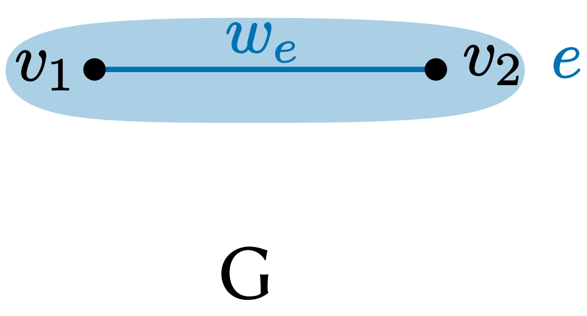
Fully Dynamic Spectral Sparsification for Directed Hypergraphs

Sebastian Forster [1] Gramoz Goranci [2] Ali Momeni [2]

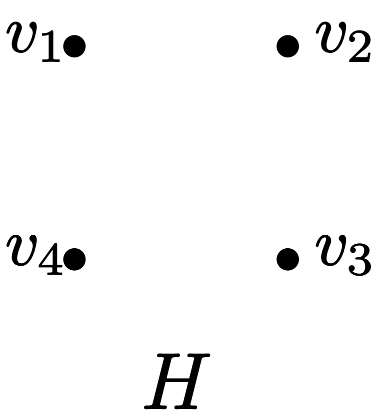
[1] University of Salzburg, Austria
[2] University of Vienna, Austria

March 2026, Grenoble

Hypergraphs



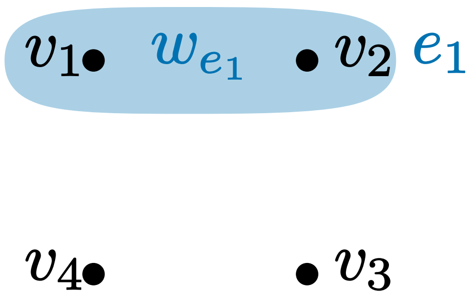
$e = \{v_1, v_2\}$



Hypergraphs

G

$e = \{v_1, v_2\}$



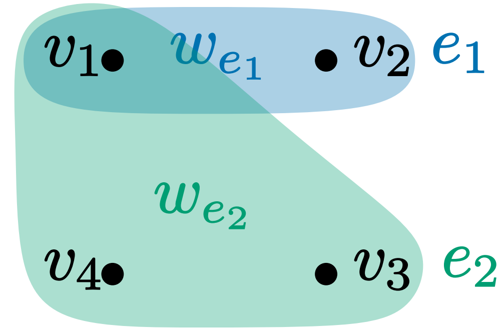
H

$e_1 = \{v_1, v_2\}$

Hypergraphs

G

$e = \{v_1, v_2\}$



H

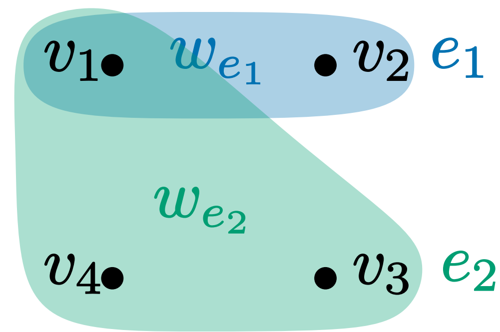
$e_1 = \{v_1, v_2\}$

$e_2 = \{v_1, v_3, v_4\}$

Hypergraphs

G

$$e = \{v_1, v_2\}$$



H

$$e_1 = \{v_1, v_2\}$$

$$e_2 = \{v_1, v_3, v_4\}$$

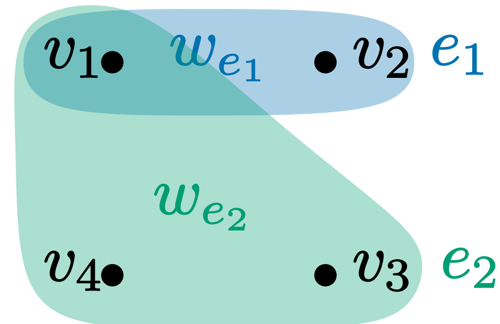
Interactions between the entities:

- Graphs: model *pairwise* interactions
- Hypergraphs: model *arbitrary* interactions

Hypergraphs

G

$e = \{v_1, v_2\}$



H

$e_1 = \{v_1, v_2\}$

$e_2 = \{v_1, v_3, v_4\}$

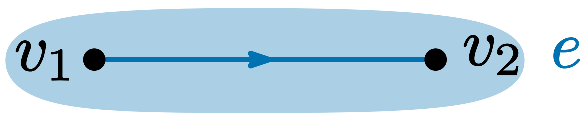
Interactions between the entities:

- Graphs: model *pairwise* interactions
- Hypergraphs: model *arbitrary* interactions

Applications:

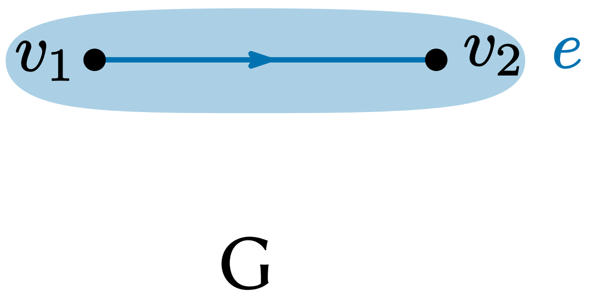
Computational biology, circuit design, database systems, ...

Directed Hypergraphs



G

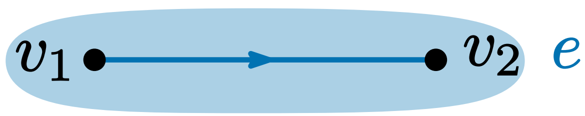
Directed Hypergraphs



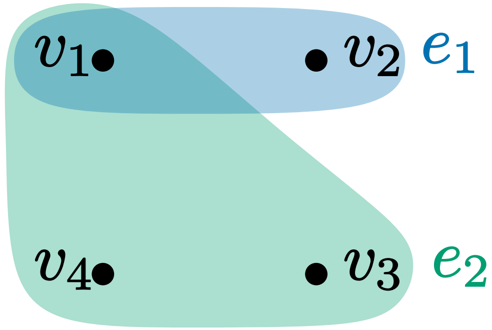
$\mathrm{t}(e) = \{v_1\}$

$\mathrm{h}(e) = \{v_2\}$

Directed Hypergraphs



G

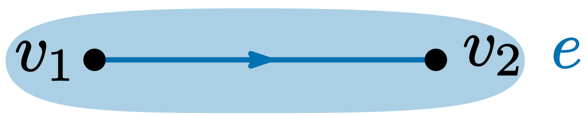


H

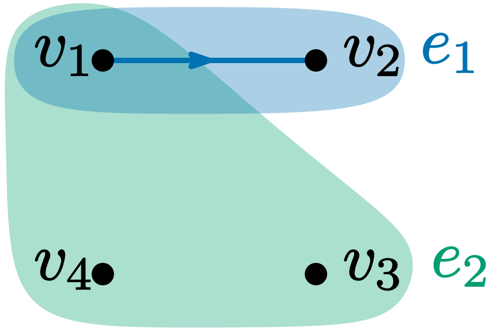
$t(e) = \{v_1\}$

$h(e) = \{v_2\}$

Directed Hypergraphs



G



H

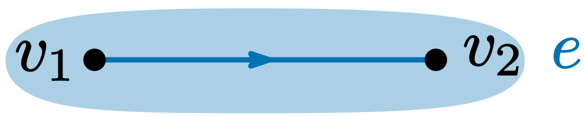
$\mathrm{t}(e) = \{v_1\}$

$\mathrm{h}(e) = \{v_2\}$

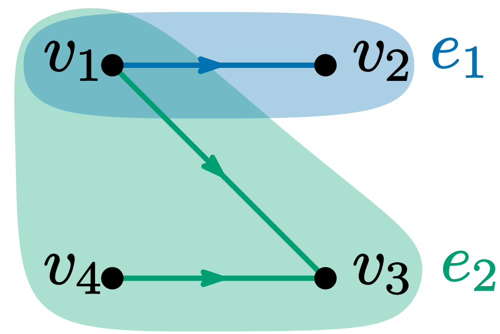
$\mathrm{t}(e_1) = \{v_1\}$

$\mathrm{h}(e_1) = \{v_2\}$

Directed Hypergraphs



G



H

$\mathrm{t}(e) = \{v_1\}$

$\mathrm{h}(e) = \{v_2\}$

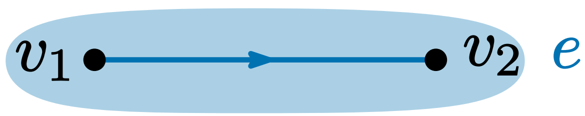
$\mathrm{t}(e_1) = \{v_1\}$

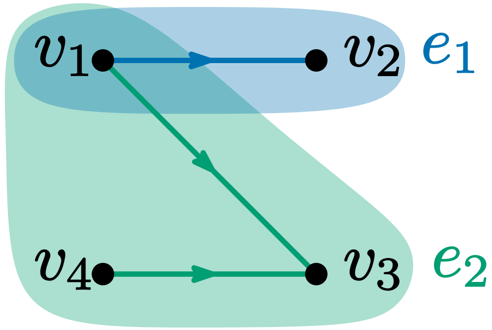
$\mathrm{h}(e_1) = \{v_2\}$

$\mathrm{t}(e_2) = \{v_1, v_4\}$

$\mathrm{h}(e_2) = \{v_3\}$

Directed Hypergraphs





$\mathrm{t}(e) = \{v_1\}$

$\mathrm{h}(e) = \{v_2\}$

$\mathrm{t}(e_1) = \{v_1\}$

$\mathrm{h}(e_1) = \{v_2\}$

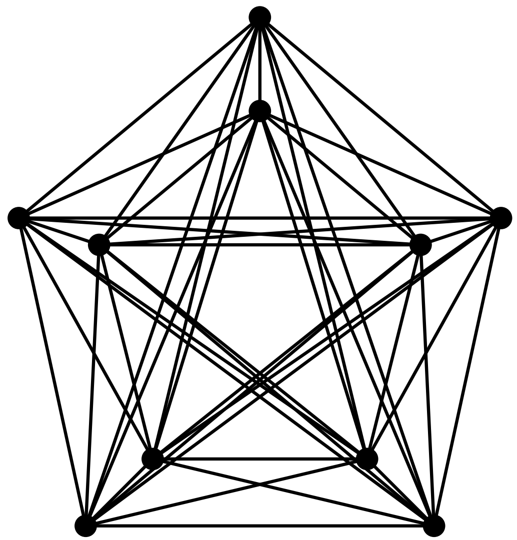
$\mathrm{t}(e_2) = \{v_1, v_4\}$

$\mathrm{h}(e_2) = \{v_3\}$

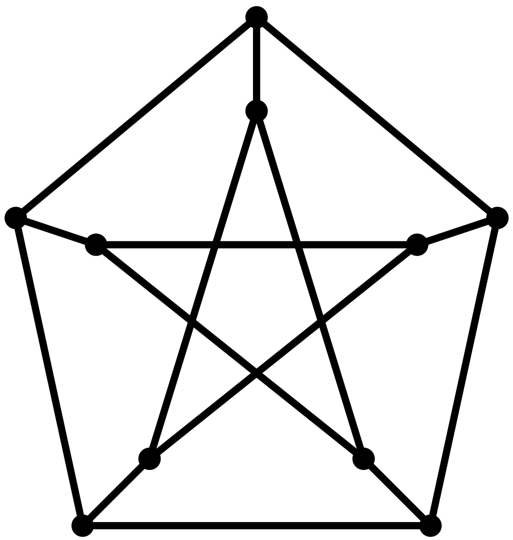
Notation: $m = |E|$, $n = |V|$, and

$$\forall e \in E \quad |e| \leq r$$

Spectral Sparsification I



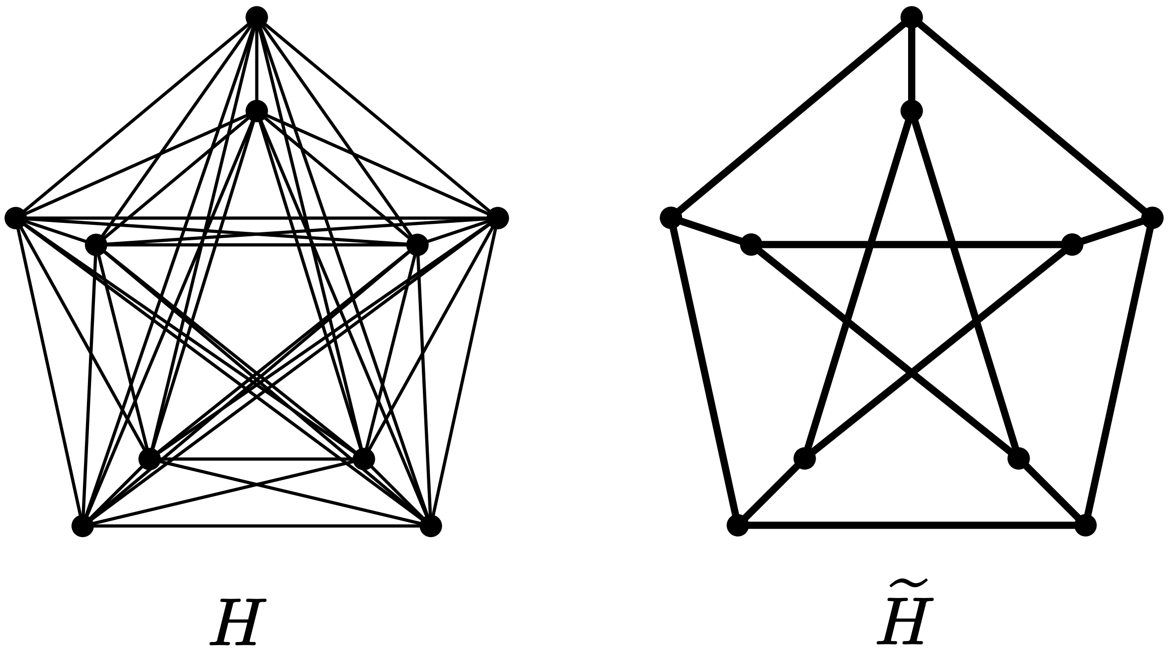
H



$\widetilde{H}$

Example from [ST11]

Spectral Sparsification I



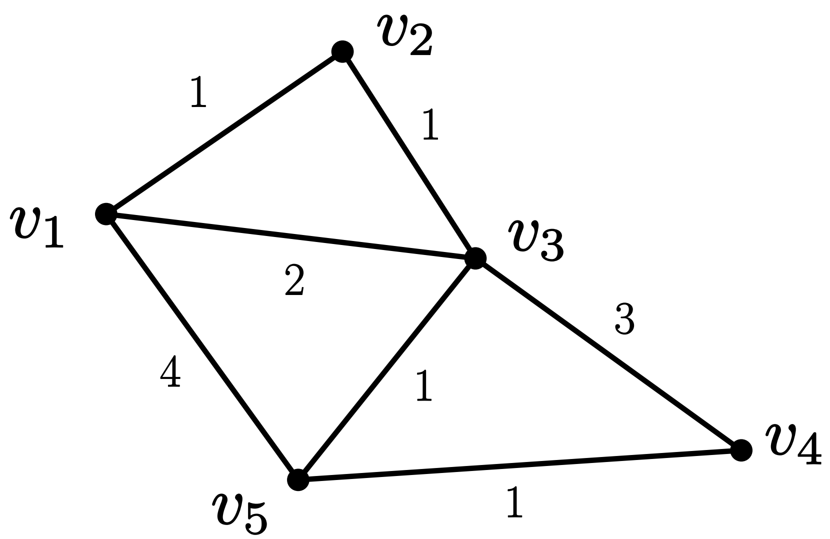
Example from [ST11]

Spectral Sparsifier: hypergraph $\widetilde{H}$ s.t. for every $\mathbf{x} \in \mathbb{R}^n$,

$$(1-\varepsilon)Q_{\widetilde{H}}(\mathbf{x}) \le Q_H(\mathbf{x}) \le (1+\varepsilon)Q_{\widetilde{H}}(\mathbf{x}).$$

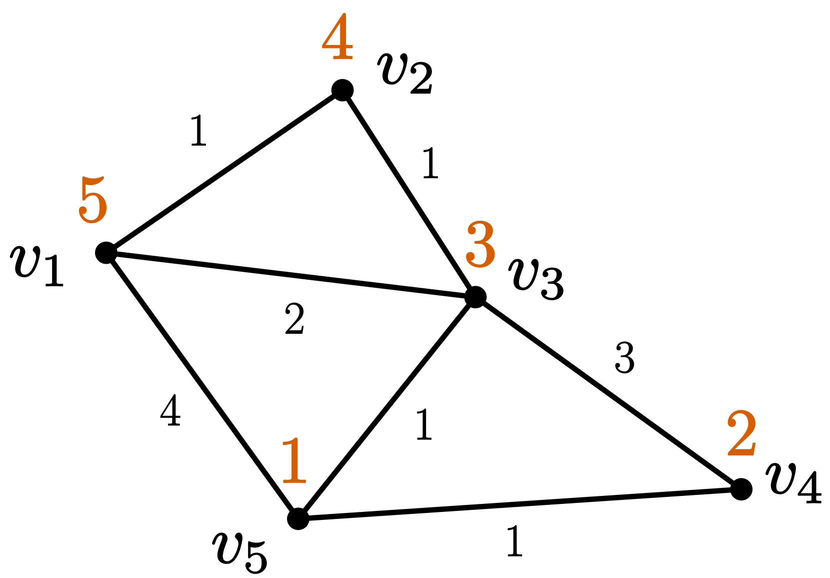
Spectral Sparsification II

Undirected Graphs: For vector $\mathbf{x} \in \mathbb{R}^n$,



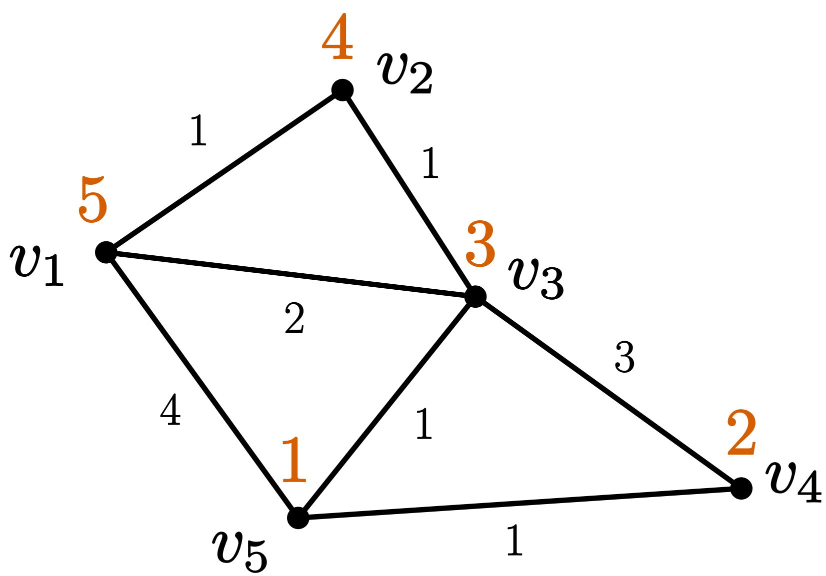
Spectral Sparsification II

Undirected Graphs: For vector $\mathbf{x} \in \mathbb{R}^n$,



Spectral Sparsification II

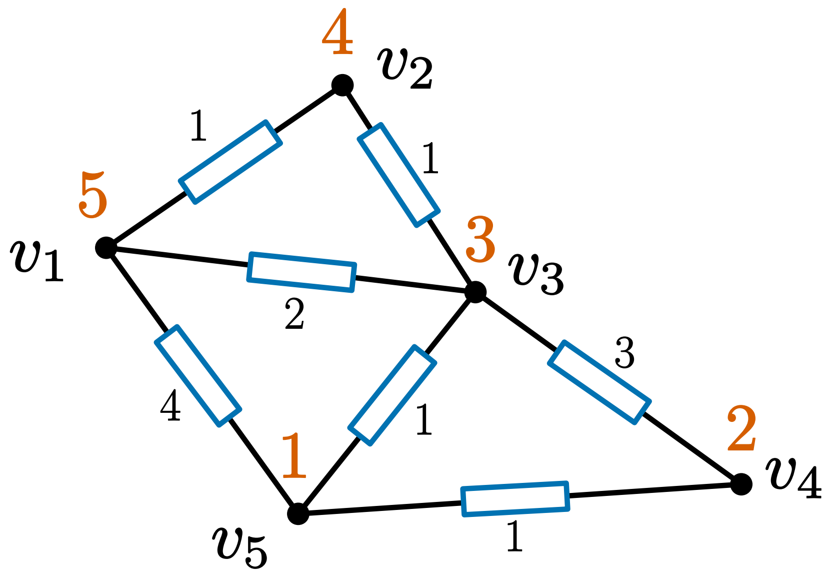
Undirected Graphs: For vector $\mathbf{x} \in \mathbb{R}^n$,



$$Q_G(\mathbf{x}) = \sum_{e \in E} w_e \left(x_u - x_v\right)^2$$

Spectral Sparsification II

Undirected Graphs: For vector $\mathbf{x} \in \mathbb{R}^n$,

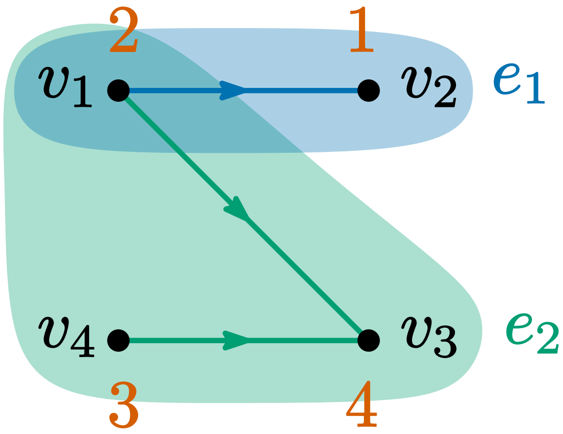


$$Q_G(\mathbf{x}) = \sum_{e \in E} w_e \left(x_u - x_v\right)^2$$

Electrical Network: electric power!

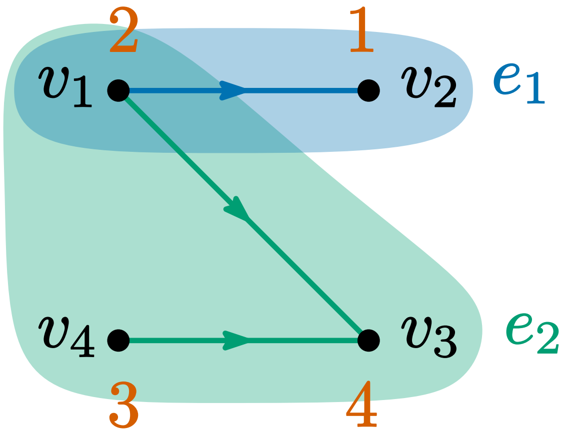
Spectral Sparsification III

Directed Hypergraphs: For vector $\mathbf{x} \in \mathbb{R}^n$,



Spectral Sparsification III

Directed Hypergraphs: For vector $\mathbf{x} \in \mathbb{R}^n$,



$$Q_H(\mathbf{x}) = \sum_{e \in E} w_e \max_{u \in t(e), v \in h(e)} (x_u - x_v)_+^2 ,$$

where

$$(x_u - x_v)_+ = \max\{x_u - x_v, 0\}.$$

Spectral Sparsification IV

	Undirected	Directed
Graphs:	[ST11,SS08] $\cdots$	
Hypergraphs:		

Spectral Sparsification IV

	Undirected	Directed
Graphs:	[ST11,SS08] $\cdots$	
Hypergraphs:	[NR13] $\cdots$ [JLS23, Lee23]	

Spectral Sparsification IV

	Undirected	Directed
Graphs:	[ST11,SS08] $\cdots$	$\cdots$ [JSSTZ25]
Hypergraphs:	[NR13] $\cdots$ [JLS23, Lee23]	[SY19] $\cdots$ [OST23]

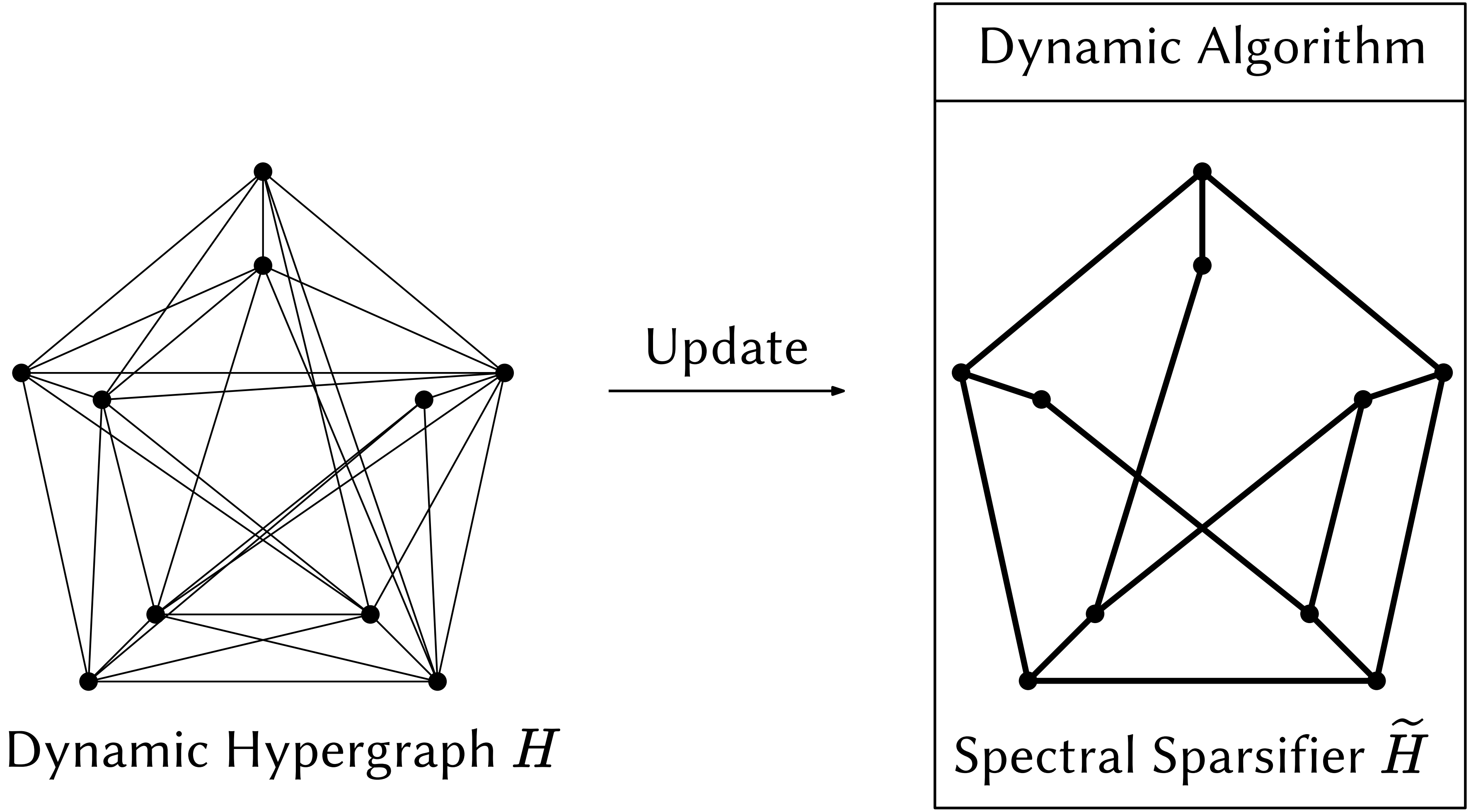
Spectral Sparsification IV

	Undirected	Directed
Graphs:	[ST11,SS08] $\cdots$	$\cdots$ [JSSTZ25]
Hypergraphs:	[NR13] $\cdots$ [JLS23, Lee23]	[SY19] $\cdots$ [OST23]

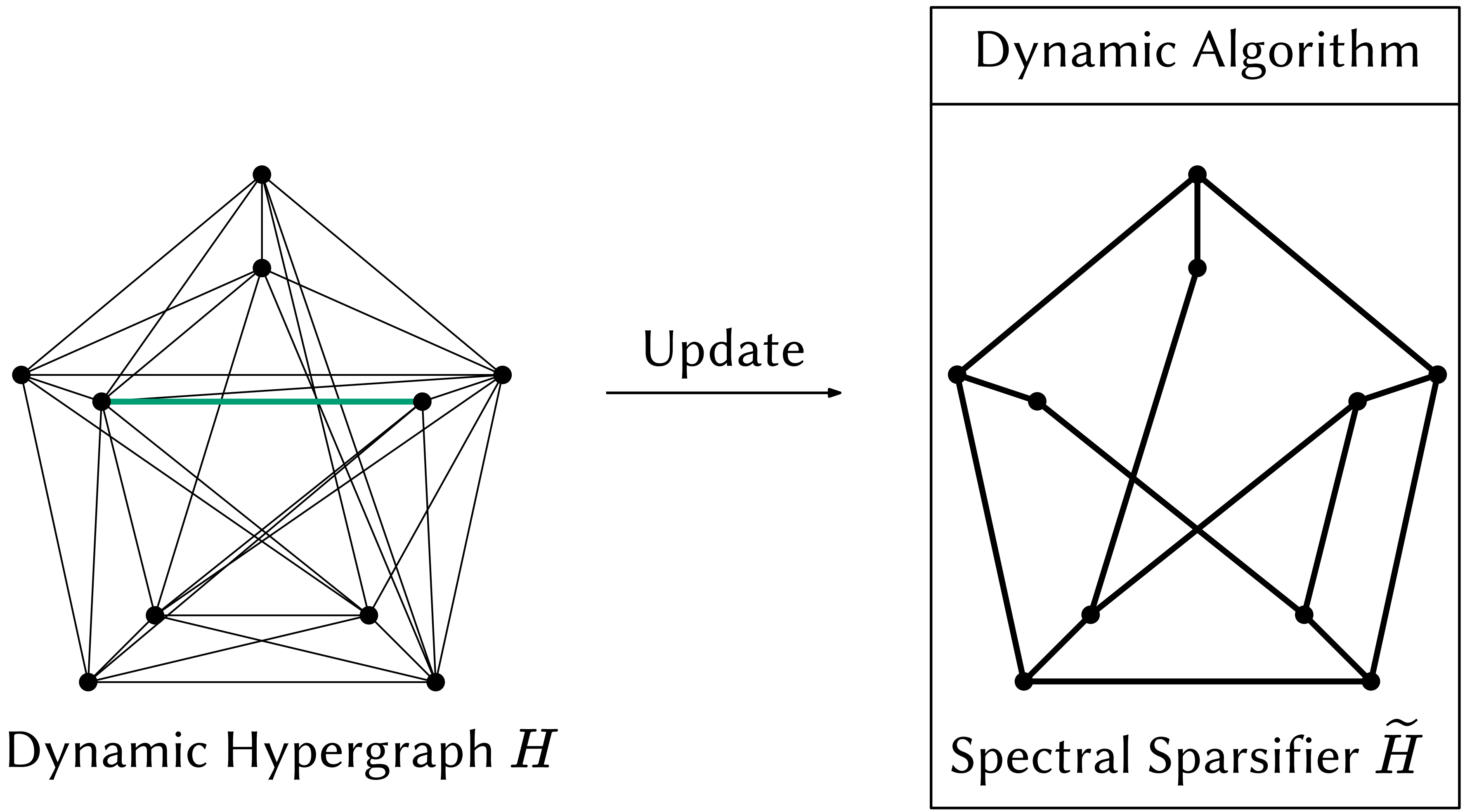
There is a lot more!

- Different settings: parallel, distributed, dynamic, ...
- Approximation: additive or multiplicative
- Different graph classes.

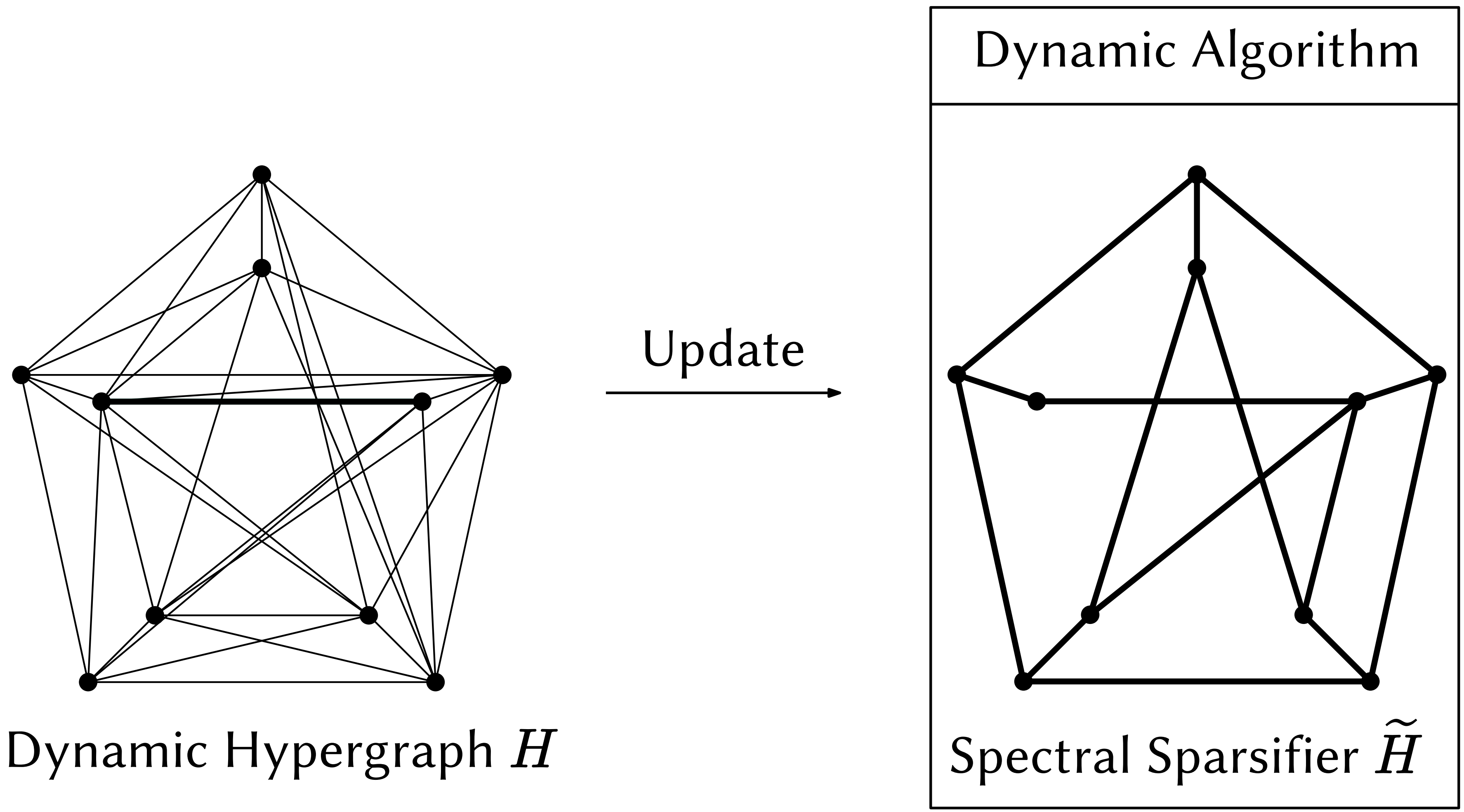
The Dynamic Setting



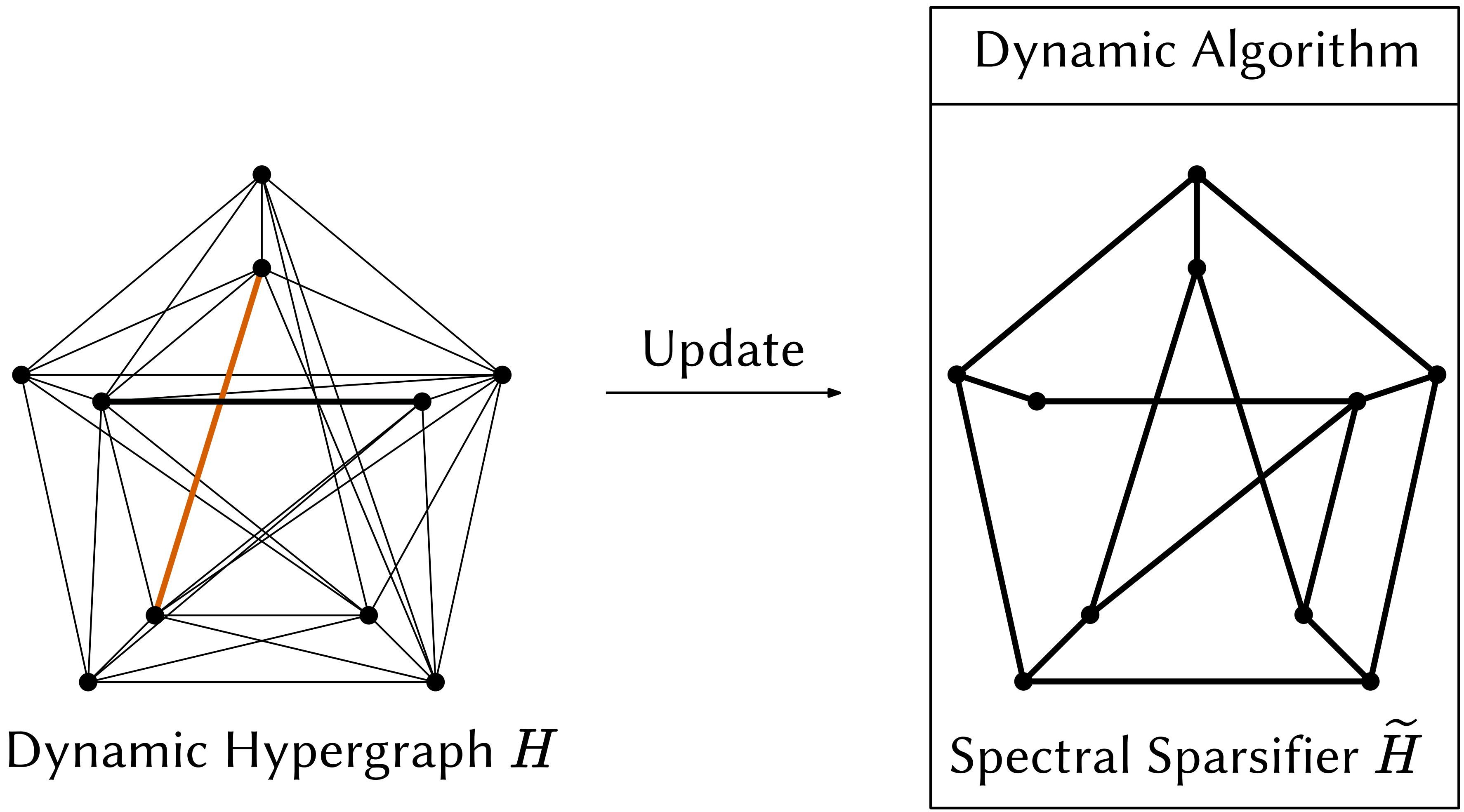
The Dynamic Setting



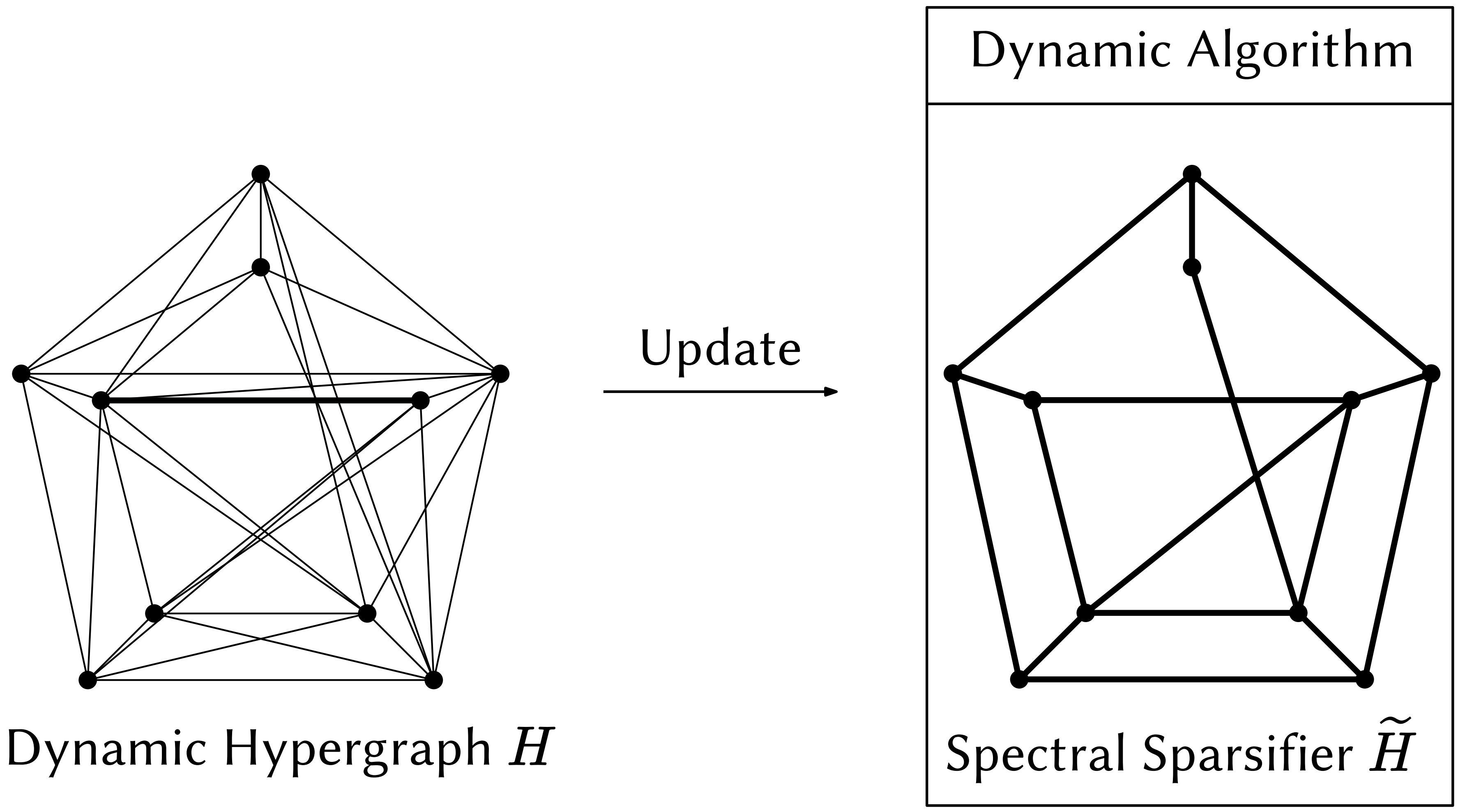
The Dynamic Setting



The Dynamic Setting



The Dynamic Setting



Our Results

Our Main Result: For $0 < \varepsilon < 1$, with high probability, we maintain a $(1 \pm \varepsilon)$-spectral sparsifier $\widetilde{H}$ of H with:

- Amortized update time $O(r^2 \log^3 m)$, and
- Size $O(n^2/\varepsilon^2 \log^7 m)$, and
- Initialization time $O(mr^2 \log^2 m)$.

Our Results

Our Main Result: For $0 < \varepsilon < 1$, with high probability, we maintain a $(1 \pm \varepsilon)$-spectral sparsifier $\widetilde{H}$ of H with:

- Amortized update time $O(r^2 \log^3 m)$, and
- Size $O(n^2/\varepsilon^2 \log^7 m)$, and
- Initialization time $O(mr^2 \log^2 m)$.

Nearly tight guarantees:

- Update time is $\Omega(r)$, and
- Size is $\Omega(n^2/\varepsilon)$ [OST23].

Our Results

Our Main Result: For $0 < \varepsilon < 1$, with high probability, we maintain a $(1 \pm \varepsilon)$-spectral sparsifier $\widetilde{H}$ of H with:

- Amortized update time $O(r^2 \log^3 m)$, and
- Size $O(n^2/\varepsilon^2 \log^7 m)$, and
- Initialization time $O(mr^2 \log^2 m)$.

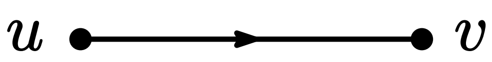
Nearly tight guarantees:

- Update time is $\Omega(r)$, and
- Size is $\Omega(n^2/\varepsilon)$ [OST23].

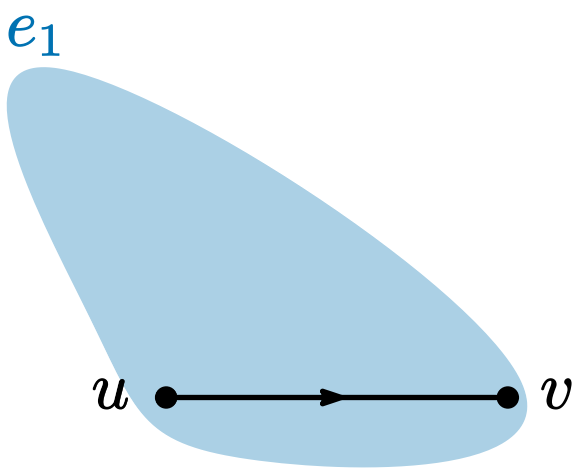
Parallel Batch-Dynamic Result: Batches of k insertions and deletions:

- Amortized work $O(kr^2 \log^3 m)$, and
- Depth $O(\log^2 m)$, and
- Same size and Initialization time.

Sparsifier Construction I

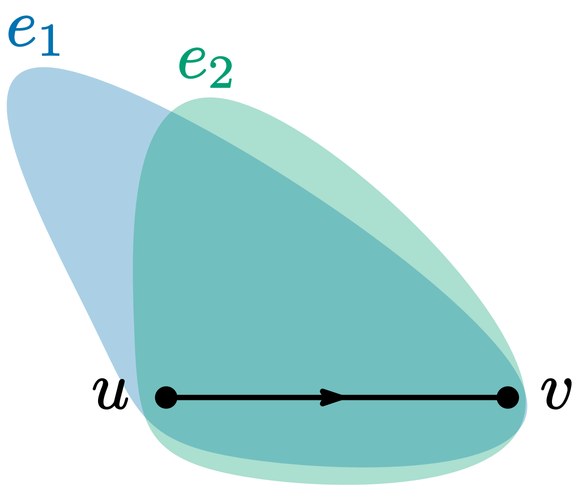


Sparsifier Construction I



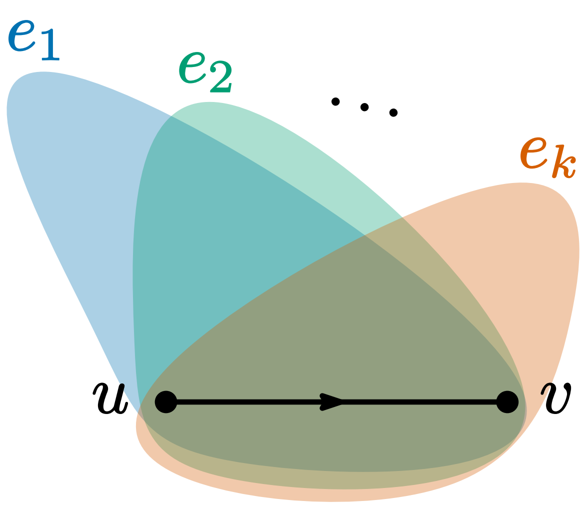
w_{e_1}

Sparsifier Construction I



$w_{e_1} \geq w_{e_2}$

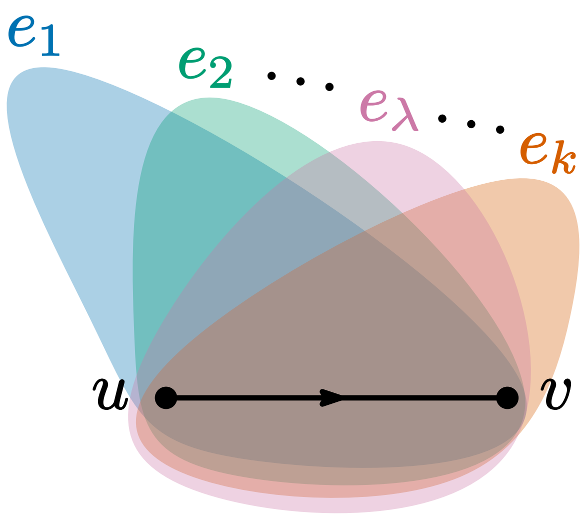
Sparsifier Construction I



$$w_{e_1} \geq w_{e_2} \geq \quad \cdots \quad \geq w_{e_k}$$

$$k \leq m$$

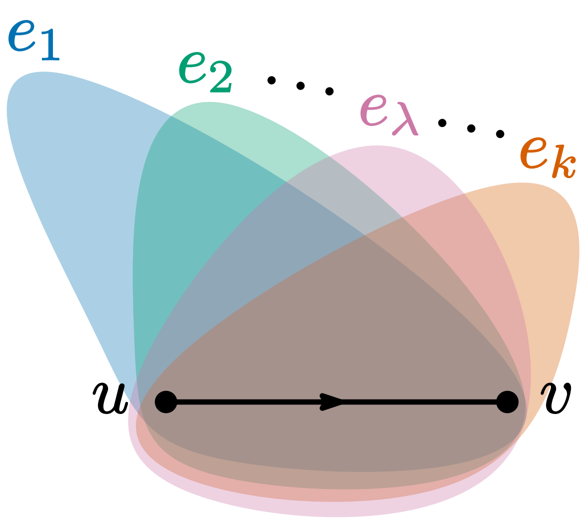
Sparsifier Construction I



$$w_{e_1} \geq w_{e_2} \geq \cdots \geq w_{e_\lambda} \geq \cdots \geq w_{e_k}$$

$$k \leq m$$

Sparsifier Construction I

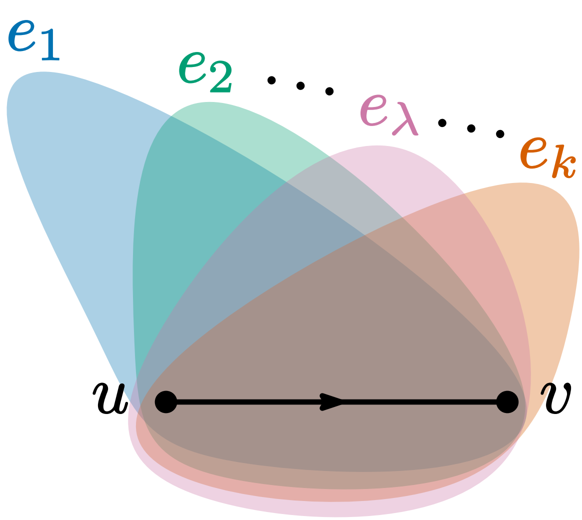


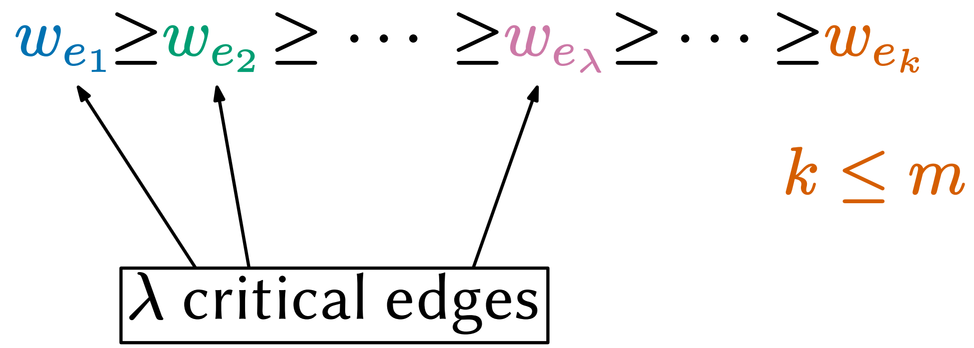
$$w_{e_1} \geq w_{e_2} \geq \cdots \geq w_{e_\lambda} \geq \cdots \geq w_{e_k}$$

λ critical edges

$k \leq m$

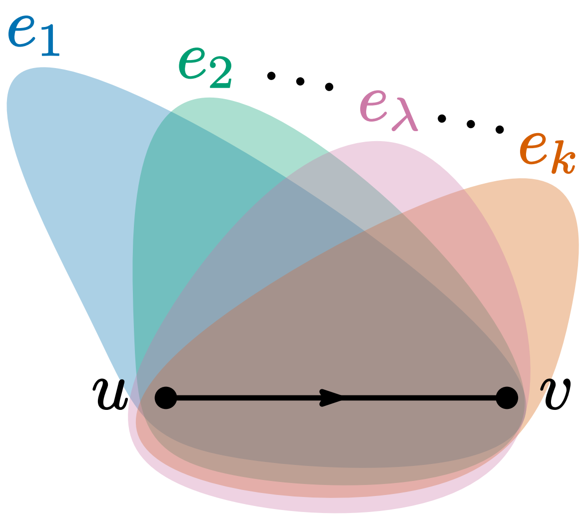
Sparsifier Construction I



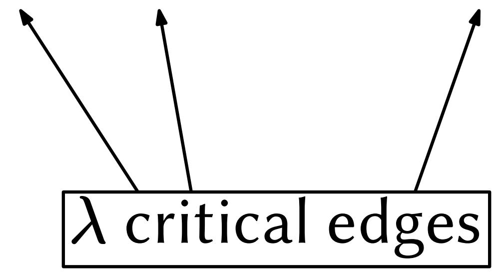


λ-coreset $C = \{\lambda$ critical edges of $uv \mid u, v \in V\}$

Sparsifier Construction I



$$w_{e_1} \geq w_{e_2} \geq \cdots \geq w_{e_\lambda} \geq \cdots \geq w_{e_k}$$



$$k \leq m$$

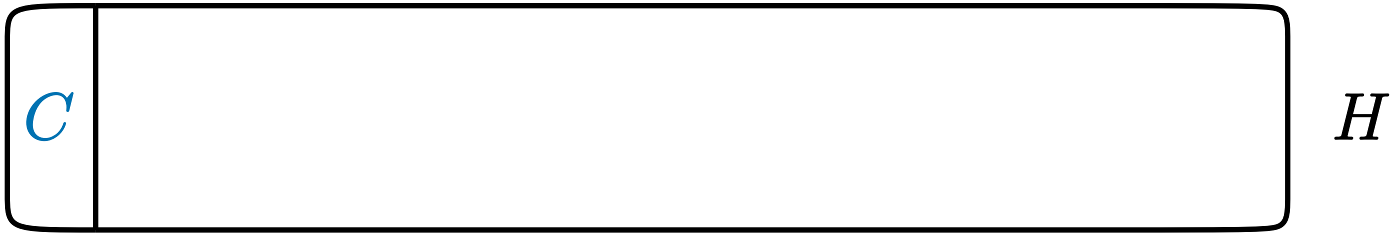
λ-coreset $C = \{\lambda \text{ critical edges of } uv \mid u, v \in V\}$

$$|C| = O(\lambda n^2)$$

$$\lambda = O(\log^3 m / \varepsilon^2)$$

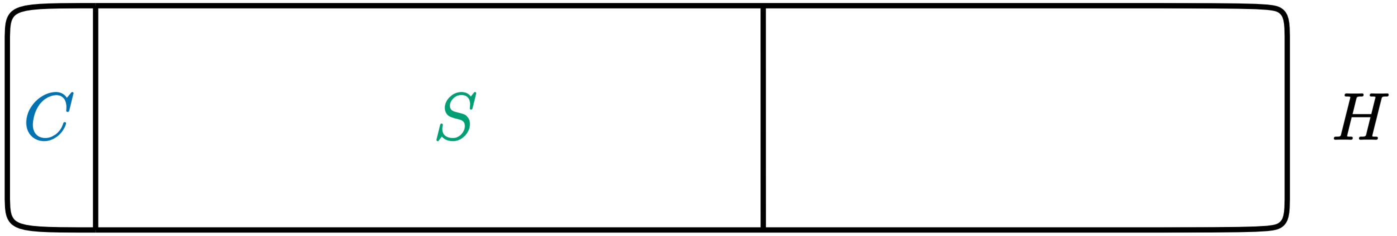
Sparsifier Construction II

The algorithm of [OST23]:



Sparsifier Construction II

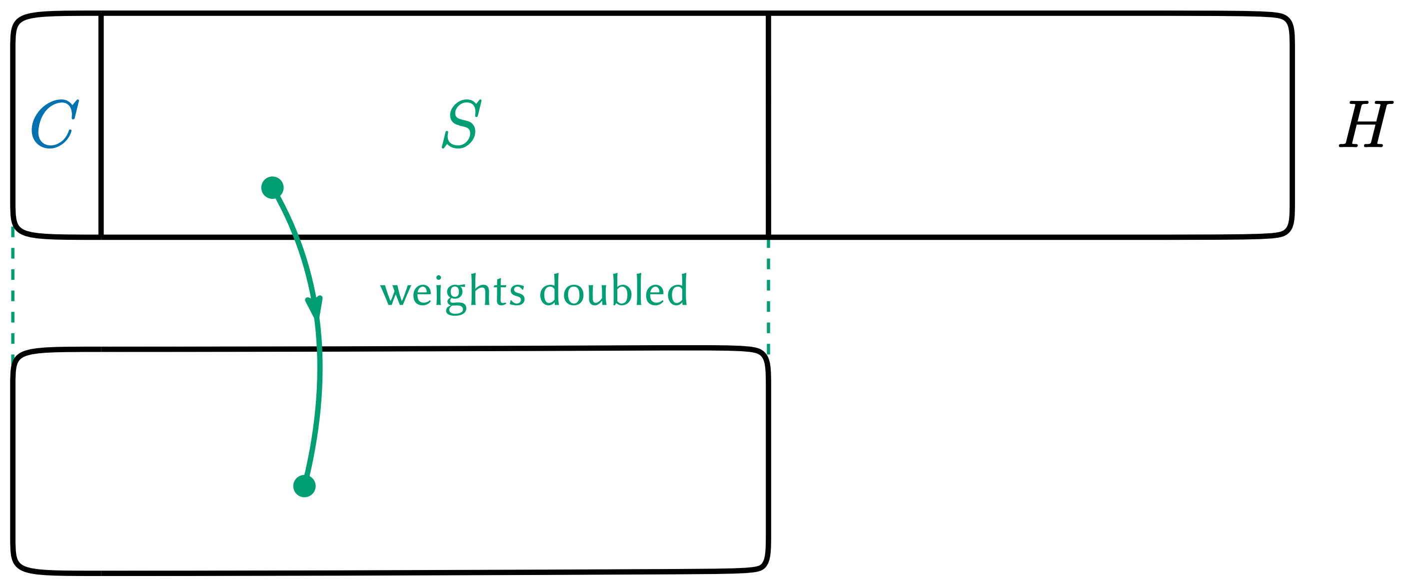
The algorithm of [OST23]:



Sparsifier Construction II

The algorithm of [OST23]:

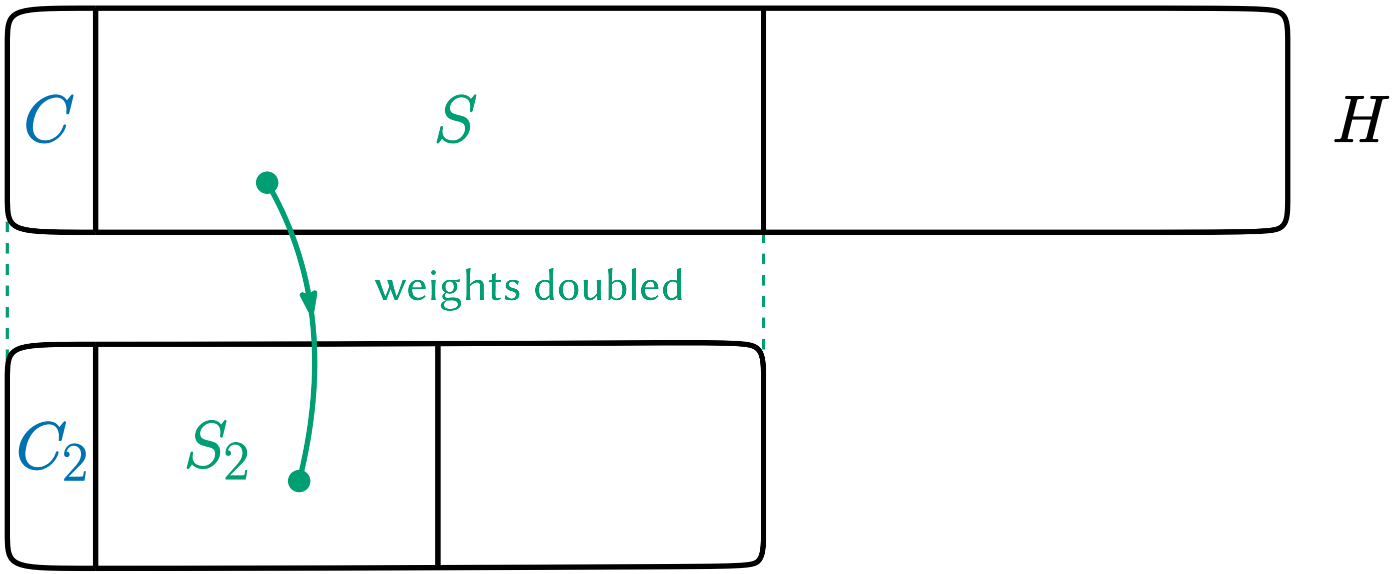
$|C \cup S| \simeq \frac{m}{2}$



Sparsifier Construction II

The algorithm of [OST23]:

$|C \cup S| \simeq \frac{m}{2}$

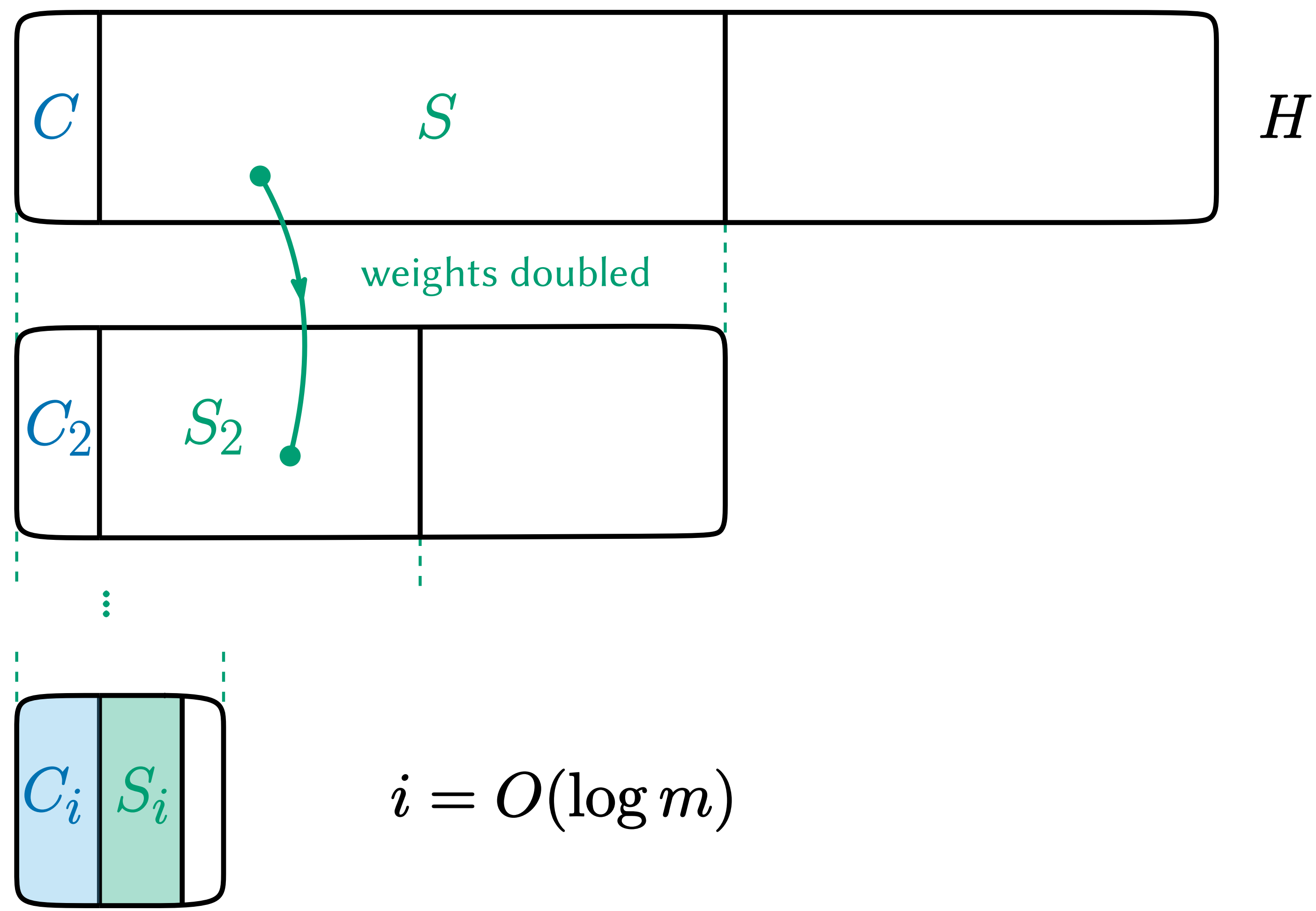


Sparsifier Construction II

The algorithm of [OST23]:

$|C \cup S| \simeq \frac{m}{2}$

$|C_i \cup S_i| \simeq \frac{m}{2^i}$



Dynamic Algorithm

Main Idea:

1. Design a decremental algorithm
2. Generalize to the fully dynamic setting

Dynamic Algorithm

Main Idea:

1. Design a decremental algorithm ⟶ adapt the static algorithm
2. Generalize to the fully dynamic setting

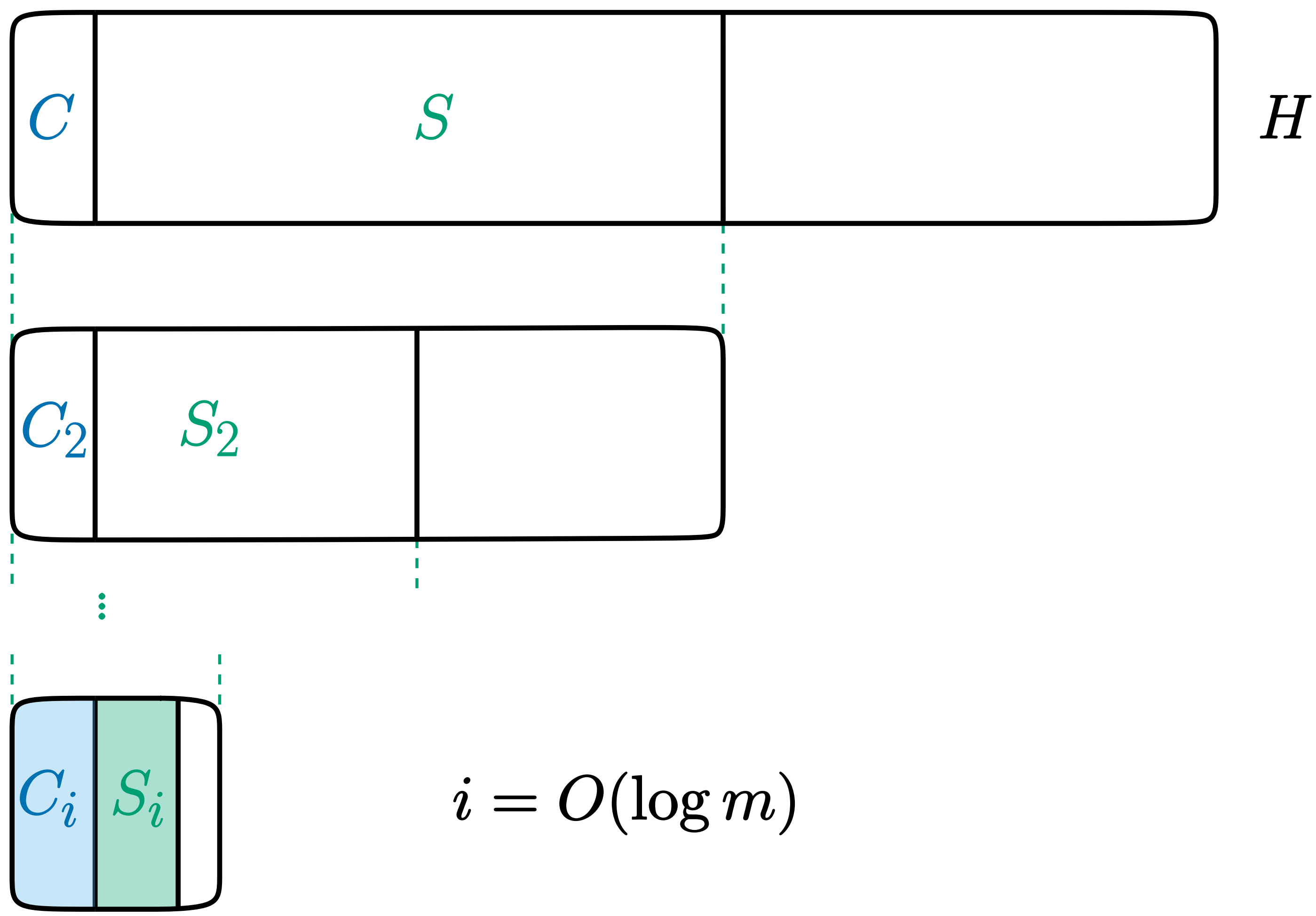
Dynamic Algorithm

Main Idea:

1. Design a decremental algorithm ⟶ adapt the static algorithm
2. Generalize to the fully dynamic setting ⟶ batching!

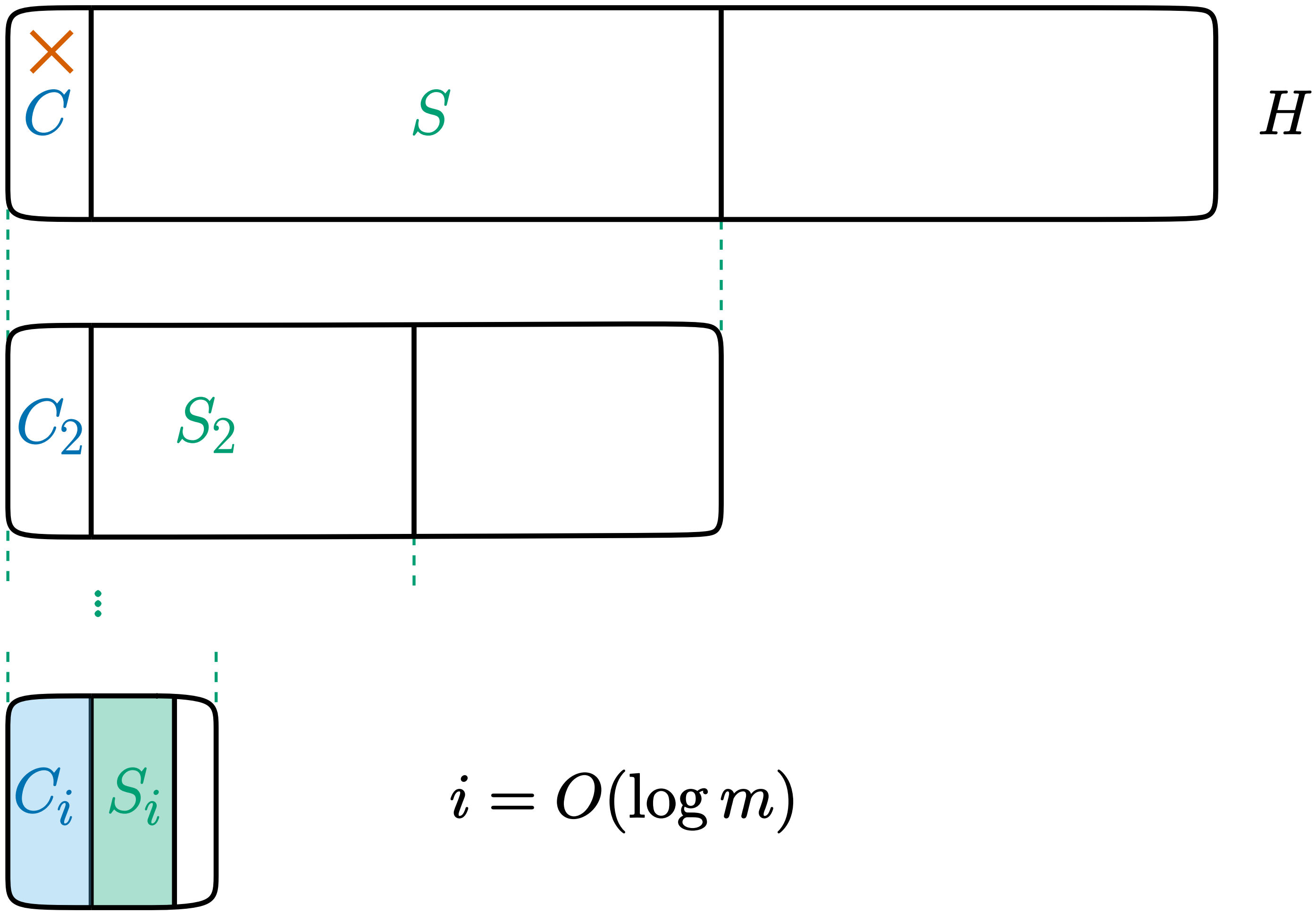
Decremental Algorithm

The algorithm of [OST23]:



Decremental Algorithm

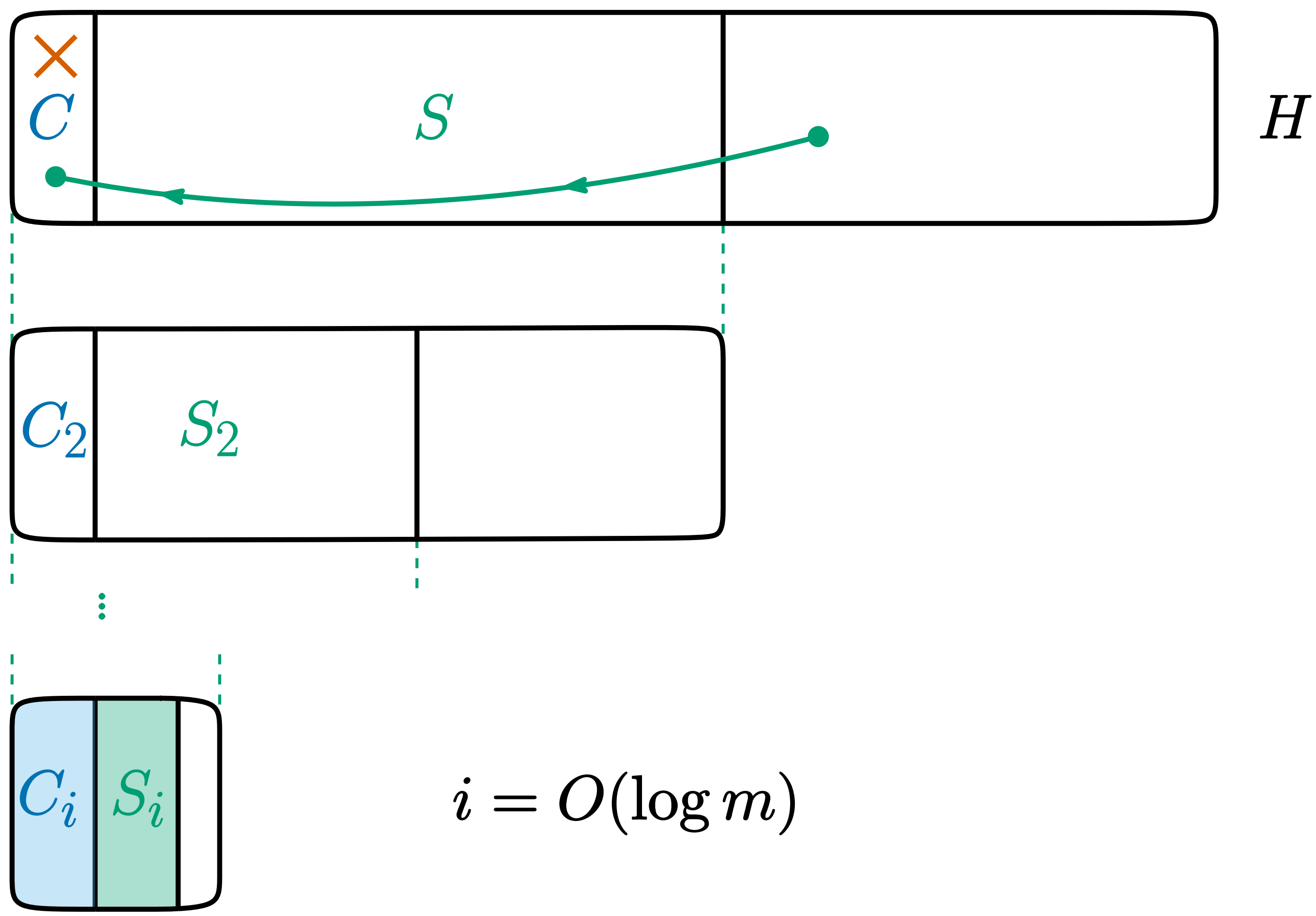
The algorithm of [OST23]:



$i = O(\log m)$

Decremental Algorithm

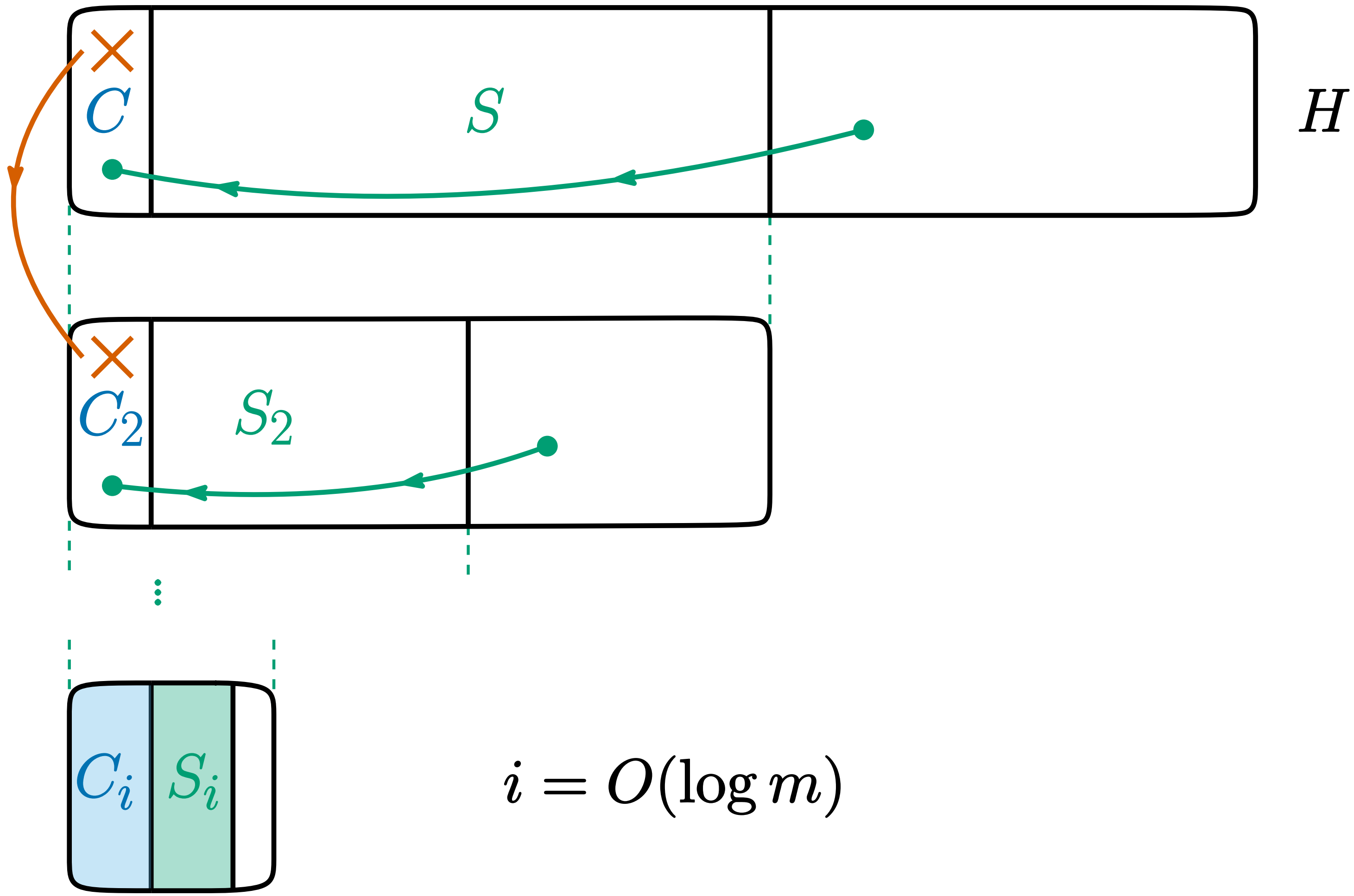
The algorithm of [OST23]:



$i = O(\log m)$

Decremental Algorithm

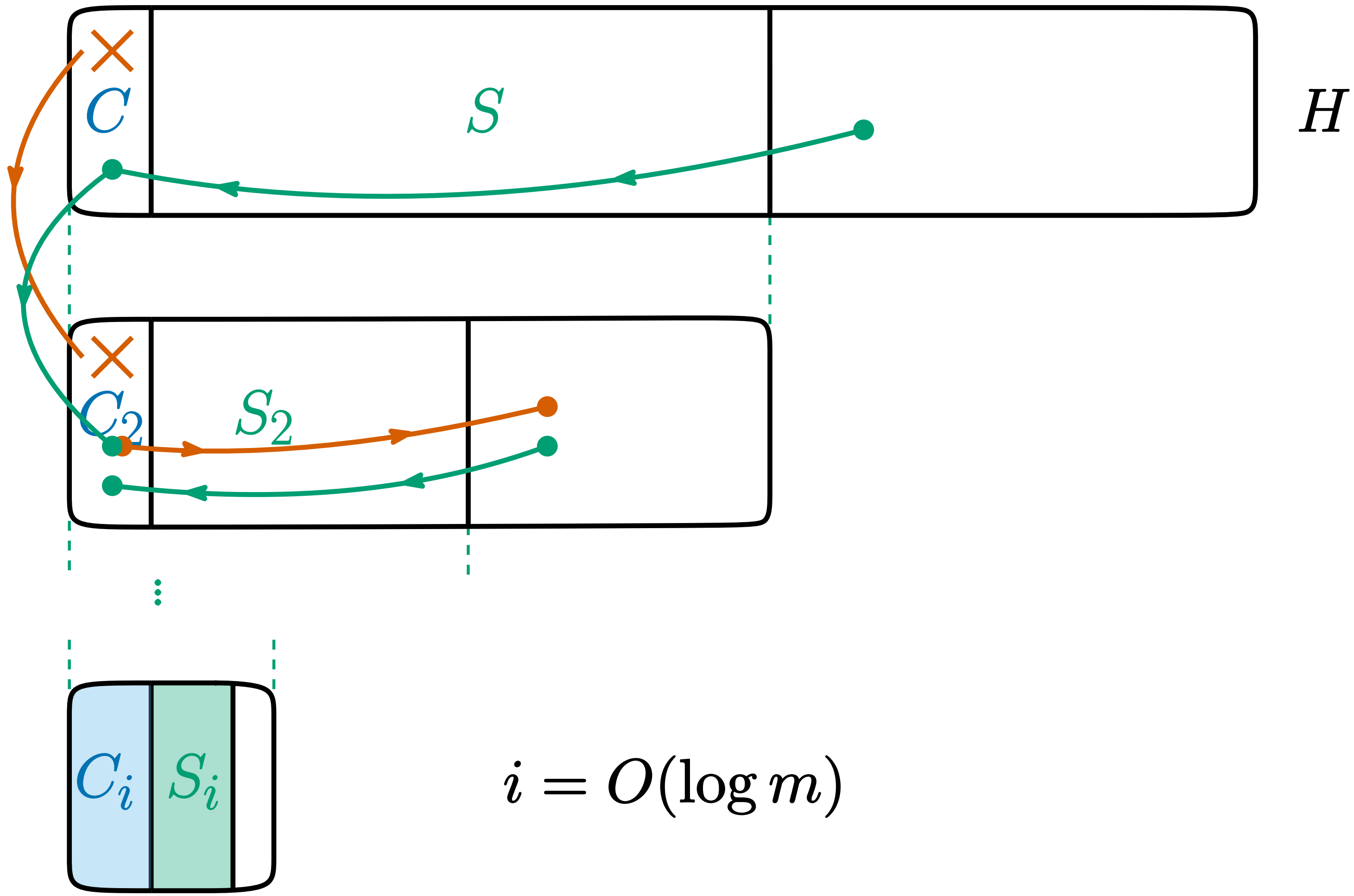
The algorithm of [OST23]:



$i = O(\log m)$

Decremental Algorithm

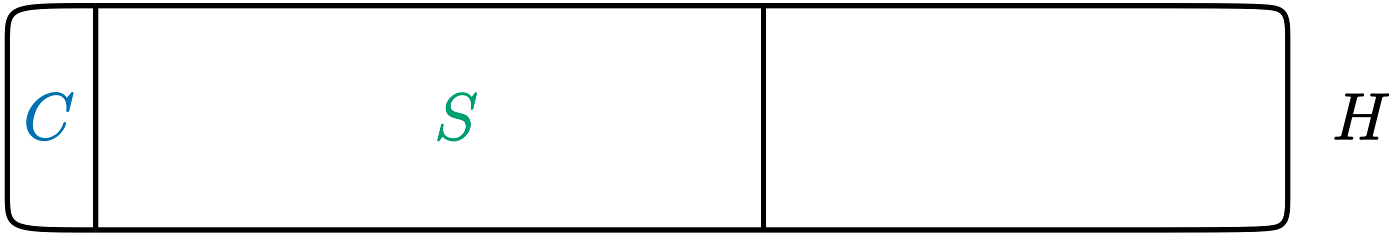
The algorithm of [OST23]:



$i = O(\log m)$

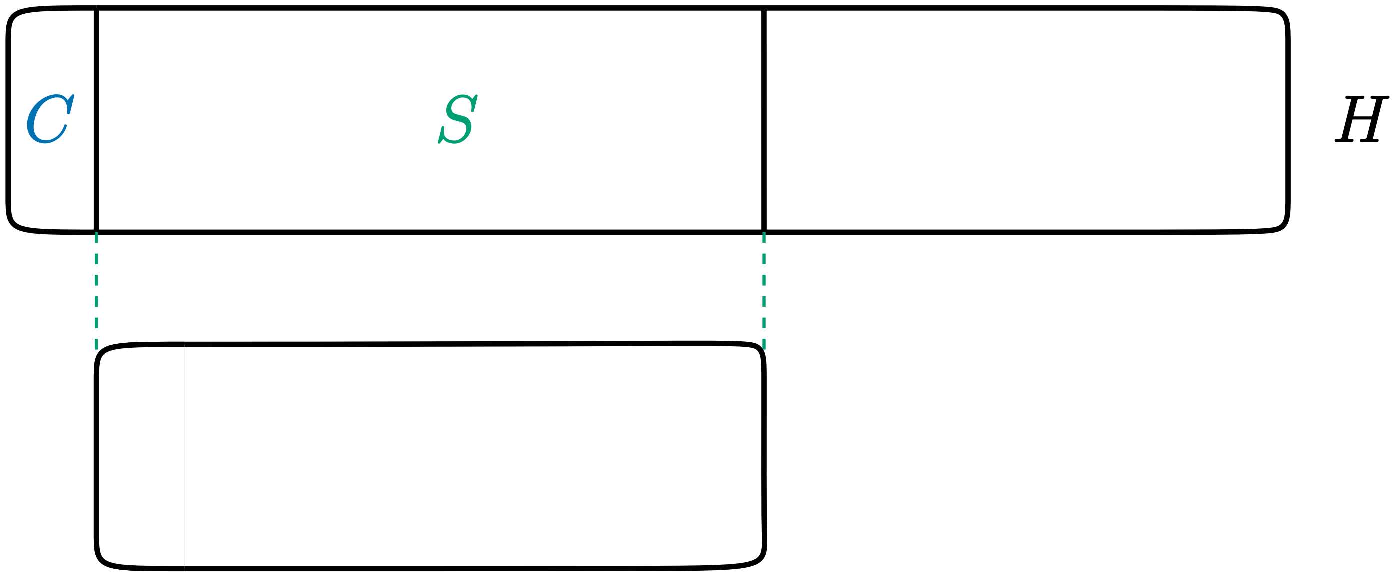
Decremental Algorithm

Our modification: similar to [ADKKP16] for graphs



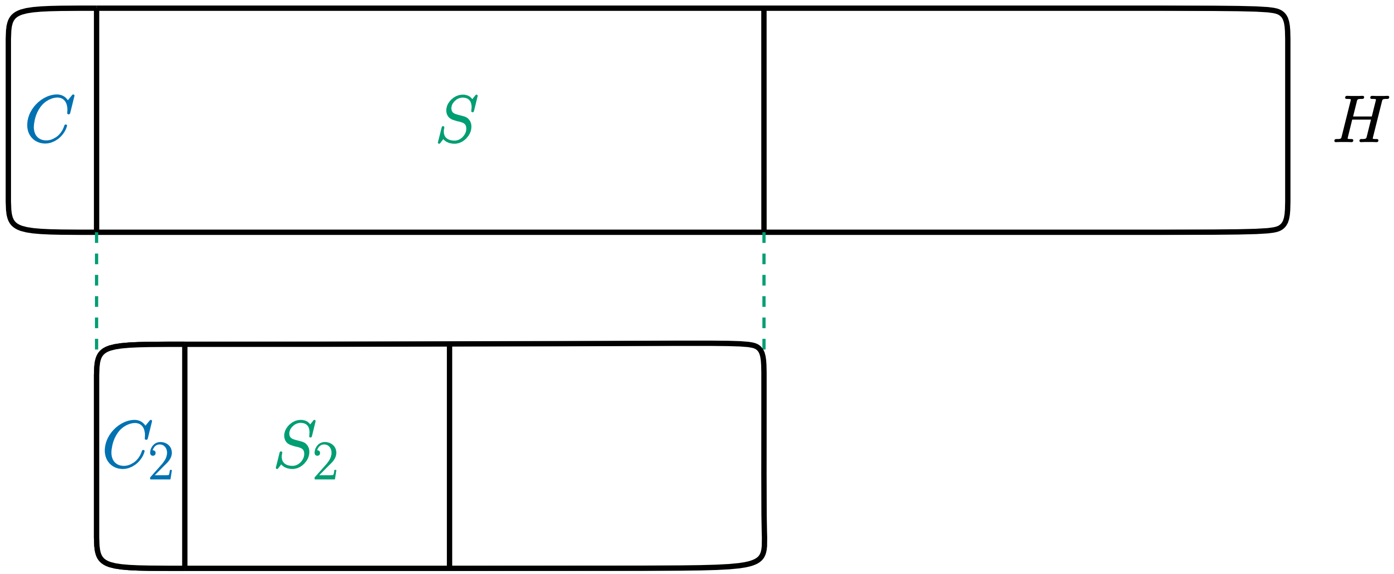
Decremental Algorithm

Our modification: similar to [ADKKP16] for graphs



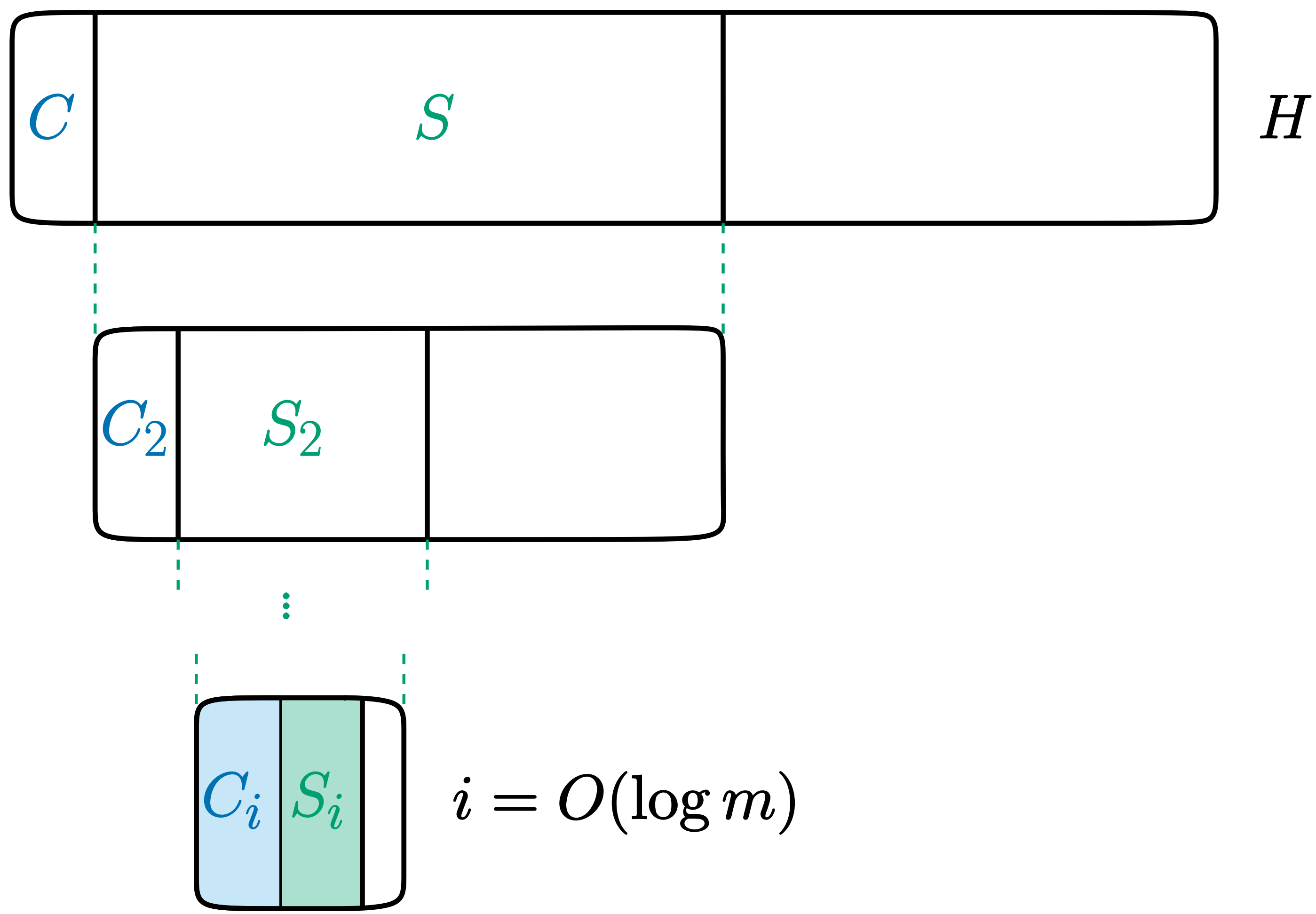
Decremental Algorithm

Our modification: similar to [ADKKP16] for graphs



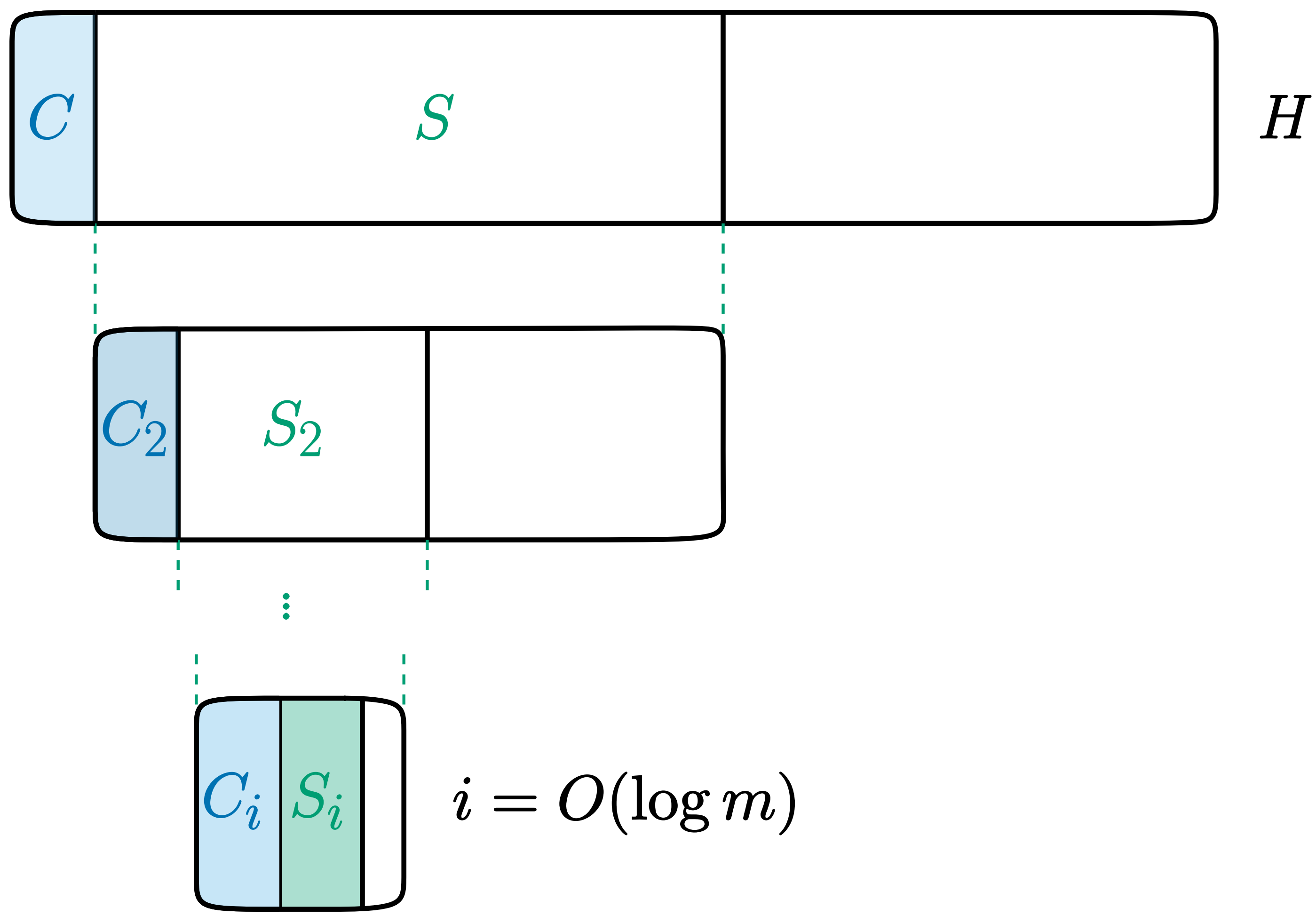
Decremental Algorithm

Our modification: similar to [ADKKP16] for graphs



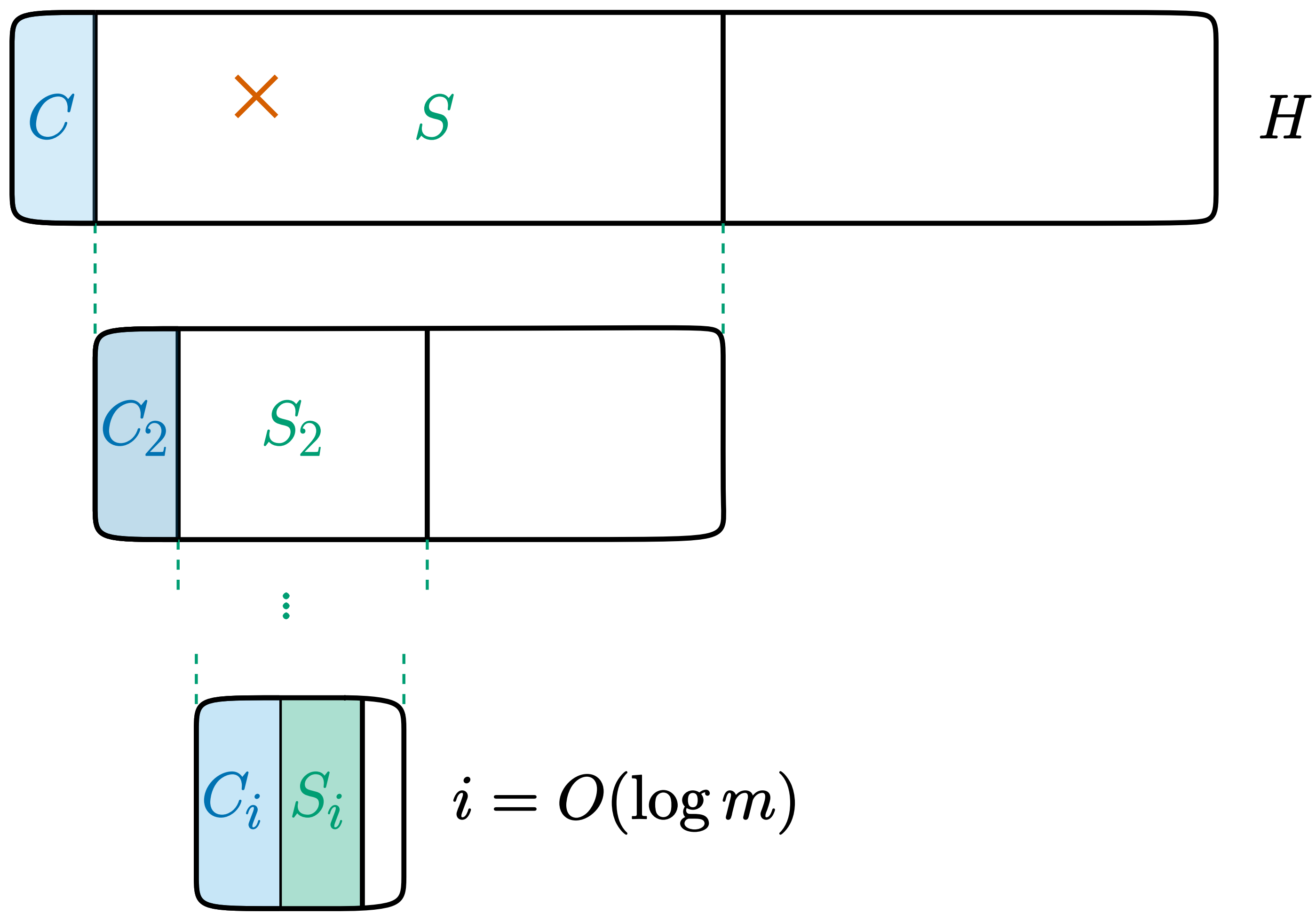
Decremental Algorithm

Our modification: similar to [ADKKP16] for graphs



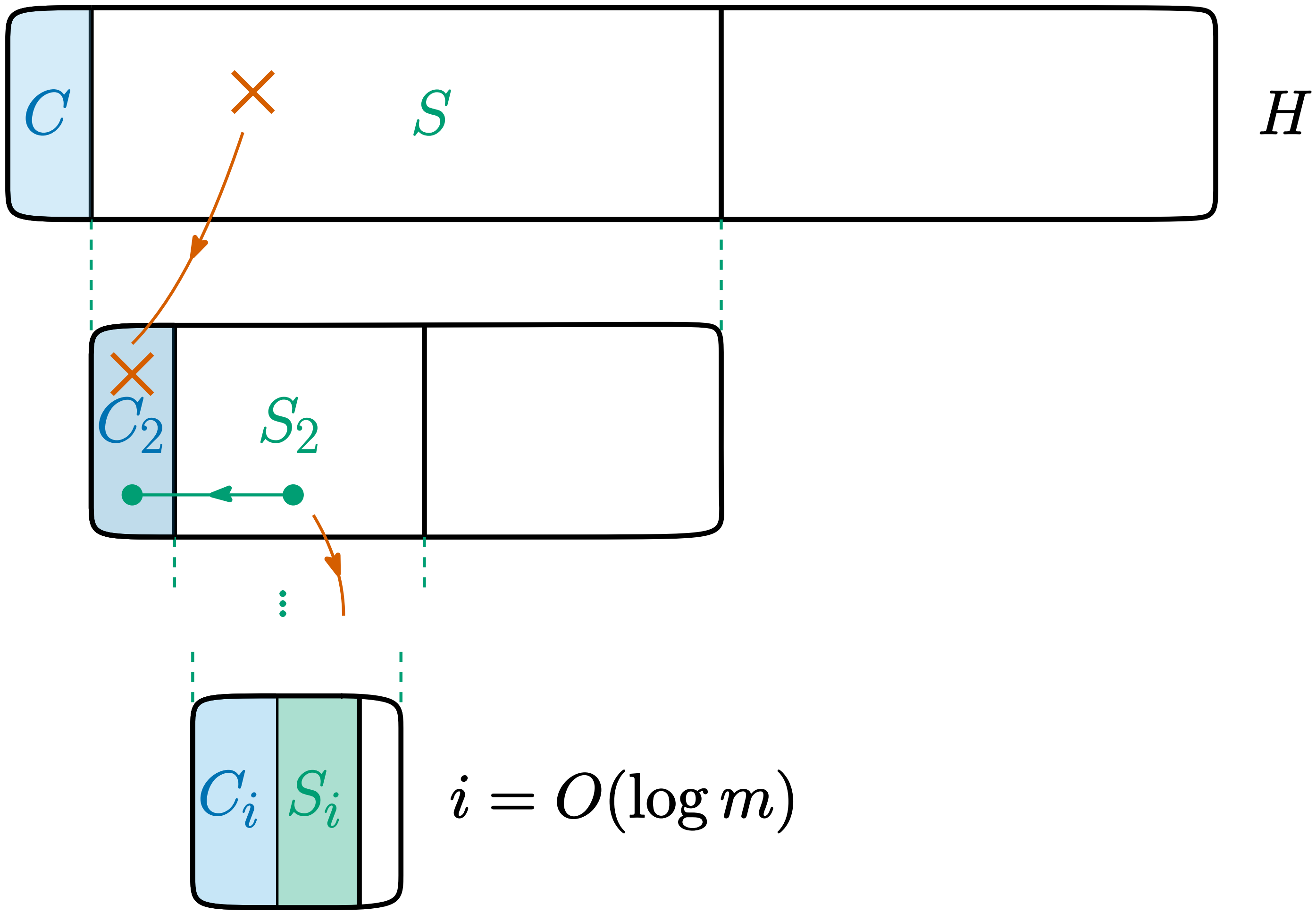
Decremental Algorithm

Our modification: similar to [ADKKP16] for graphs



Decremental Algorithm

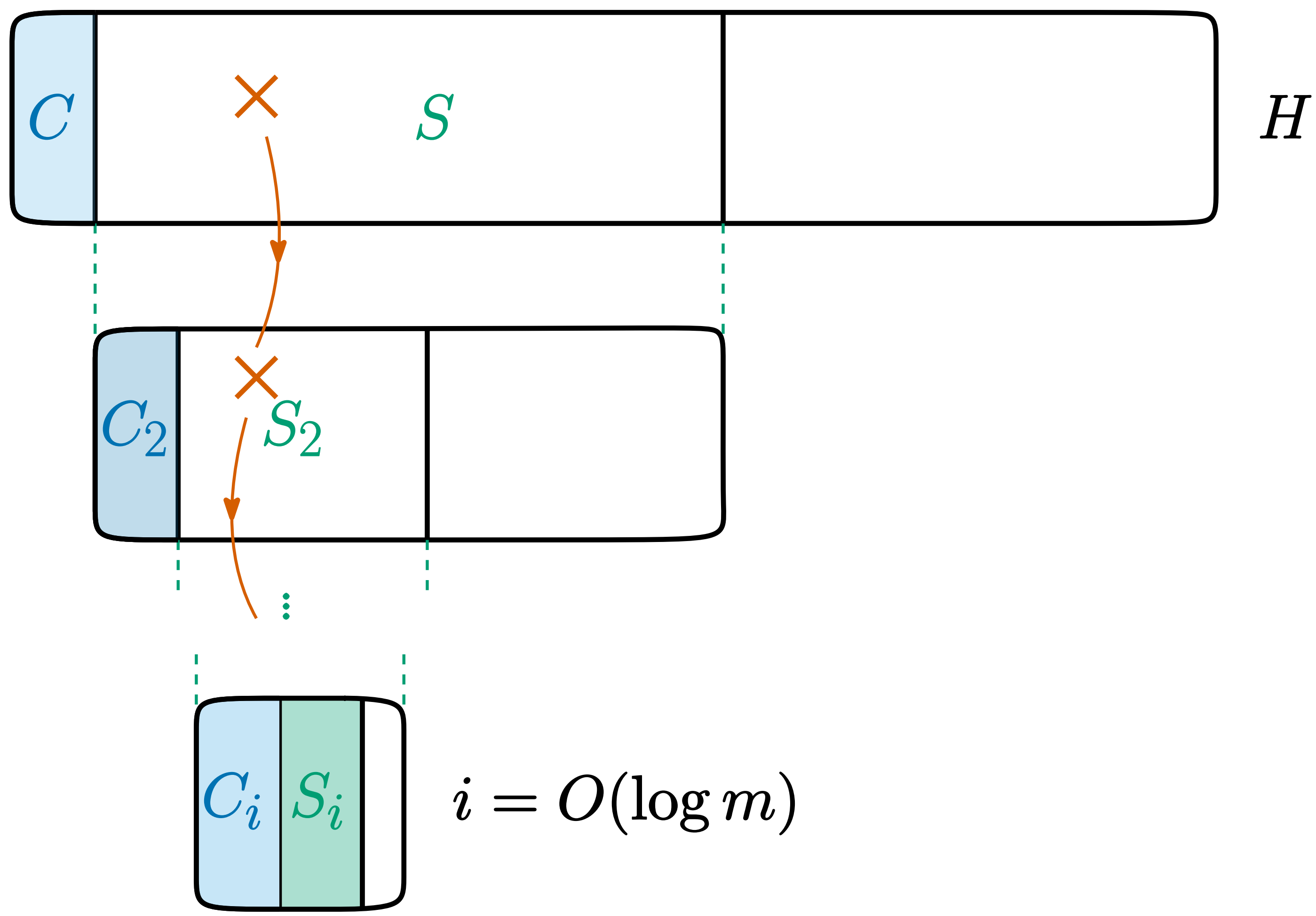
Our modification: similar to [ADKKP16] for graphs



$i = O(\log m)$

Decremental Algorithm

Our modification: similar to [ADKKP16] for graphs



References

[ADKKP16] ttai Abraham, David Durfee, Ioannis Koutis, Sebastian Krinninger, and Richard Peng.
On Fully Dynamic Graph Sparsifiers. FOCS, 2016.

[JLS23] Arun Jambulapati, Yang P. Liu, and Aaron Sidford.
Chaining, Group Leverage Score Overestimates, and Fast Spectral Hypergraph Sparsification. STOC, 2023.

[JSSTZ25] Arun Jambulapati, Sushant Sachdeva, Aaron Sidford, Kevin Tian, and Yibin Zhao.
Eulerian Graph Sparsification by Effective Resistance Decomposition. SODA, 2025.

[Lee23] James R. Lee.
Spectral Hypergraph Sparsification via Chaining. STOC, 2023.

[NR13] Ilan Newman and Yuri Rabinovich.
On Multiplicative λ-Approximations and Some Geometric Applications. SIAM J. Comput, 2013.

[OST23] Kazusato Oko, Shinsaku Sakaue, and Shin-ichi Tanigawa.
Nearly Tight Spectral Sparsification of Directed Hypergraphs. ICALP, 2023.

[ST11] Daniel A. Spielman and Shang-Hua Teng.
Spectral Sparsification of Graphs. SIAM J. Comput, 2011.

[SS08] Daniel A. Spielman and Nikhil Srivastava.
Graph sparsification by effective resistances. STOC, 2008.

[SY19] Tasuku Soma and Yuichi Yoshida.
Spectral Sparsification of Hypergraphs. SODA, 2019.