic conditions

Definition: **dynamic** low-stretch trees

- Graph G suffers a sequence of edge insertions / deletions
- Want to maintain a spanning tree
    - Low average stretch, say $n^{o(1)}$
    - Fast update time, ideally $\log^{O(1)} n$
- Raised in [BKS'12]; first studied in [FG'19]

Our results

- [FG'19]
  Average stretch: $n^{o(1)}$
  Randomized update time: $n^{1/2+o(1)}$

- [C**Z**'20]
  Average stretch: $n^{o(1)}$
  Randomized update time: $n^{o(1)}$

- Extend to **weighted** graphs but only with edge **deletions**

- In this talk, we only focus on **unweighted** graphs

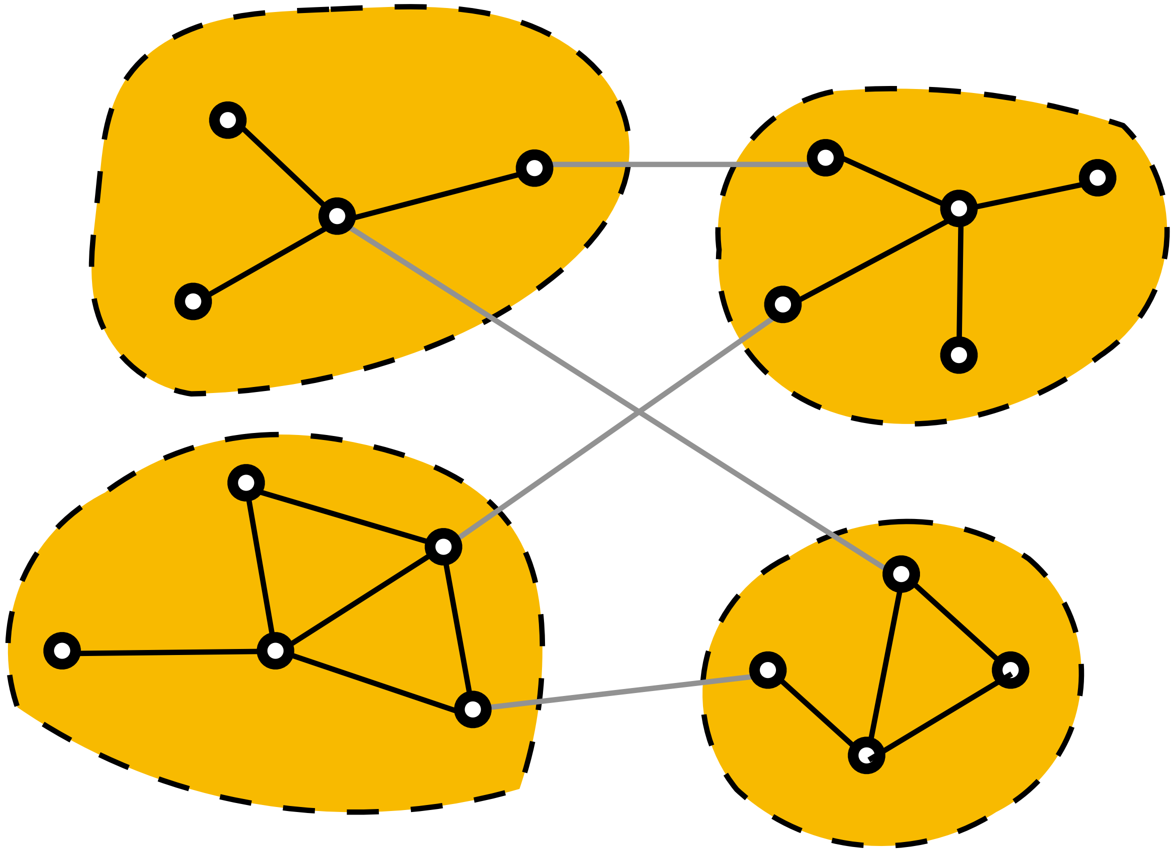
An overview of [Alon+’95]

Low diameter decomposition

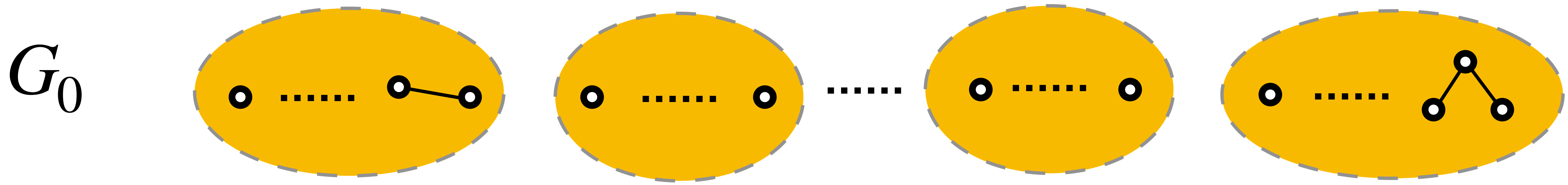
Definition: A β-decomposition is a partition of V into clusters $C_1, C_2, \cdots, C_k$ such that:

(1) diameter of each $G[C_i]$ is $\leq O(\frac{\log n}{\beta})$

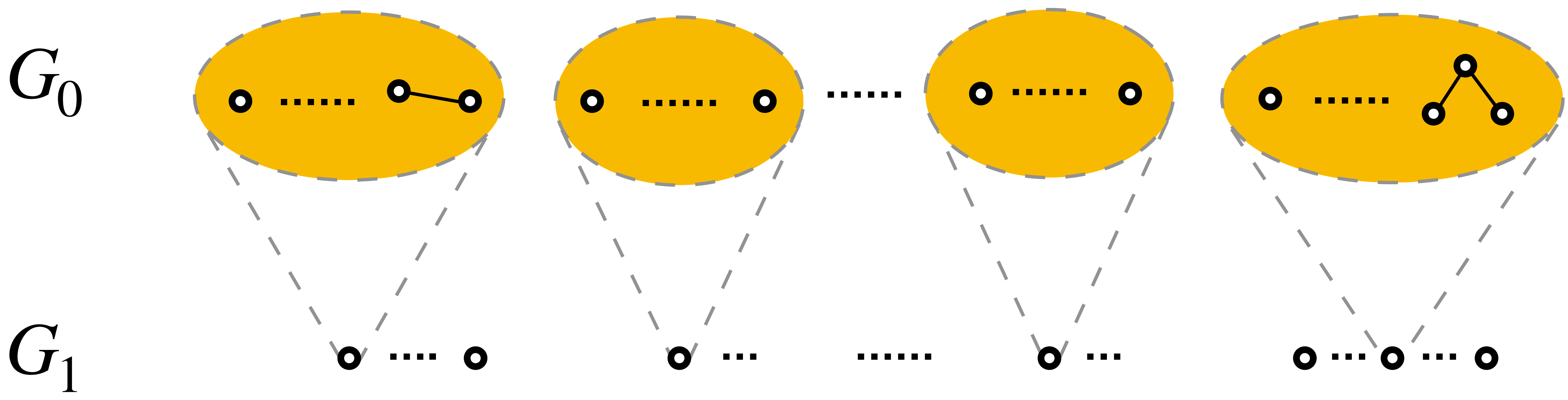
(2) number of inter-cluster edges is $\leq \beta m$



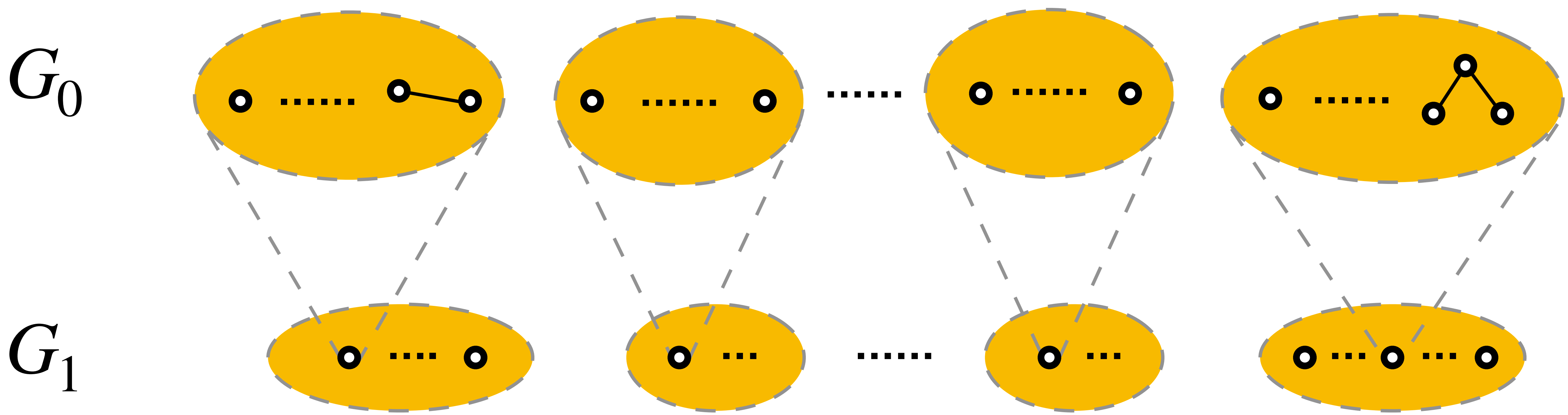
Hierarchical clustering



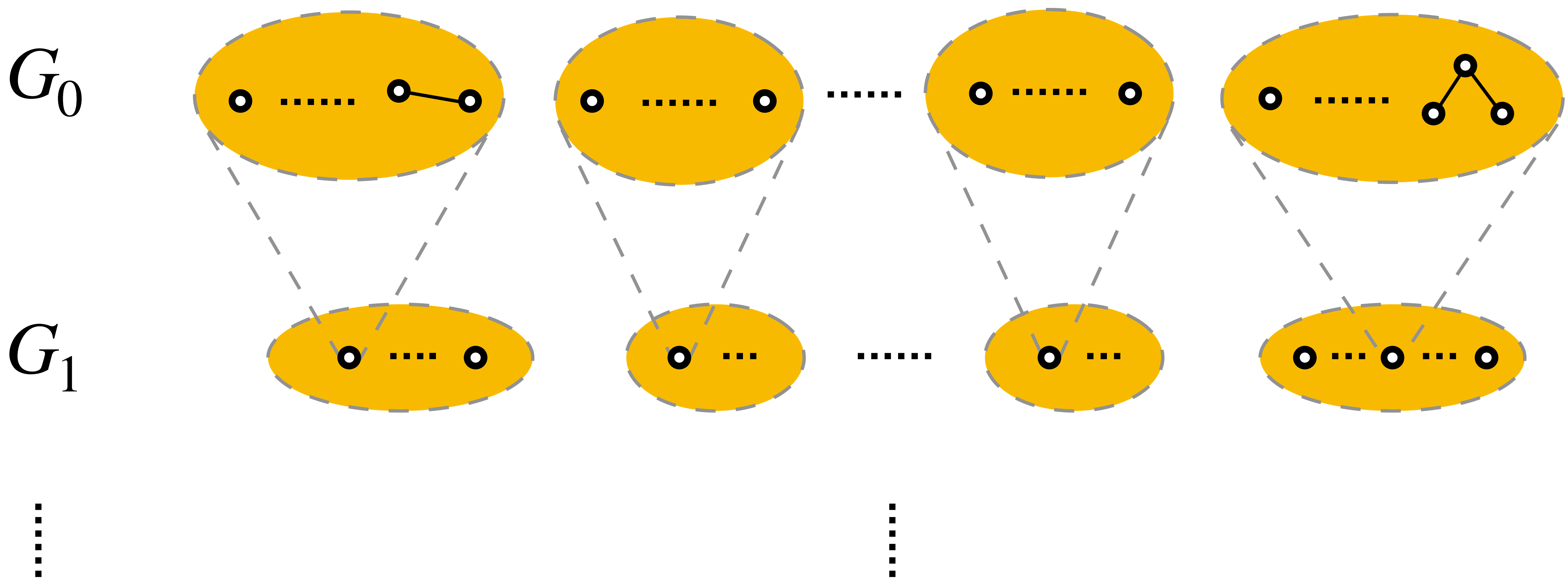
Hierarchical clustering



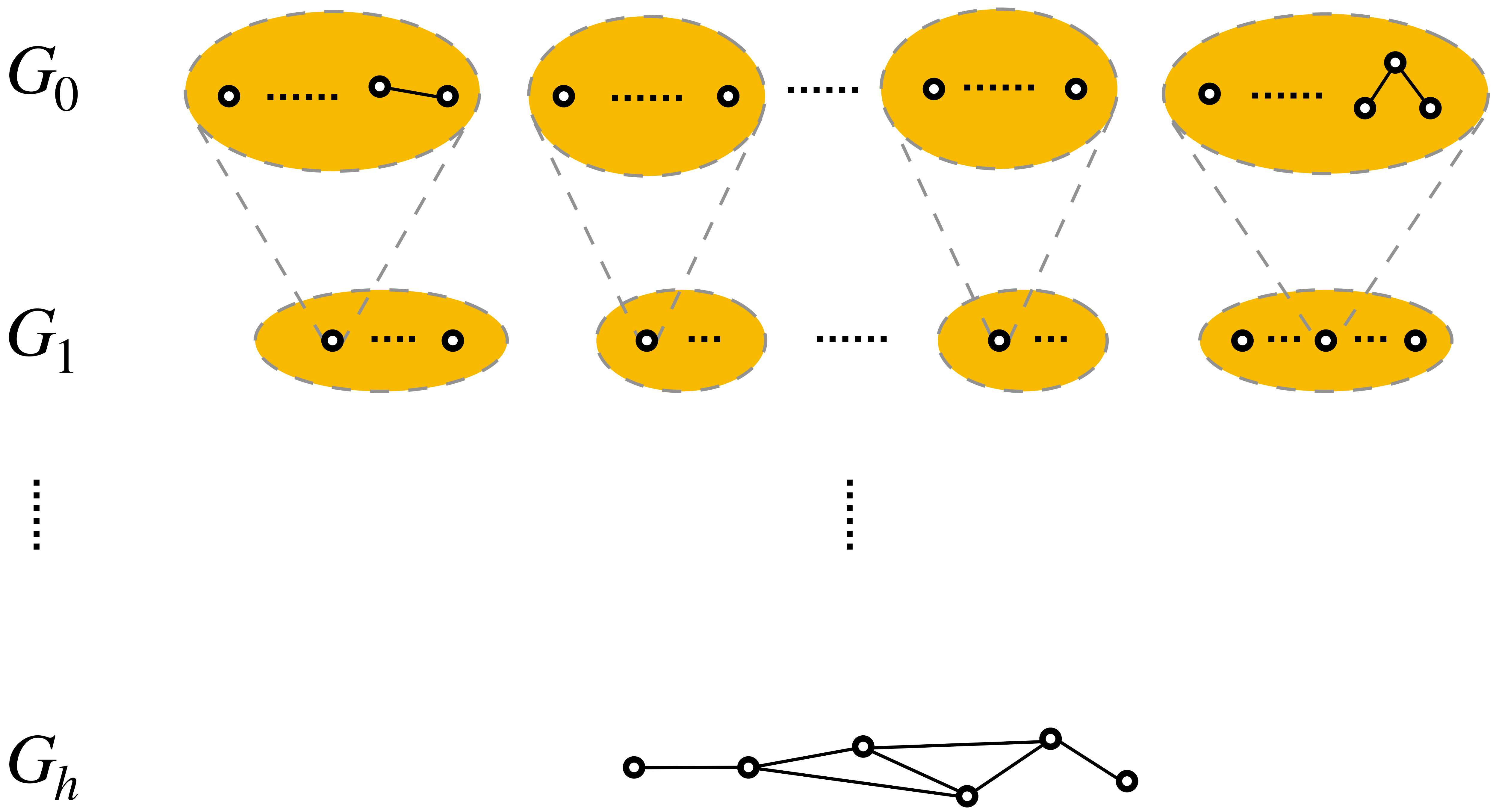
Hierarchical clustering



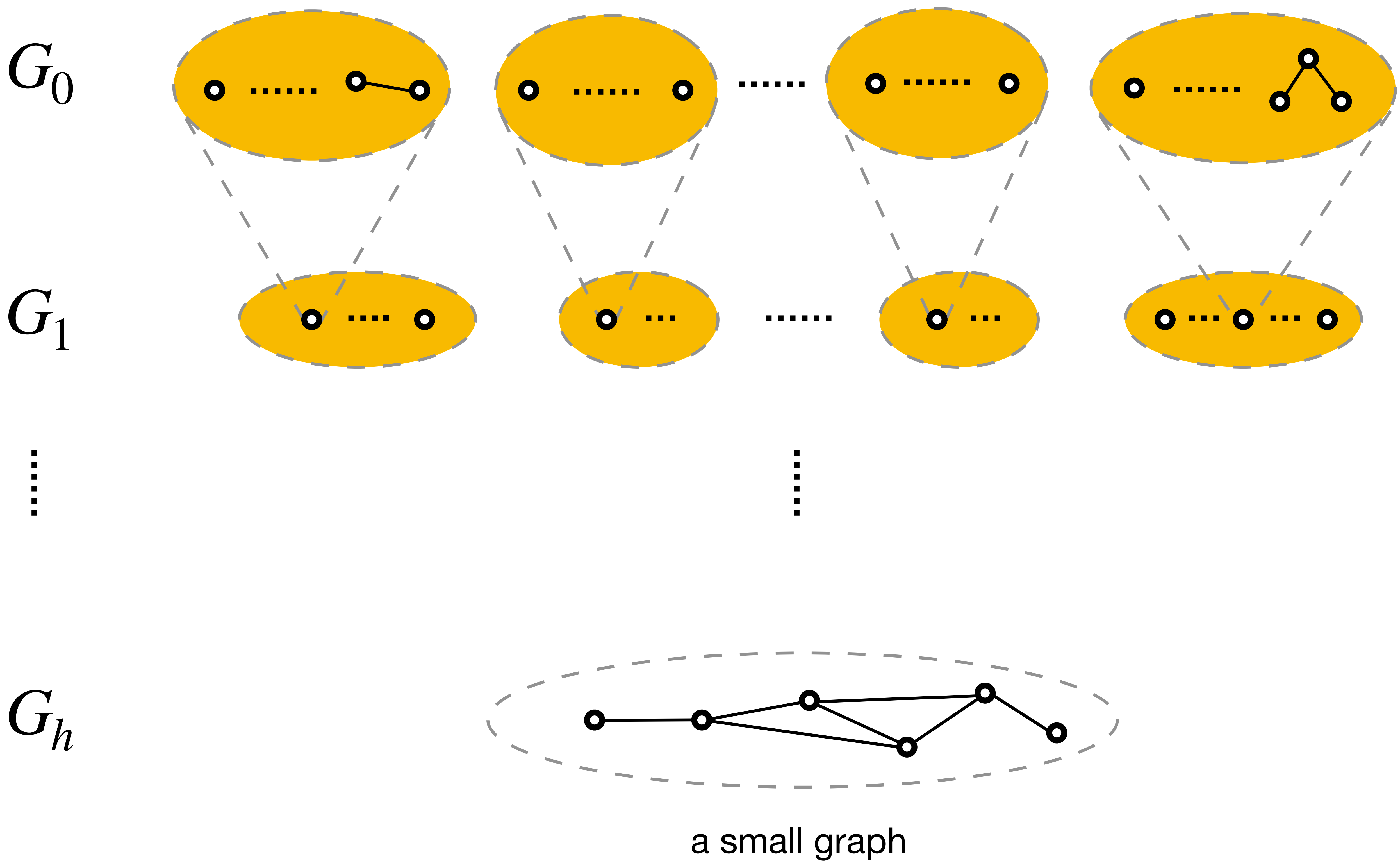
Hierarchical clustering



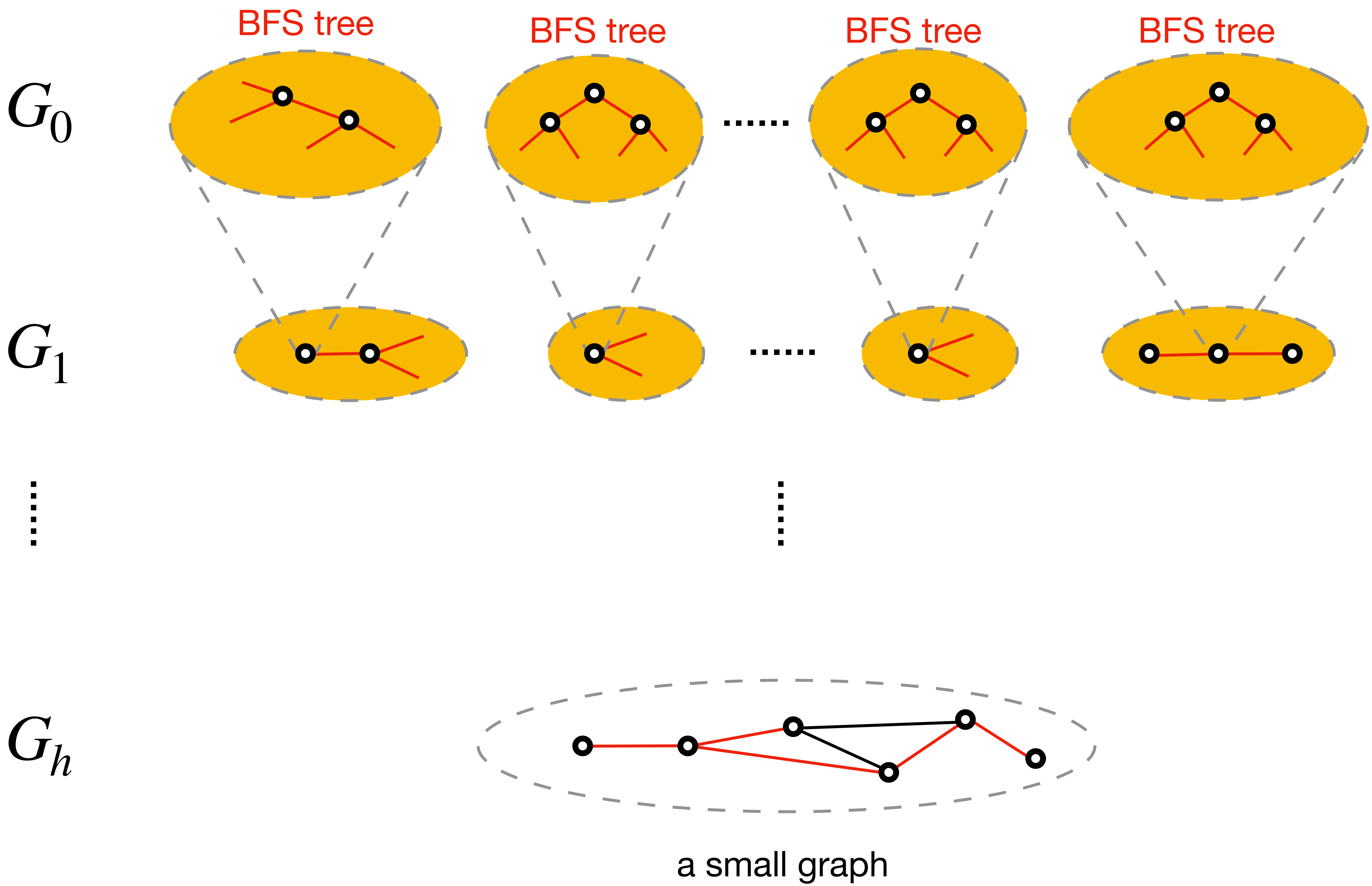
Hierarchical clustering



Hierarchical clustering



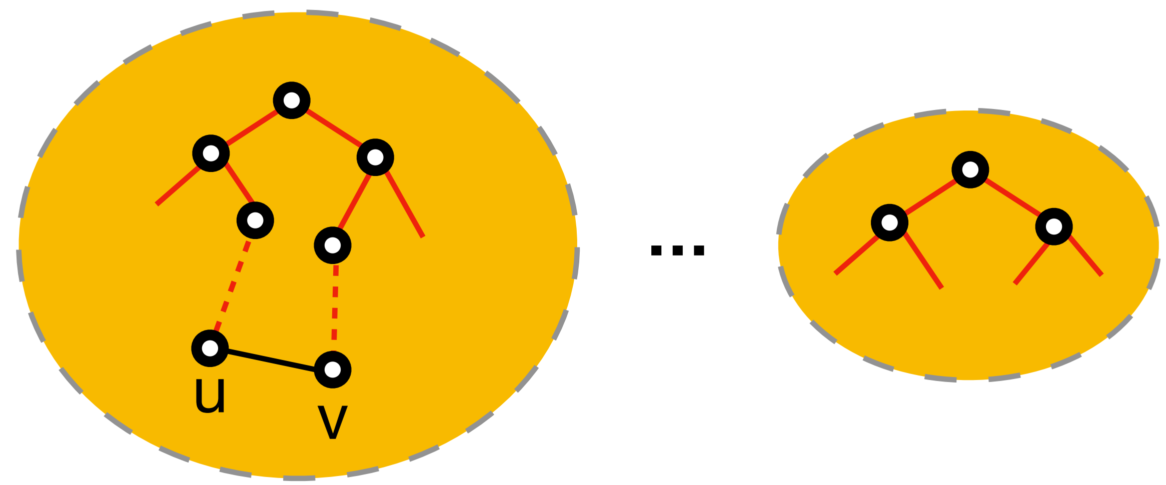
Low-stretch spanning tree



Average stretch

contracted graph G_i

- Diameter of node in G_i is at most $(\log n/\beta)^i$
- Number of edges in G_i is at most $\beta^i m$
- Stretch of any intra-cluster edge in $G_i \backslash G_{i+1}$ is $(\log n/\beta)^{i+1}$
- Total stretch becomes $m^{1+o(1)}$



$\mathbf{dist}_T(u,v) \le (\log n/\beta)^{i+1}$

$$\sum_{(u,v)\in G_i \backslash G_{i+1}} \mathbf{dist}_T(u,v) \le m \log^{i+1} n/\beta$$

$$\sum_{0\le i\le h} \sum_{(u,v)\in G_i \backslash G_{i+1}} \mathbf{dist}_T(u,v) \le m \log^{h+1} n/\beta$$

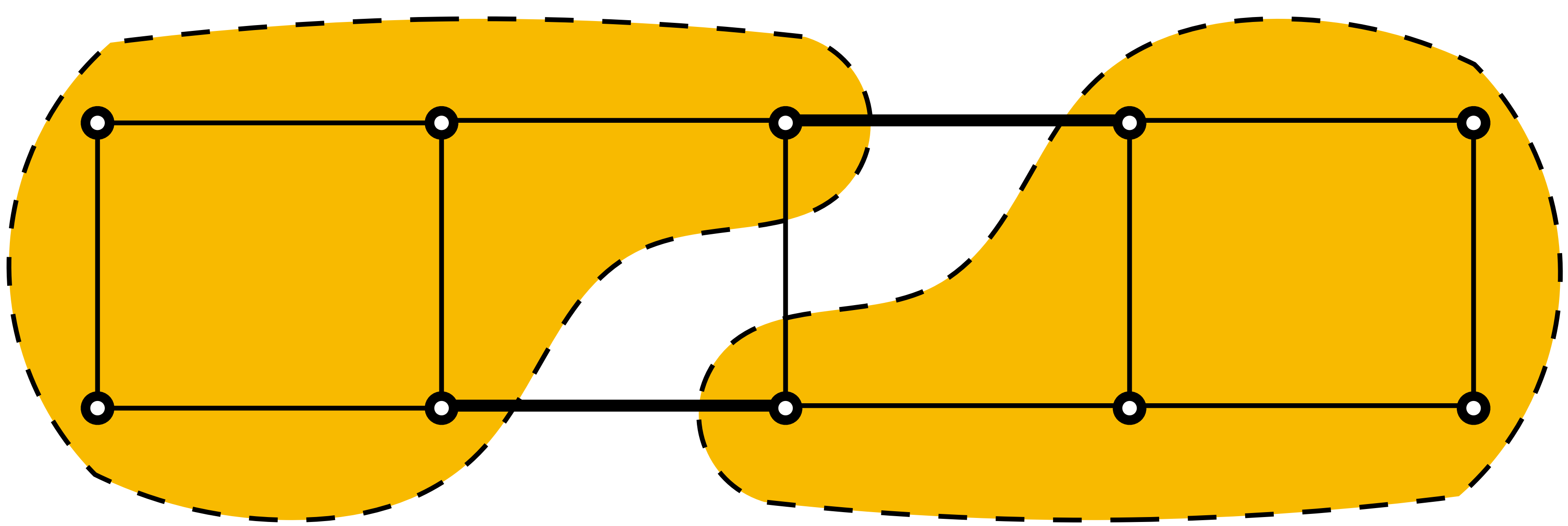
$$\le m^{1+o(1)}$$

$$h = \sqrt{\log n}, \beta = m^{-1/h}$$

An overview of [FG'19]

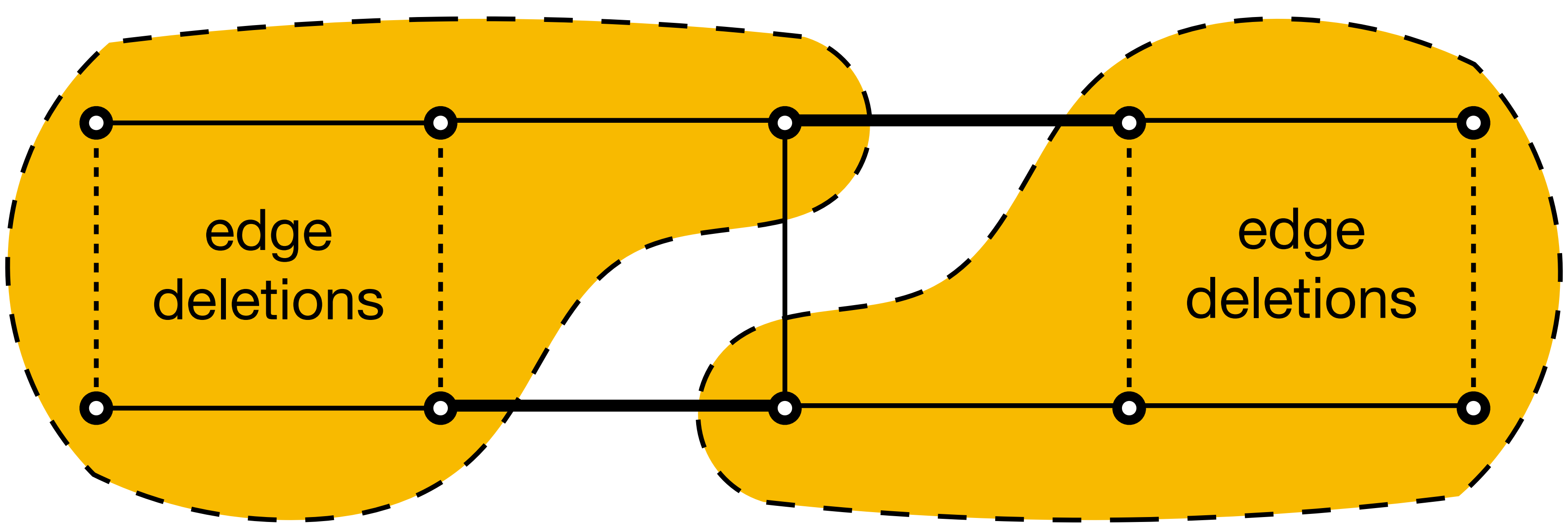
Decremental low-diameter decomposition

- Lemma: [FG'19]
  A β-decomposition can be maintained under edge deletions such that:
  (1) total update time is $\tilde{O}(m/\beta)$
  (2) total number of **changes to inter-cluster** edges is $\tilde{O}(m)$



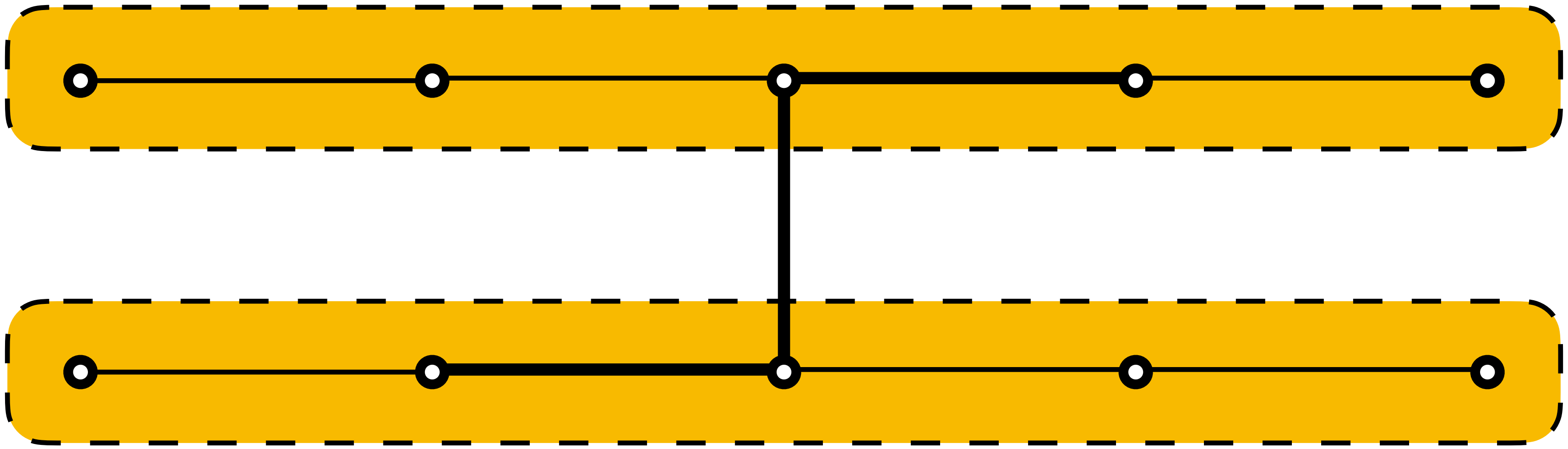
Decremental low-diameter decomposition

- Lemma: [FG'19]
  A β-decomposition can be maintained under edge deletions such that:
  (1) total update time is $\tilde{O}(m/\beta)$
  (2) total number of **changes to inter-cluster** edges is $\tilde{O}(m)$



Decremental low-diameter decomposition

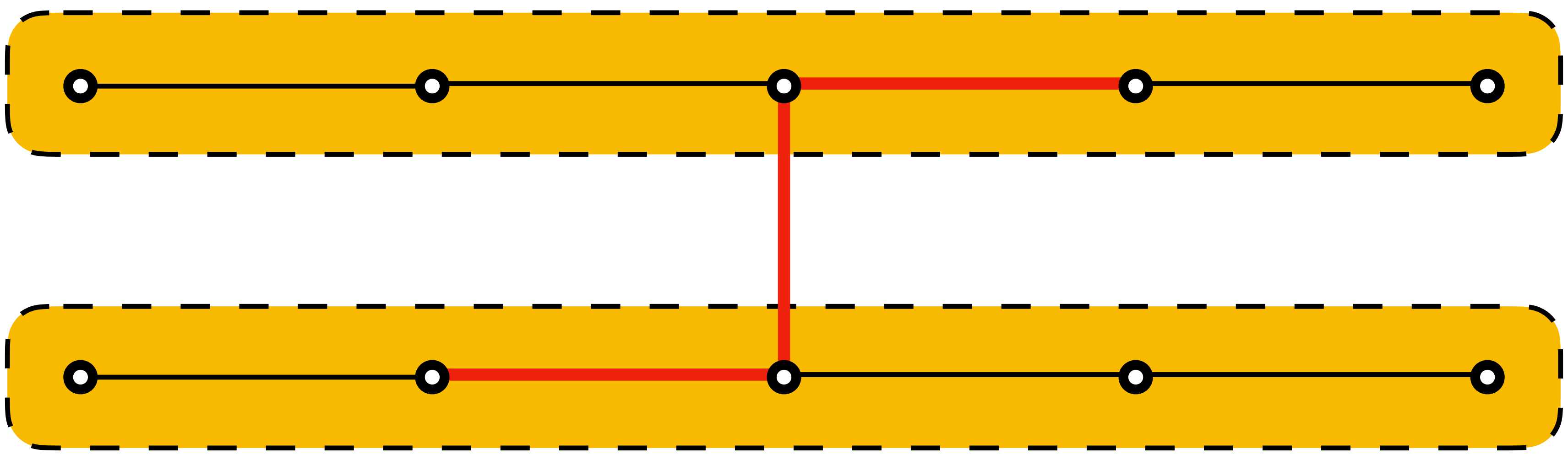
- Lemma: [FG'19]
  A β-decomposition can be maintained under edge deletions such that:
  (1) total update time is $\tilde{O}(m/\beta)$
  (2) total number of **changes to inter-cluster** edges is $\tilde{O}(m)$



update the β-decomposition

Decremental low-diameter decomposition

- Lemma: [FG'19]
  A β-decomposition can be maintained under edge deletions such that:
  (1) total update time is $\tilde{O}(m/\beta)$
  (2) total number of **changes to inter-cluster** edges is $\tilde{O}(m)$



total #inter-cluster edges = 3

Fully dynamic low-diameter decomposition

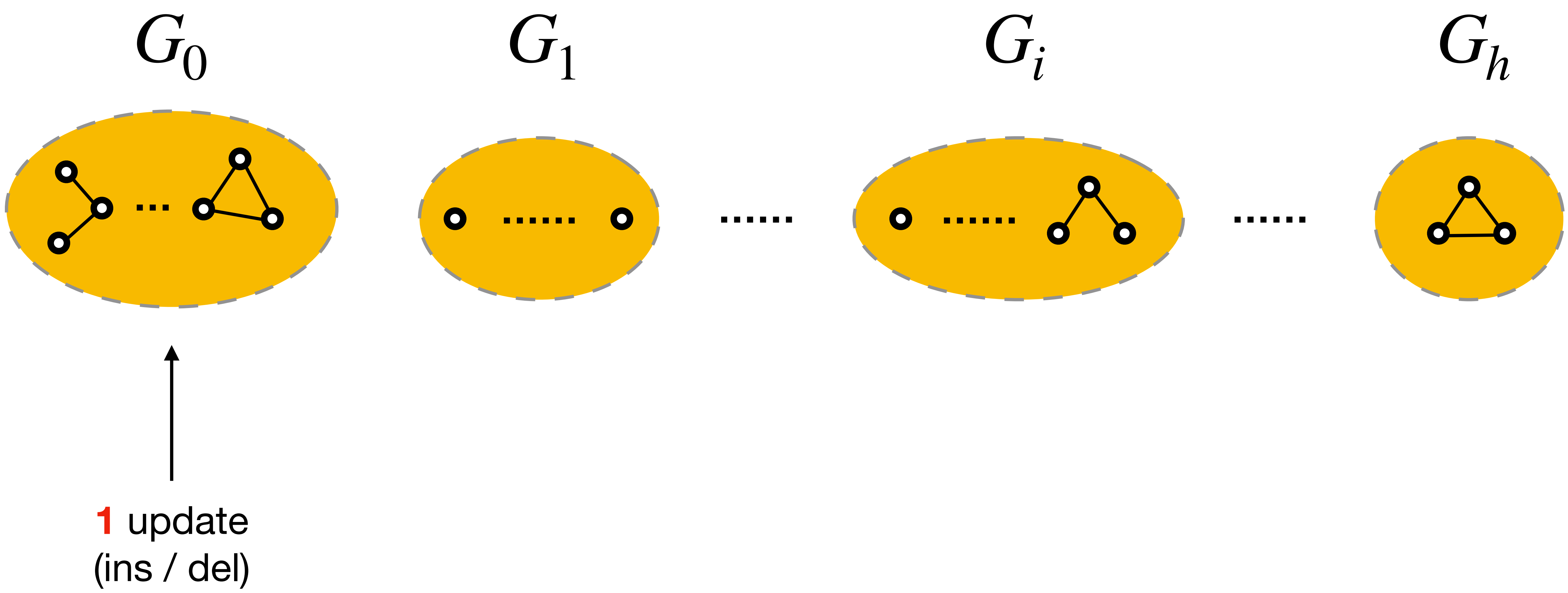
- Lemma: [FG'19]
  A β-decomposition can be maintained under edge updates such that:
  (1) amortized update time is $\tilde{O}(1/\beta^2)$
  (2) amortized **changes to inter-cluster** edges is $\tilde{O}(1/\beta)$

- Handle edge insertions **lazily**

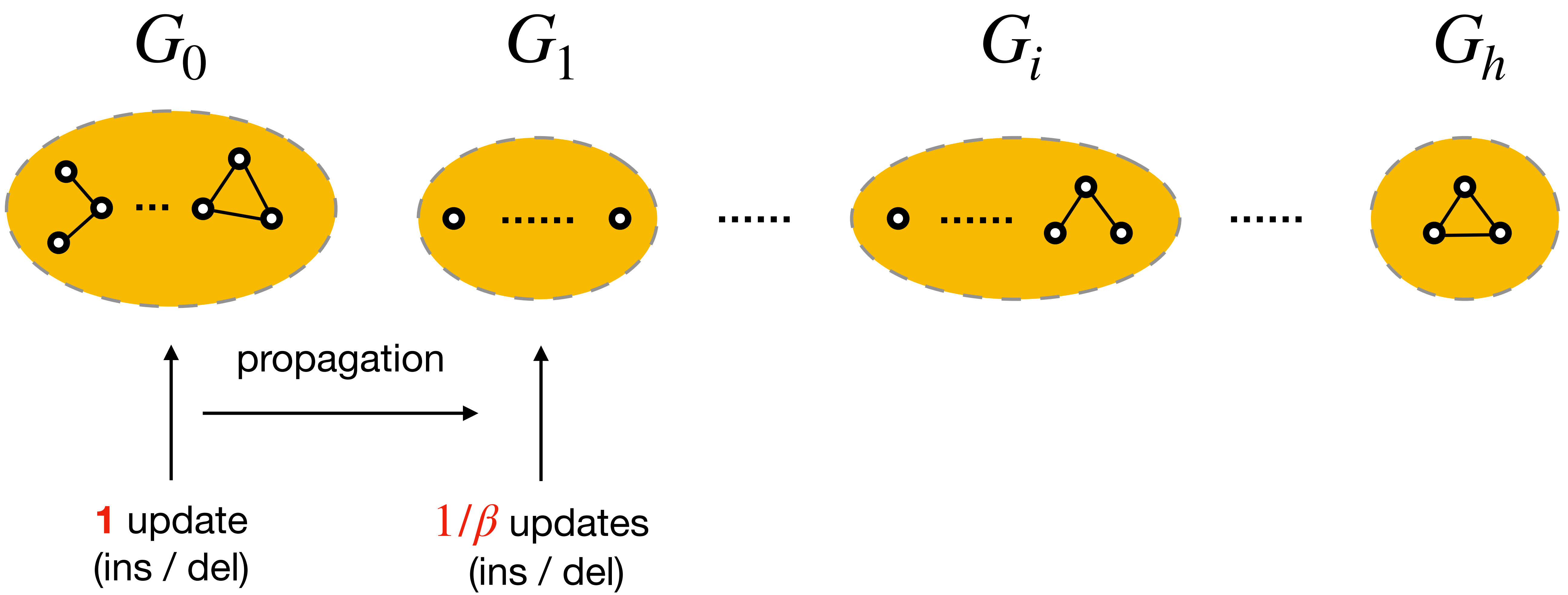
  Apply decremental β-decomposition, and ignore all edge insertions, and **rebuild after every** βm edge updates

- **Amortized changes** $\approx \dfrac{\text{total changes}}{\text{total updates}} = \dfrac{m}{\beta m} = 1/\beta$

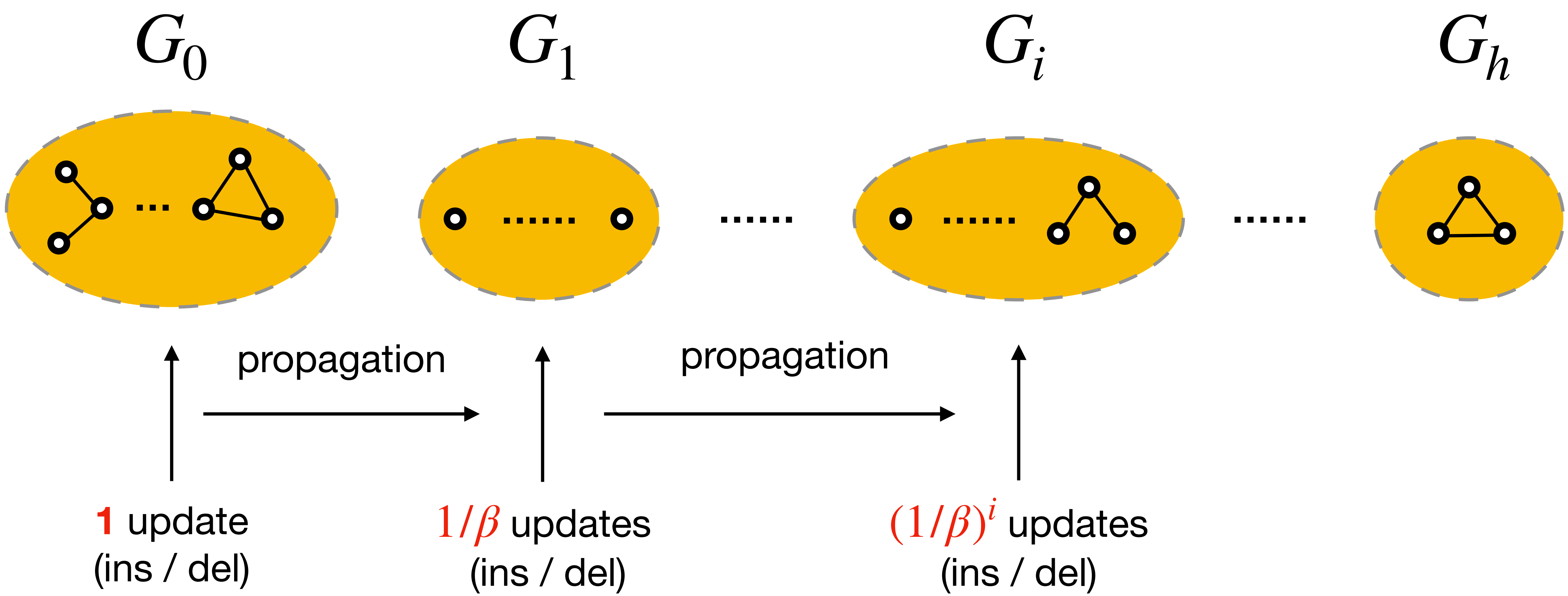
Total update time of hierarchical clustering



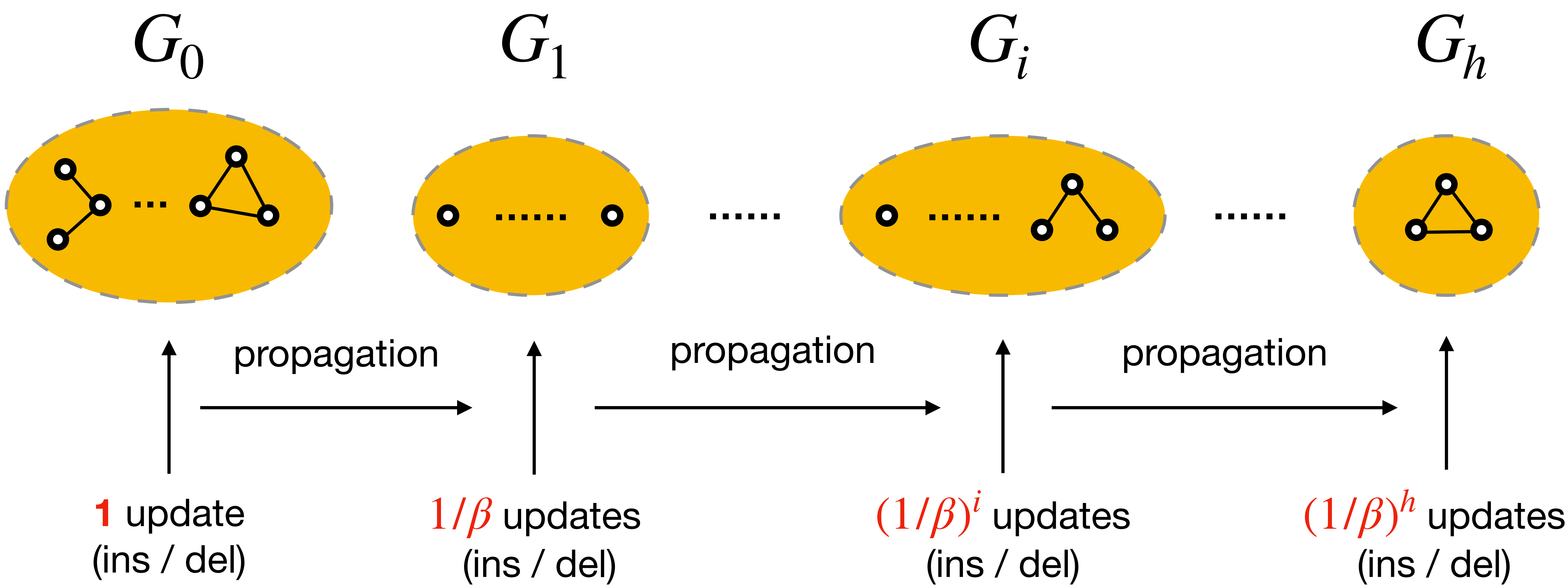
Total update time of hierarchical clustering



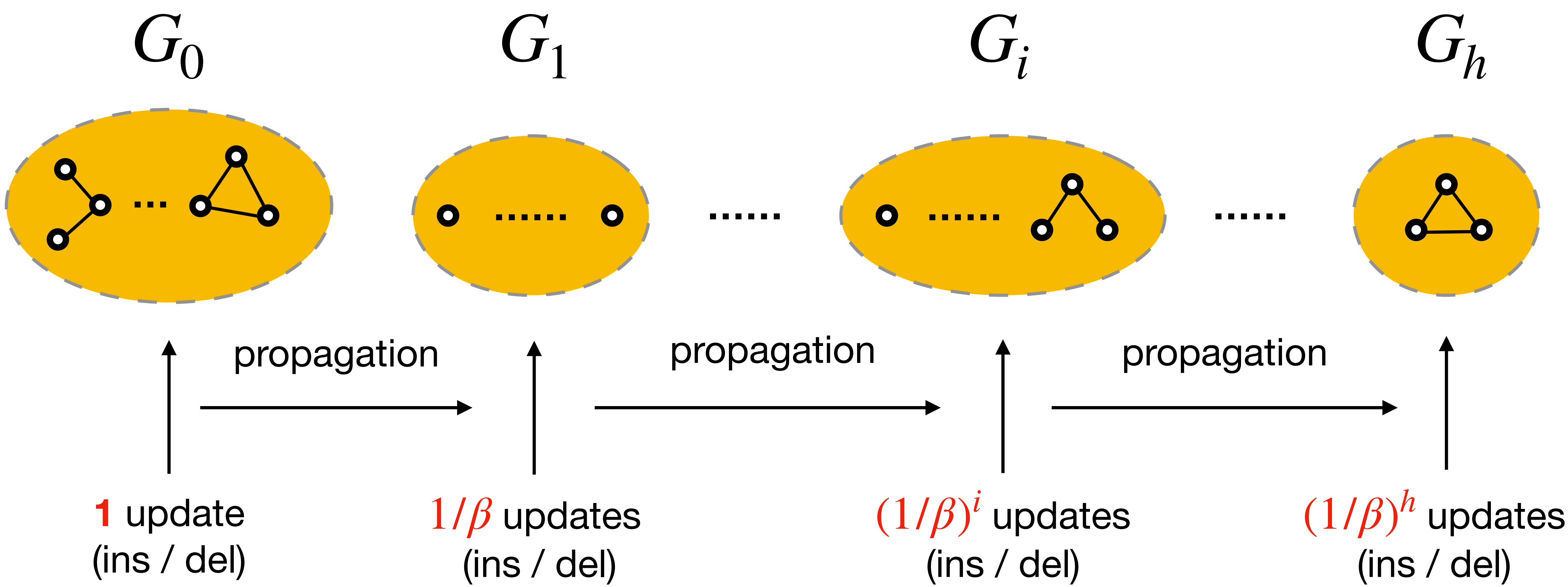
Total update time of hierarchical clustering



Total update time of hierarchical clustering



Total update time of hierarchical clustering



- Overall update time $\approx (1/\beta)^{h+2} + m\beta^h = m^{0.5+o(1)}$
- Graph sparsification can improve $m^{0.5+o(1)}$ to $n^{0.5+o(1)}$

Our improvement

Reducing total changes to inter-cluster edges

- Lemma: [FG'19]
  A β-decomposition can be maintained under edge deletions such that:
  (1) total update time is $\tilde{O}(m/\beta)$
  (2) total number of **changes to inter-cluster** is $\tilde{O}(m)$

- Lemma:
  A β-decomposition can be maintained under edge deletions such that:
  (1) total update time is $\tilde{O}(m/\beta)$
  (2) total number of **changes to inter-cluster** is $\tilde{O}(\beta \cdot m)$

Fully dynamic low-diameter decomposition

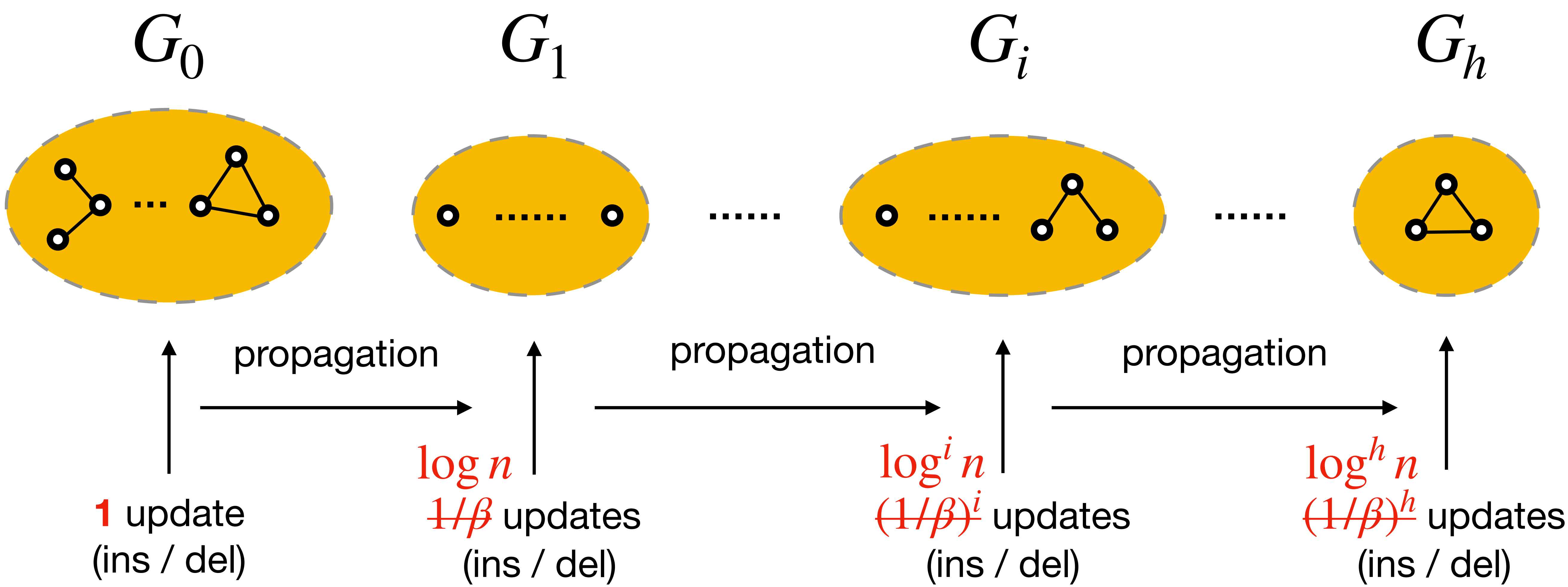
- Lemma:
  A β-decomposition can be maintained under edge updates such that:
  (1) amortized update time is $\tilde{O}(1/\beta^2)$
  (2) amortized **changes to inter-cluster** edges is $\tilde{O}(1)$

- Handle edge insertions **lazily**

  Apply decremental β-decomposition, and ignore all edge insertions, and **rebuild after every βm** edge updates

- **Amortized changes** $\approx \dfrac{\text{total changes}}{\text{total updates}} = \dfrac{\beta m}{\beta m} = 1$

Total update time of hierarchical clustering

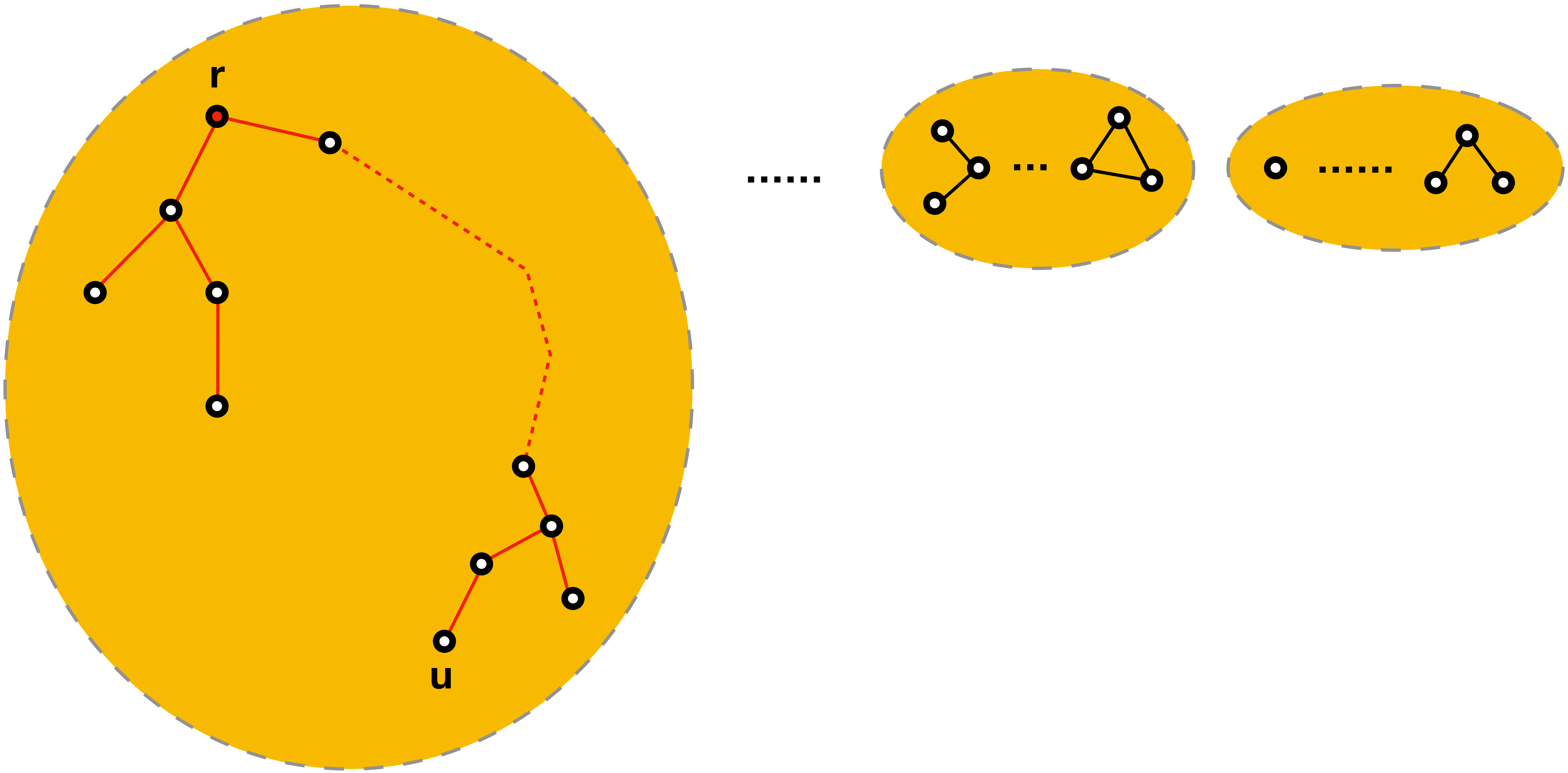


- Overall update time $\approx (1/\beta)^{h+2} + m\beta^h = m^{0.5+o(1)}$

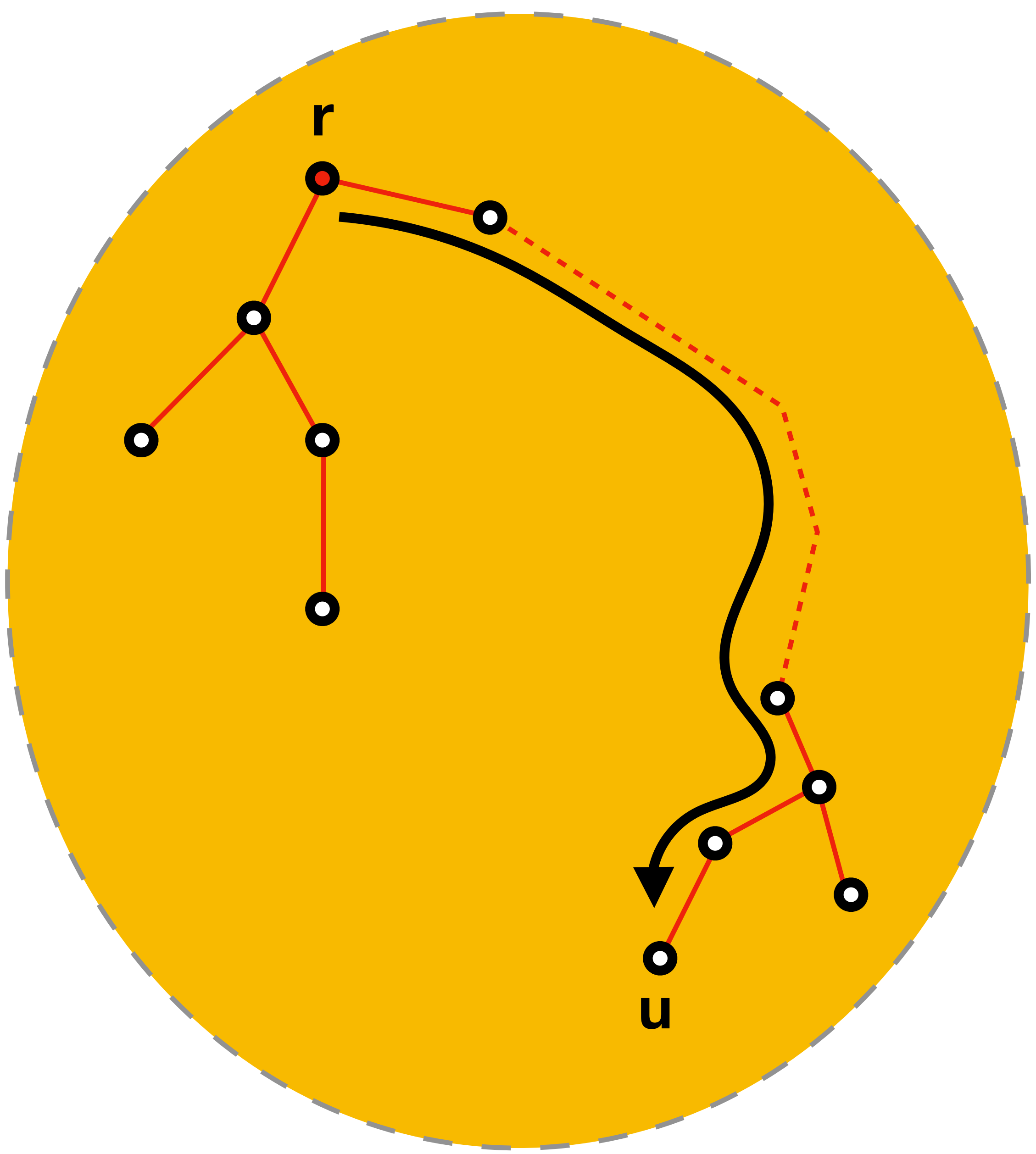
Overall update time $\approx \beta^{-2} \log^h n + m\beta^h = n^{o(1)}$

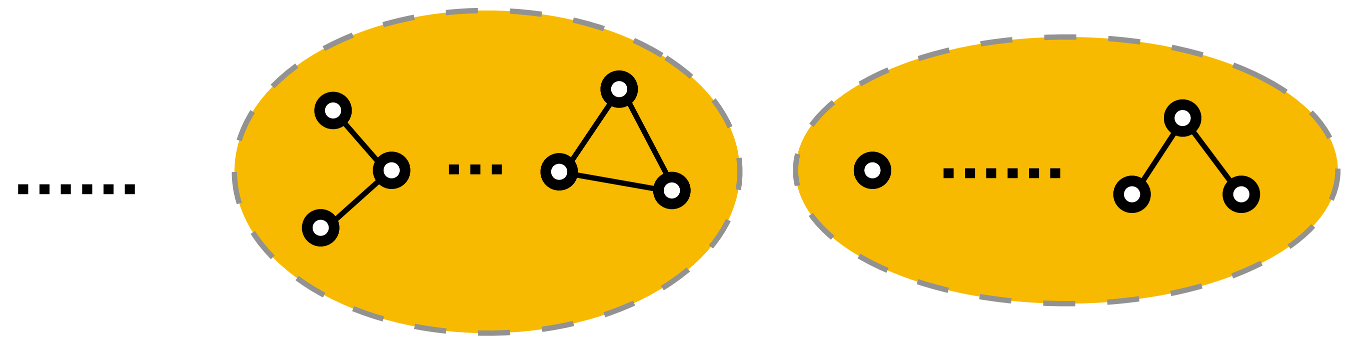
- Graph sparsification can improve $m^{0.5+o(1)}$ to $n^{0.5+o(1)}$

Decremental β-decomposition



Decremental β-decomposition

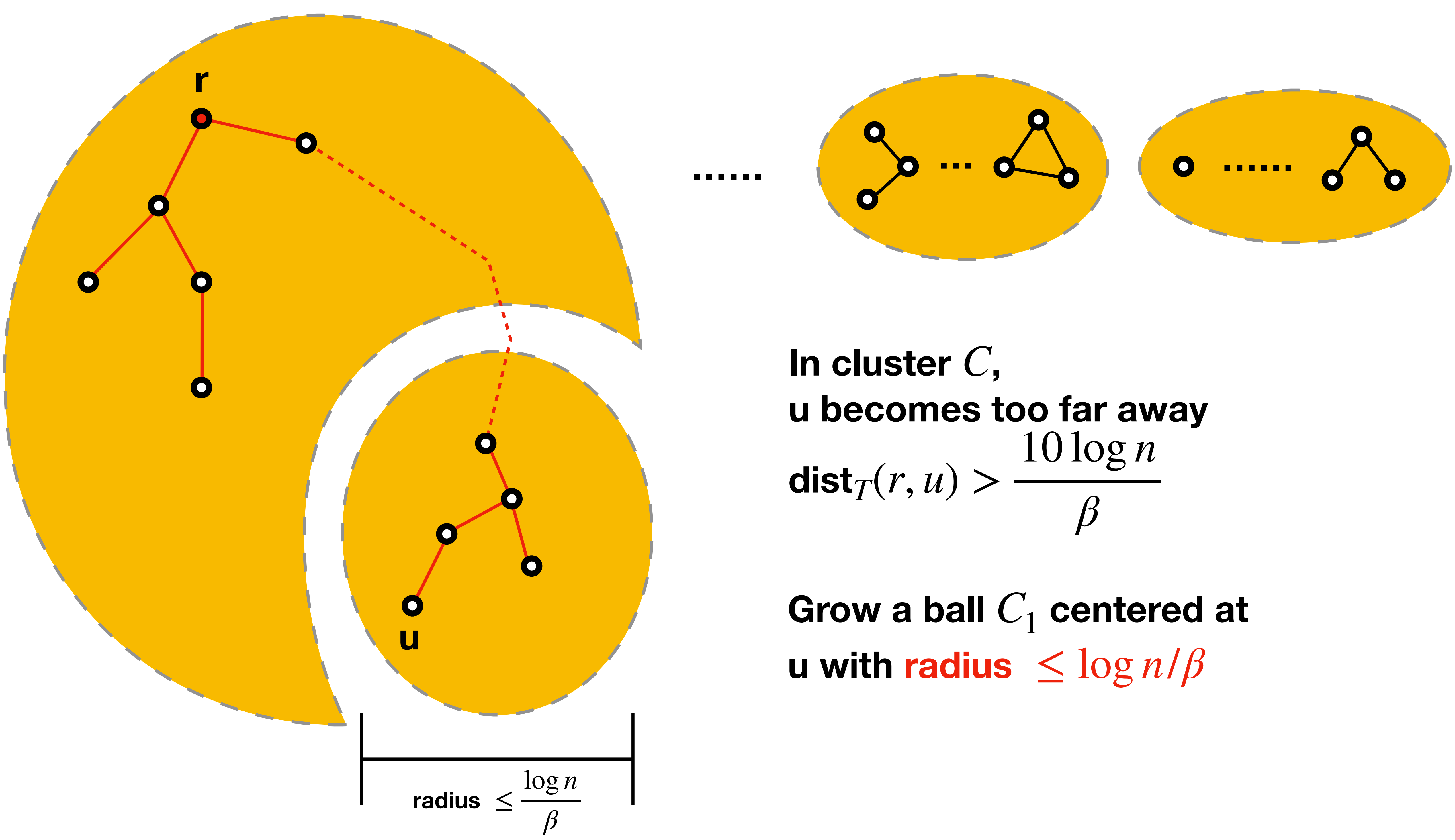




In cluster C,
u becomes too far away

dist$_T(r, u) > \dfrac{10 \log n}{\beta}$

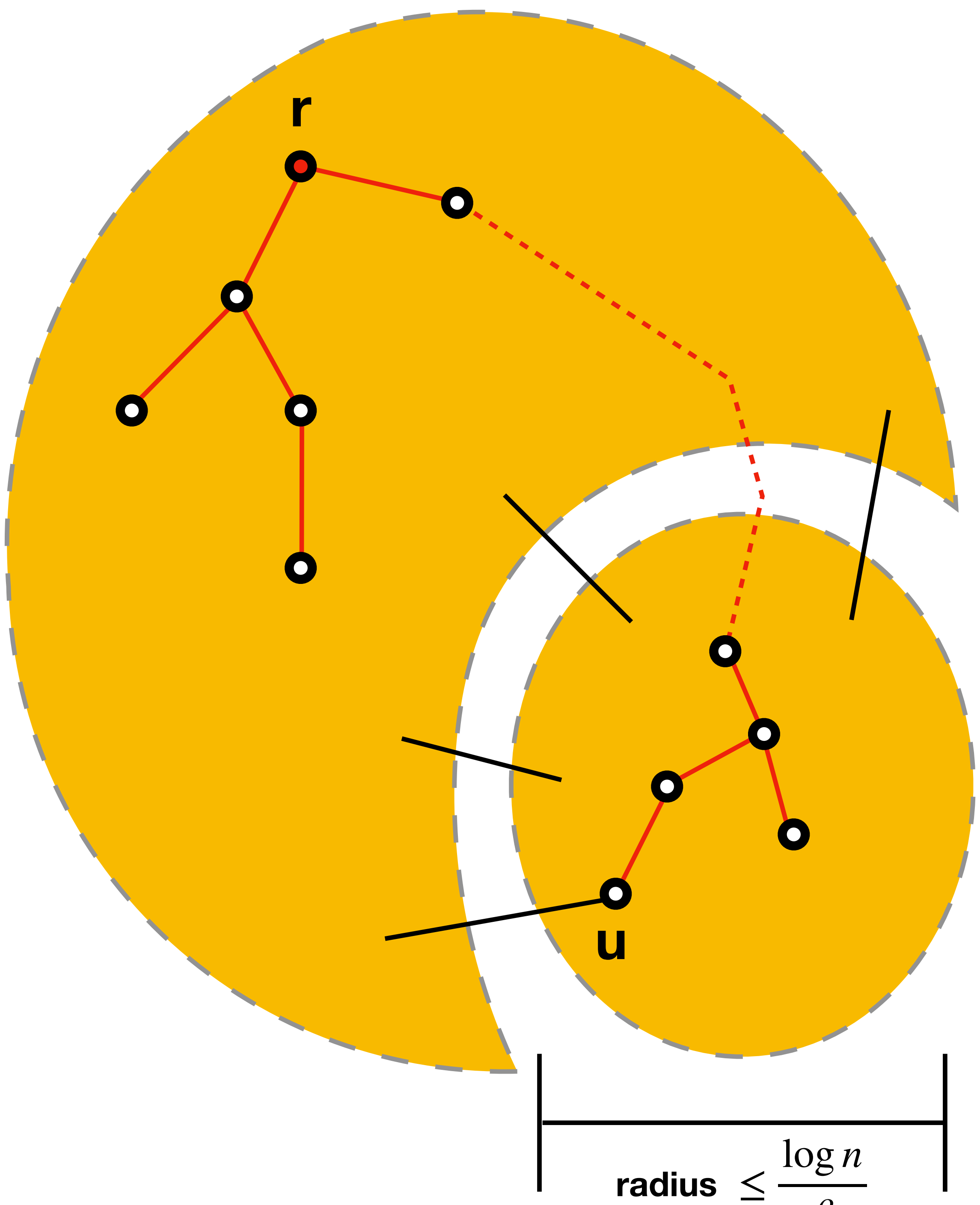
Decremental β-decomposition

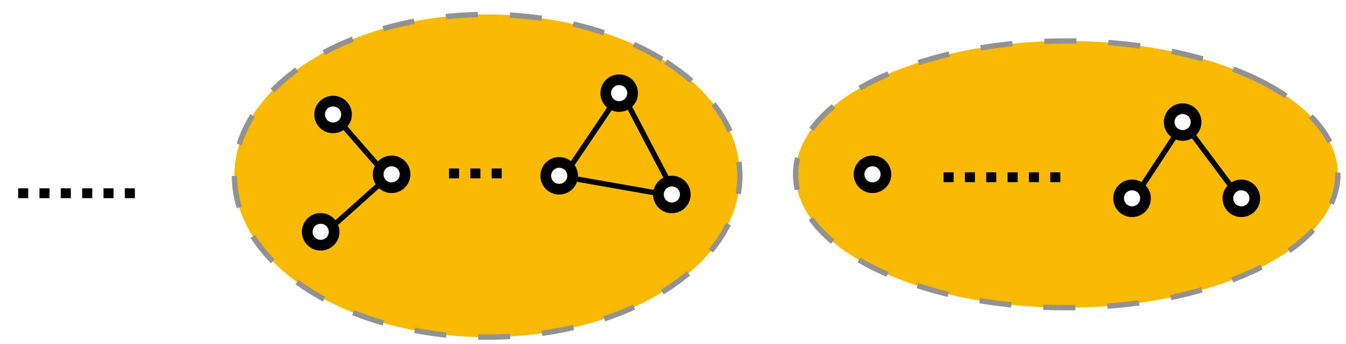


In cluster C,
u becomes too far away
dist$_T(r, u) > \frac{10 \log n}{\beta}$

Grow a ball C_1 centered at u with radius $\leq \log n / \beta$

Decremental β-decomposition



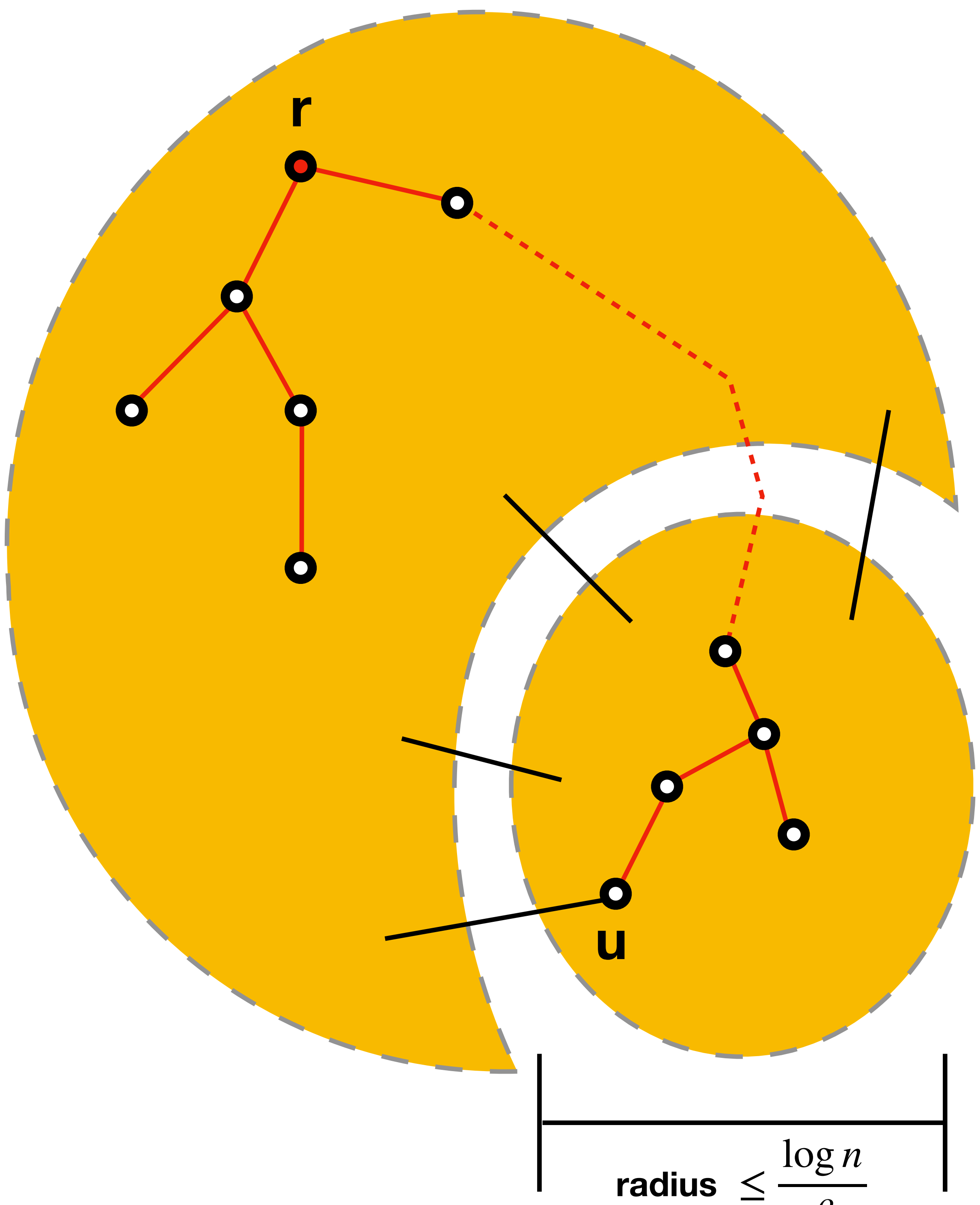


In cluster C,
u becomes too far away
dist$_T(r, u) > \frac{10 \log n}{\beta}$

Grow a ball C_1 centered at u with radius $\leq \log n / \beta$

The cut is sparse
$|E \cap (C \times C_1)| \leq \beta \cdot \mathbf{vol}(C_1)$

Decremental β-decomposition



Let C_1 be a new cluster if it does not contain too many edges, namely:

$$\mathbf{vol}(C_1) \leq \frac{1}{2}\mathbf{vol}(C^{\mathbf{init}})$$

Otherwise ……

In cluster C,
u becomes too far away

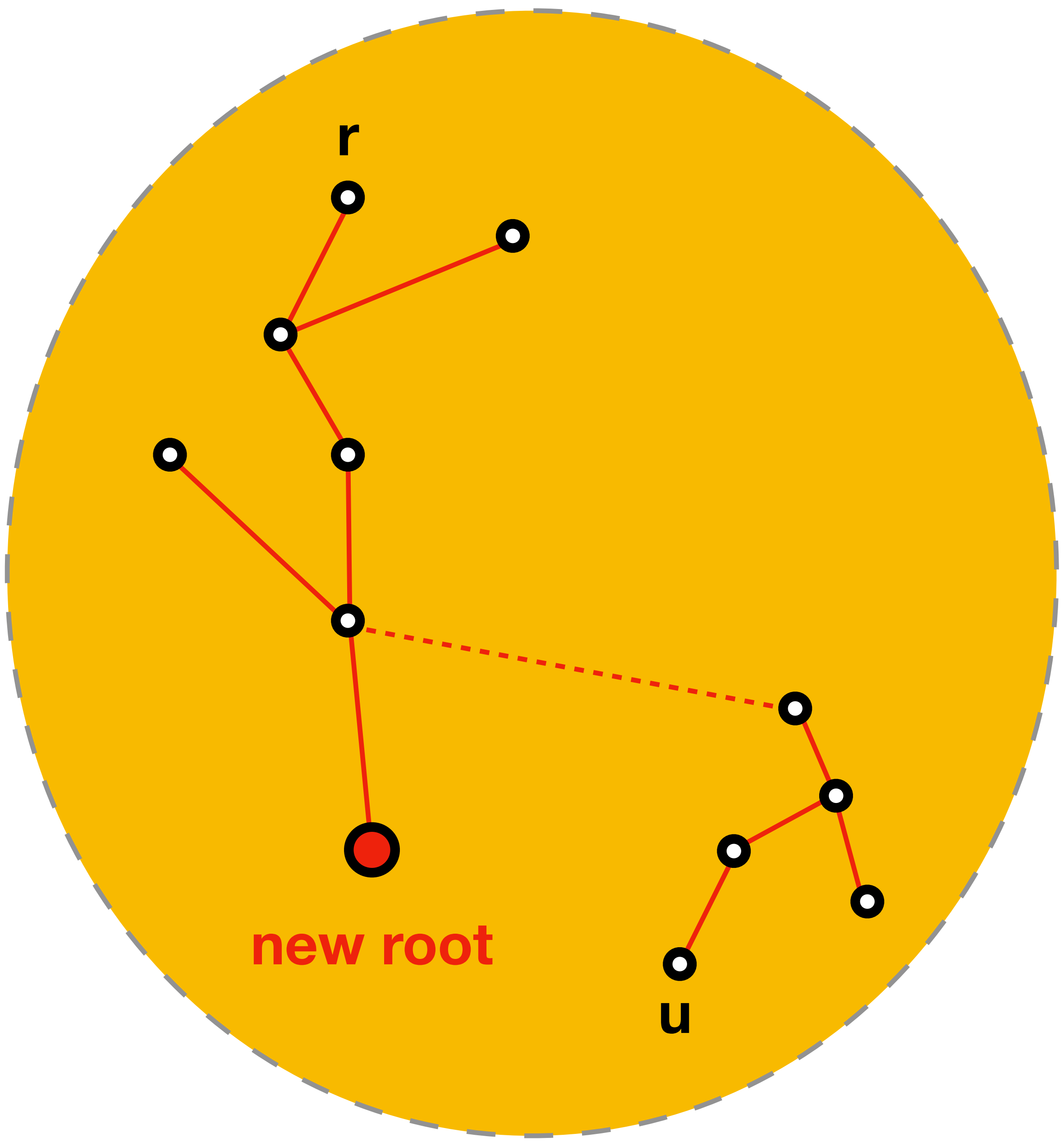
$$\mathbf{dist}_T(r, u) > \frac{10 \log n}{\beta}$$

Grow a ball C_1 centered at u with radius $\leq \log n / \beta$

Cut size is small

$|E \cap (C \times C_1)| \leq \beta \cdot \mathbf{vol}(C_1)$

Decremental β-decomposition



Let C_1 be a new cluster if it does not contain too many edges, namely:

$$\text{vol}(C_1) \leq \frac{1}{2}\text{vol}(C^{\text{init}})$$

Otherwise, randomly reassign the root and rebuild a new BFS tree

In cluster C,
u becomes too far away

$$\text{dist}_T(r, u) > \frac{10 \log n}{\beta}$$

Grow a ball C_1 centered at u with radius $\leq \log n / \beta$

Cut size is small

$$|E \cap (C \times C_1)| \leq \beta \cdot \text{vol}(C_1)$$

Correctness & running time

- Lemma: (total # inter-cluster edges)
  Each time a new cluster C_1 is created,
  # new inter-cluster edges $\leq \beta \cdot \text{vol}(C_1) \leq \beta \cdot \frac{1}{2}\text{vol}(C^{\text{init}})$
  Therefore, eventually #inter-cluster edges $\leq m\beta \log n$

- Lemma: (total running time)
  Each cluster reassigns its BFS tree root for $O(\log n)$ times, with high probability. Hence total time is $\tilde{O}(m/\beta)$

Thanks