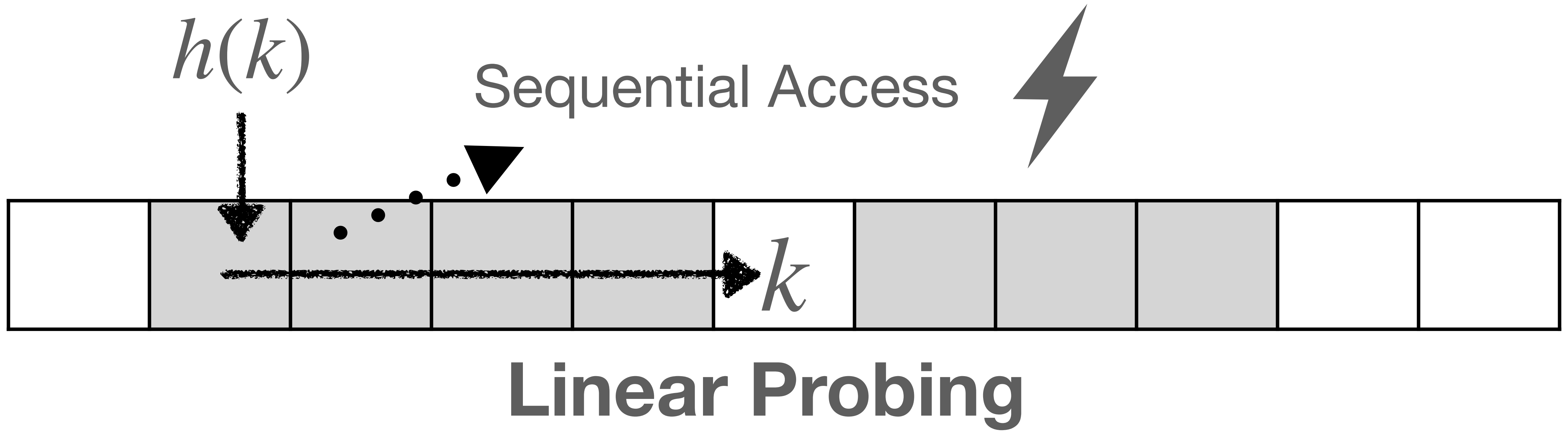
Zombie Hashing

Reanimating Tombstones in a Graveyard

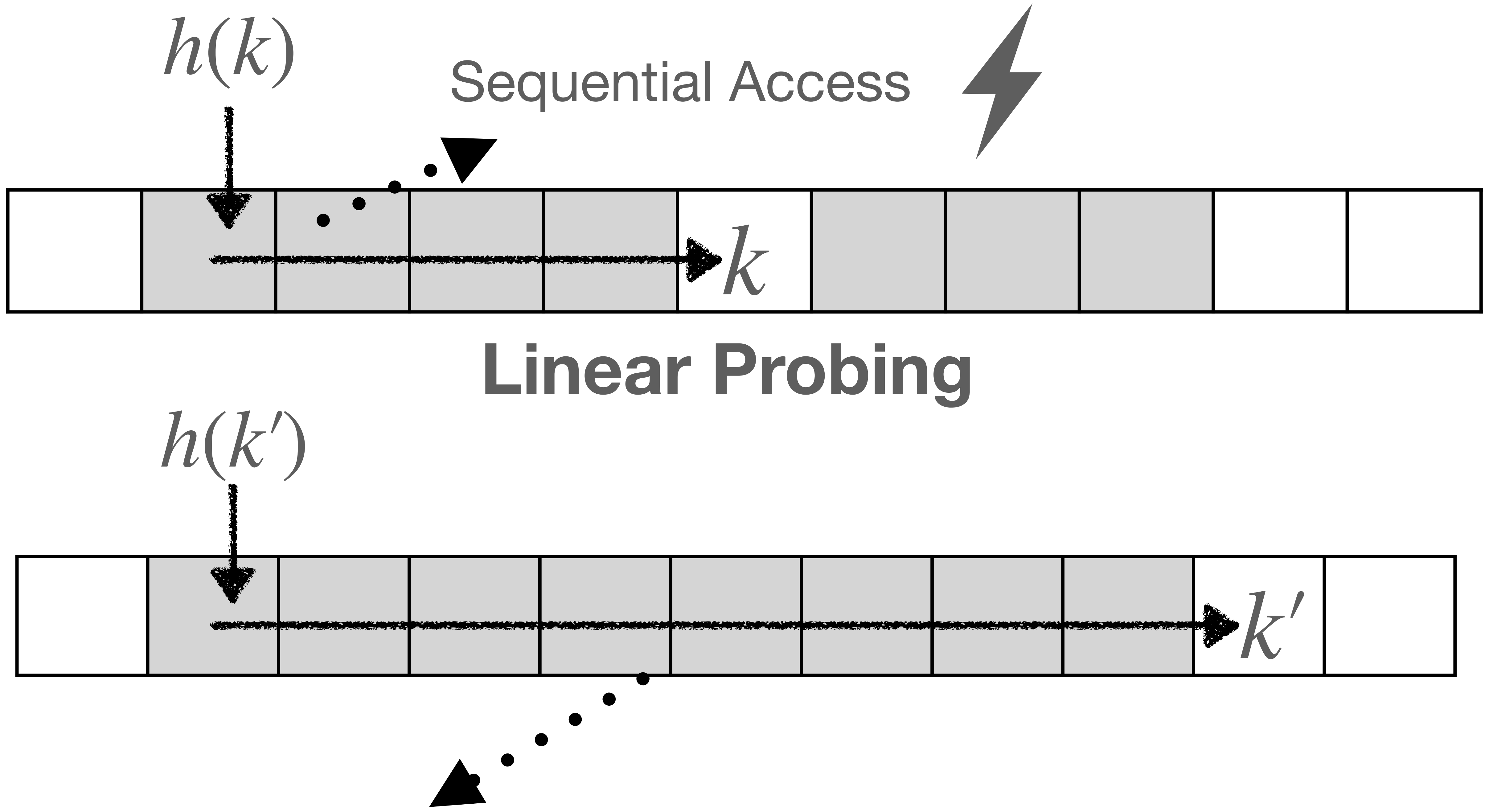
SIGMOD 2025

Yuvaraj Chesetti*, Benwei Shi*, Jeff M. Phillips, **Prashant Pandey**

Linear probing offers high cache locality



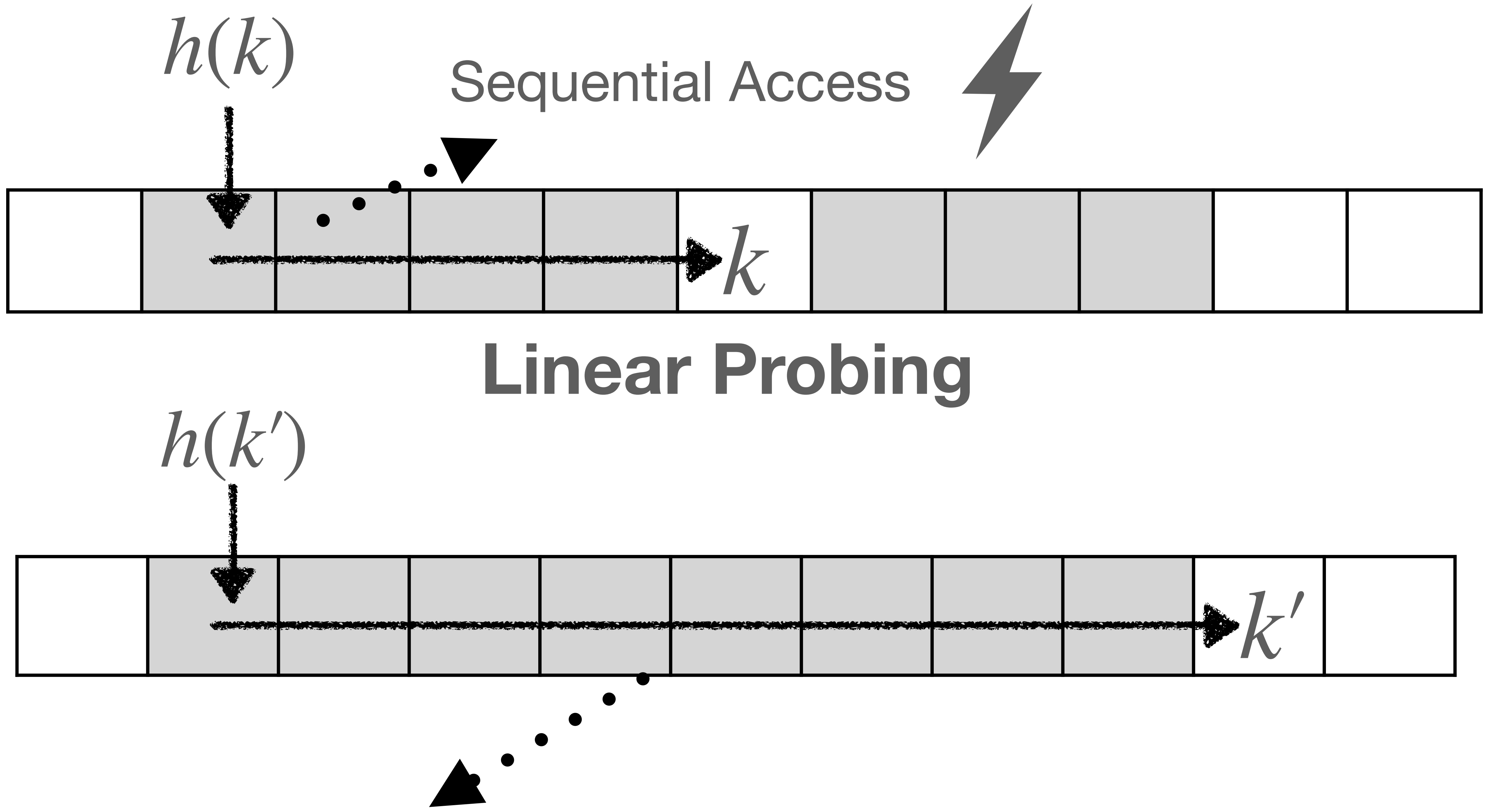
Primary clustering degrades performance



Primary Clustering

Clusters grow longer with more inserts

Primary clustering degrades performance



Linear Probing

Primary Clustering

Clusters grow longer with more inserts

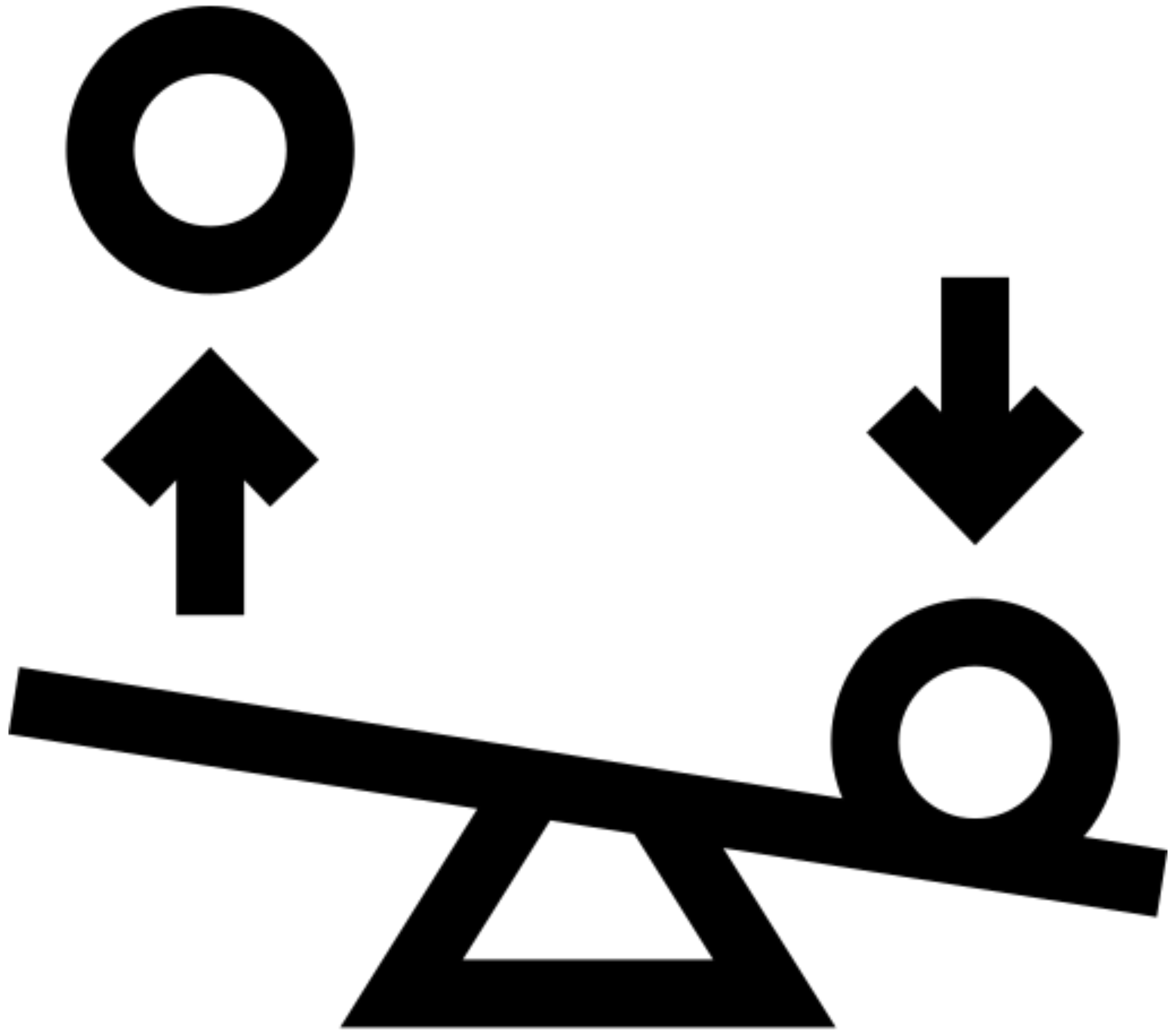
α = Load factor

$$\alpha = 1 - \frac{1}{x}$$

$$\Theta(x^2)$$

All operations (Knuth'93)

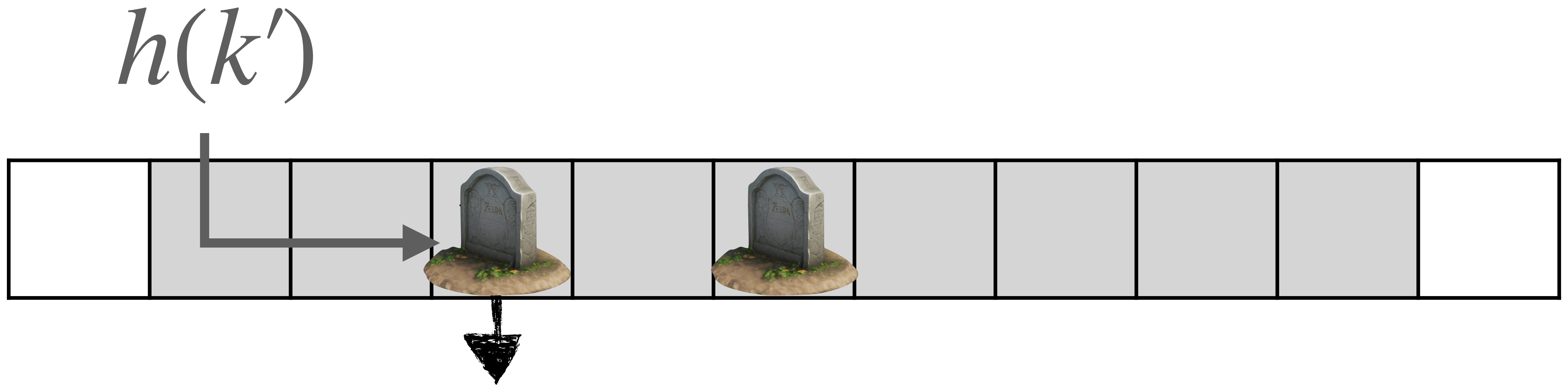
Linear probing suffers from performance trade-offs



Performance

Space efficiency

Tombstones can help avoid primary clustering

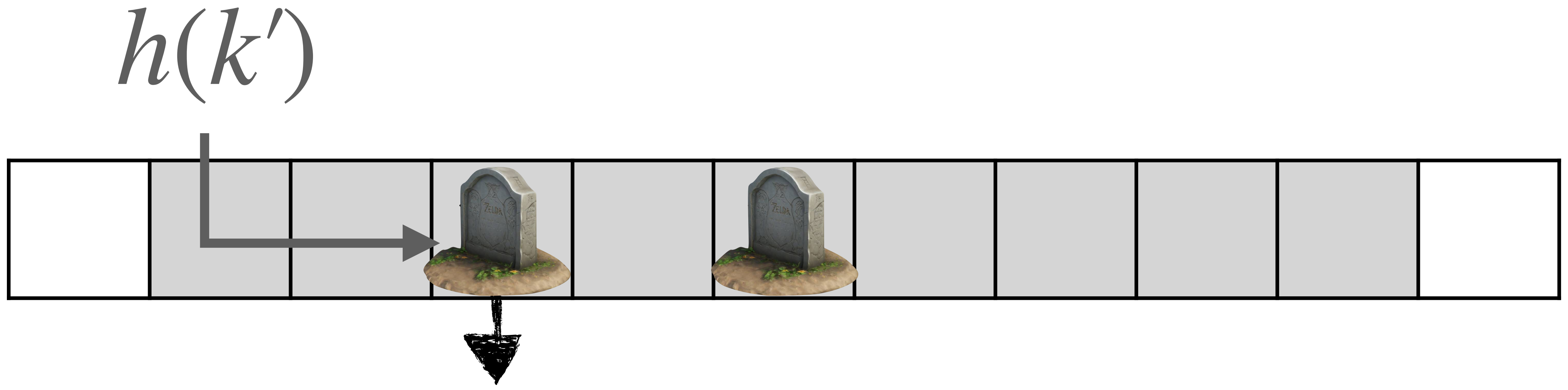


Tombstones
(deletion markers)

$\tilde{O}(x)$ [1]
(**amortized**)

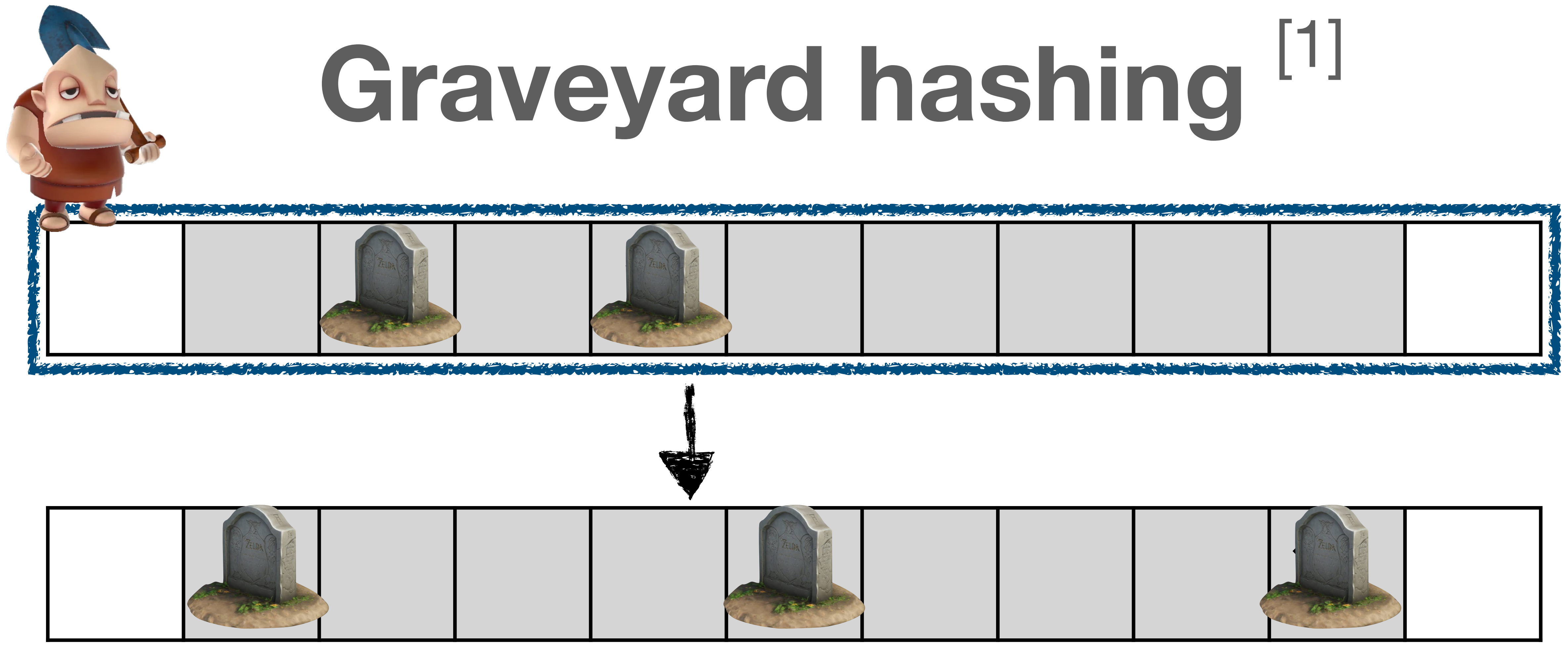
[1]: M. A. Bender, B. C. Kuszmaul and W. Kuszmaul, "Linear Probing Revisited: Tombstones Mark the Demise of Primary Clustering"

Tombstones can help avoid primary clustering



Tombstones
(deletion markers)

$\tilde{O}(x)$ [1]
(**amortized**)

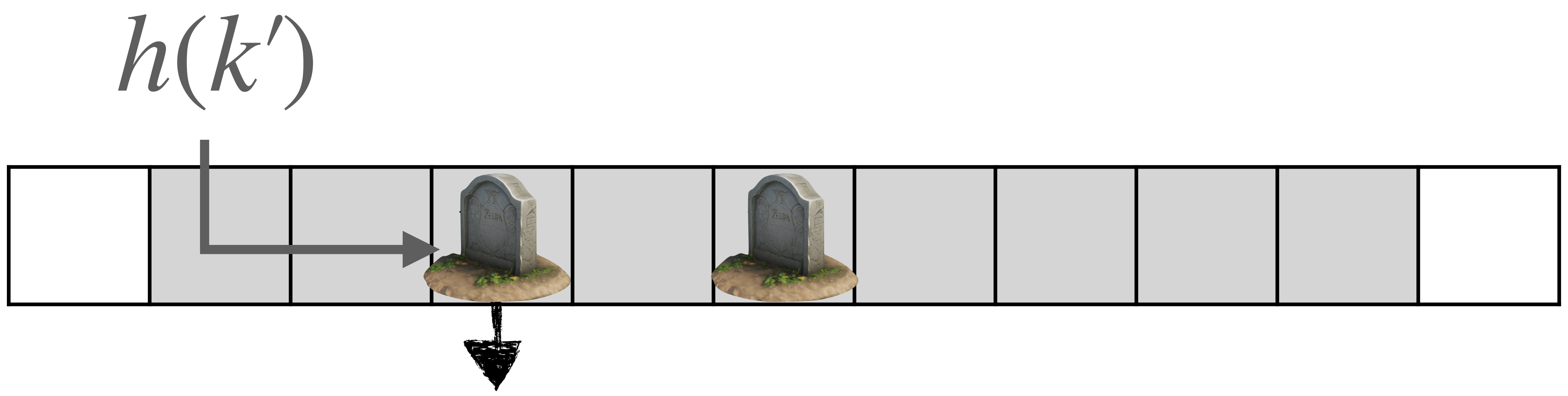


Graveyard hashing [1]

$O(x)$
(**amortized**)

[1]: M. A. Bender, B. C. Kuszmaul and W. Kuszmaul, "Linear Probing Revisited: Tombstones Mark the Demise of Primary Clustering"

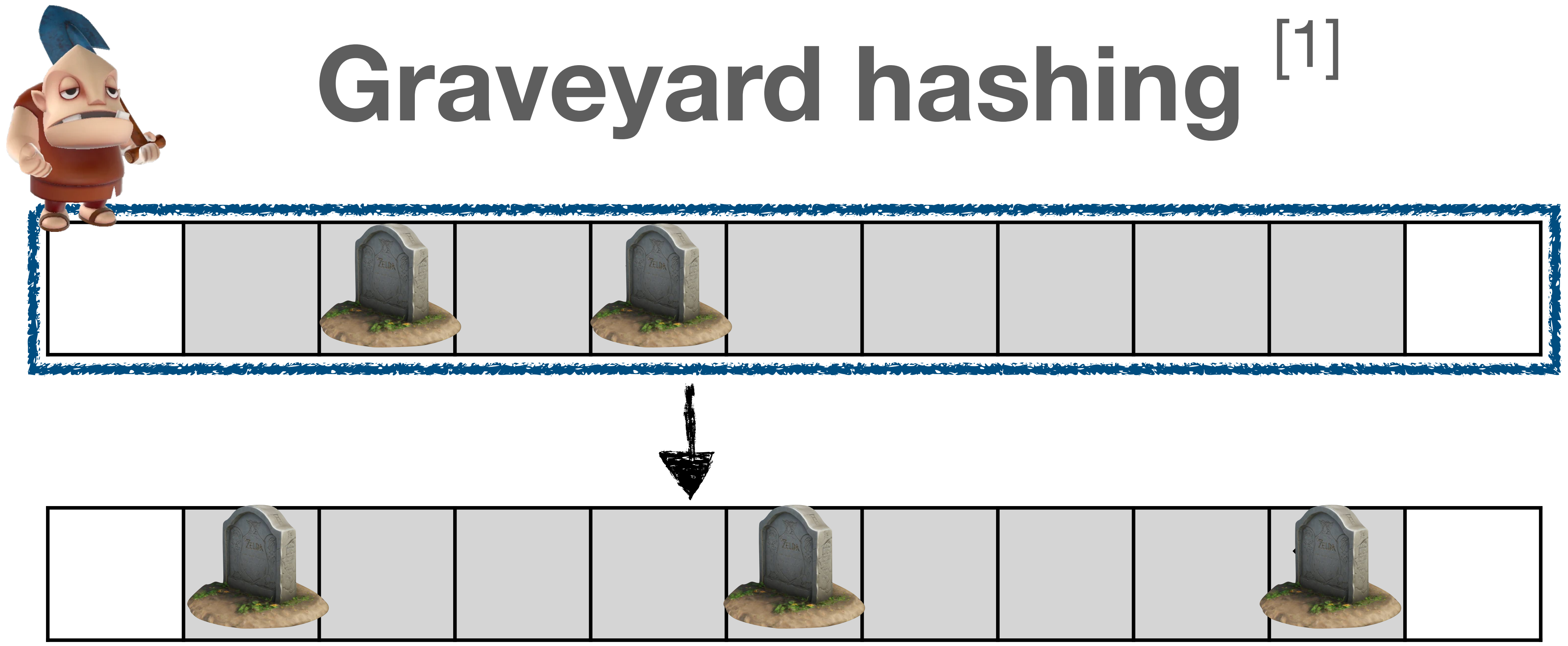
Avoiding primary clustering introduces trade-offs



Tombstones
(deletion markers)

$\tilde{O}(x)$ [1]
(**amortized**)

Graveyard hashing [1]

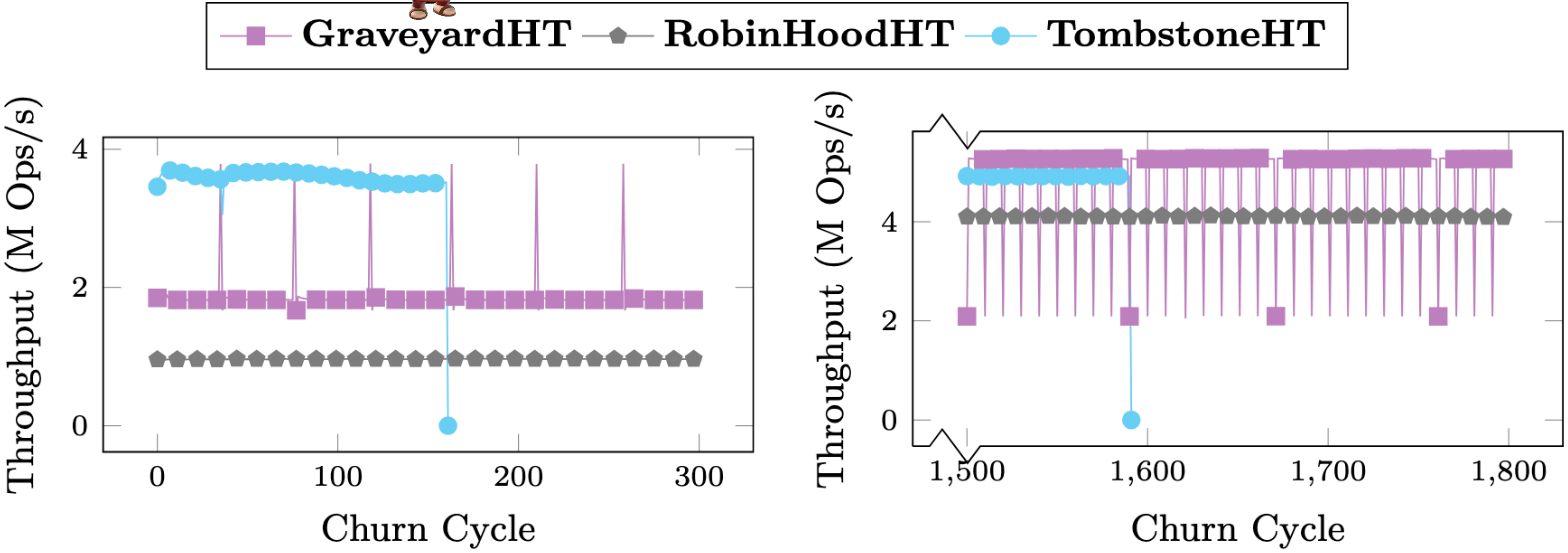


$O(n)$ to rebuild
High Latency!

$O(x)$
(**amortized**)

[1]: M. A. Bender, B. C. Kuszmaul and W. Kuszmaul, "Linear Probing Revisited: Tombstones Mark the Demise of Primary Clustering"

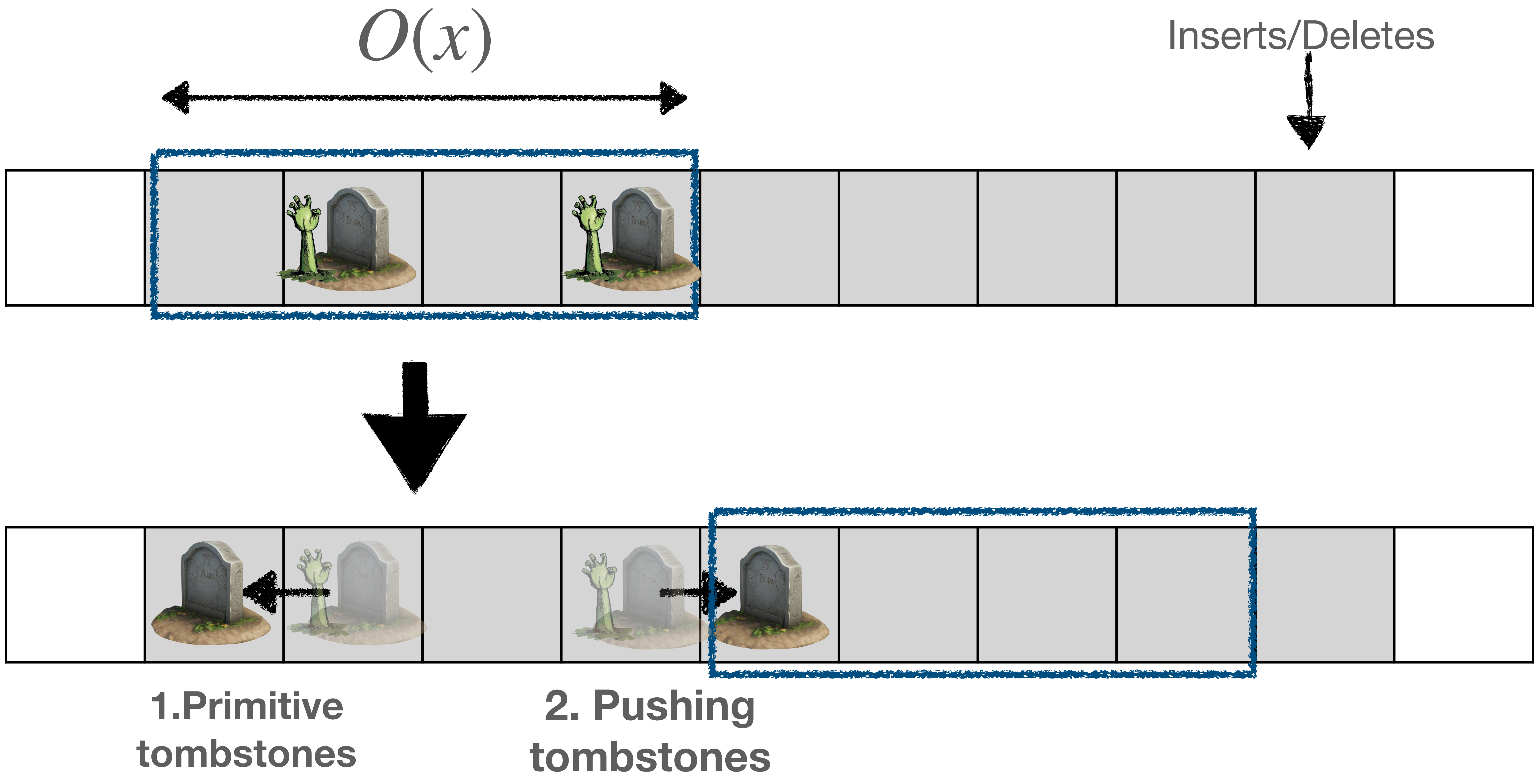
Graveyard hashing has regular performance drops



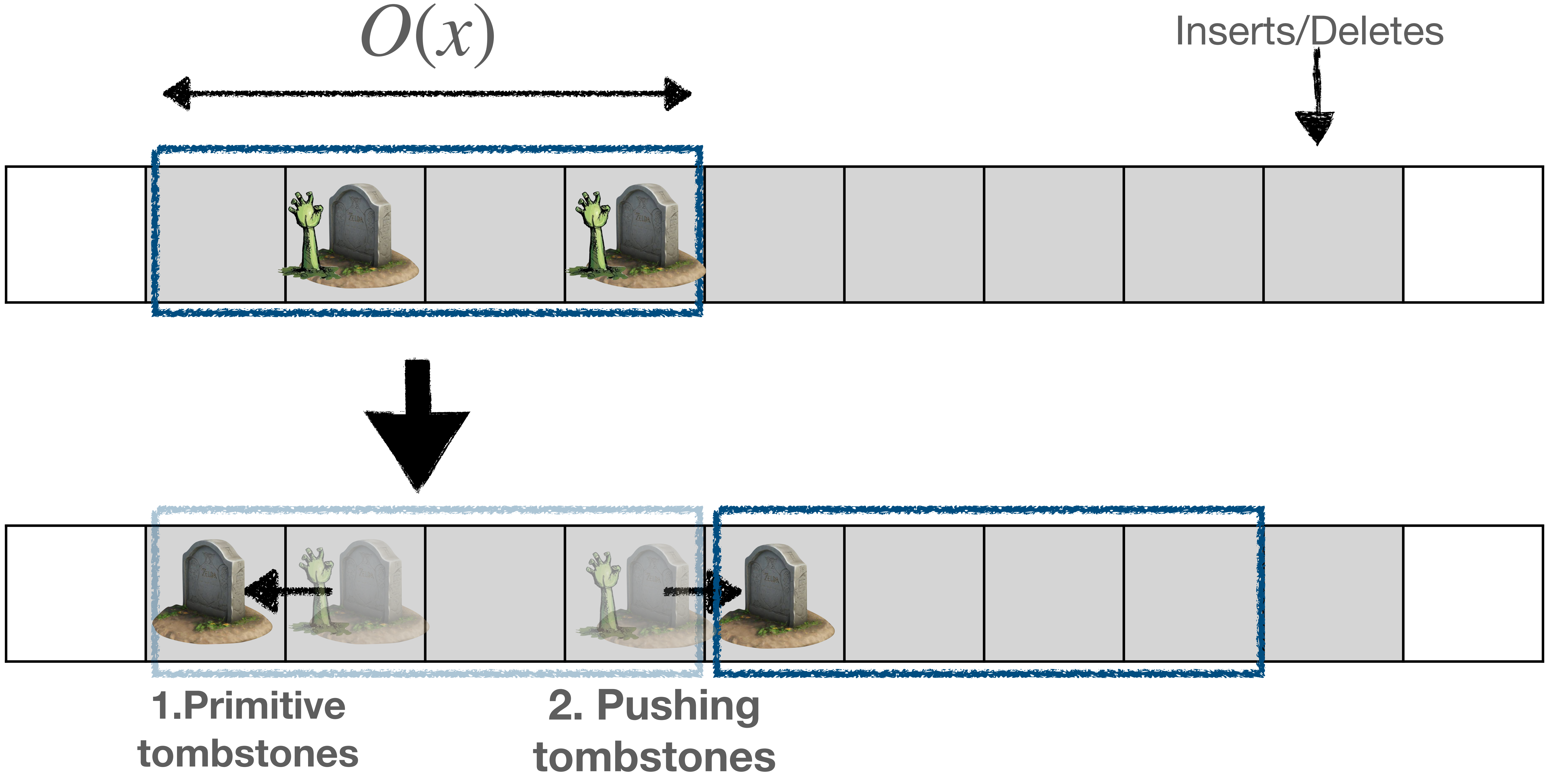
(a) 50% updates and 50% queries.

(b) 5% updates and 95% queries.

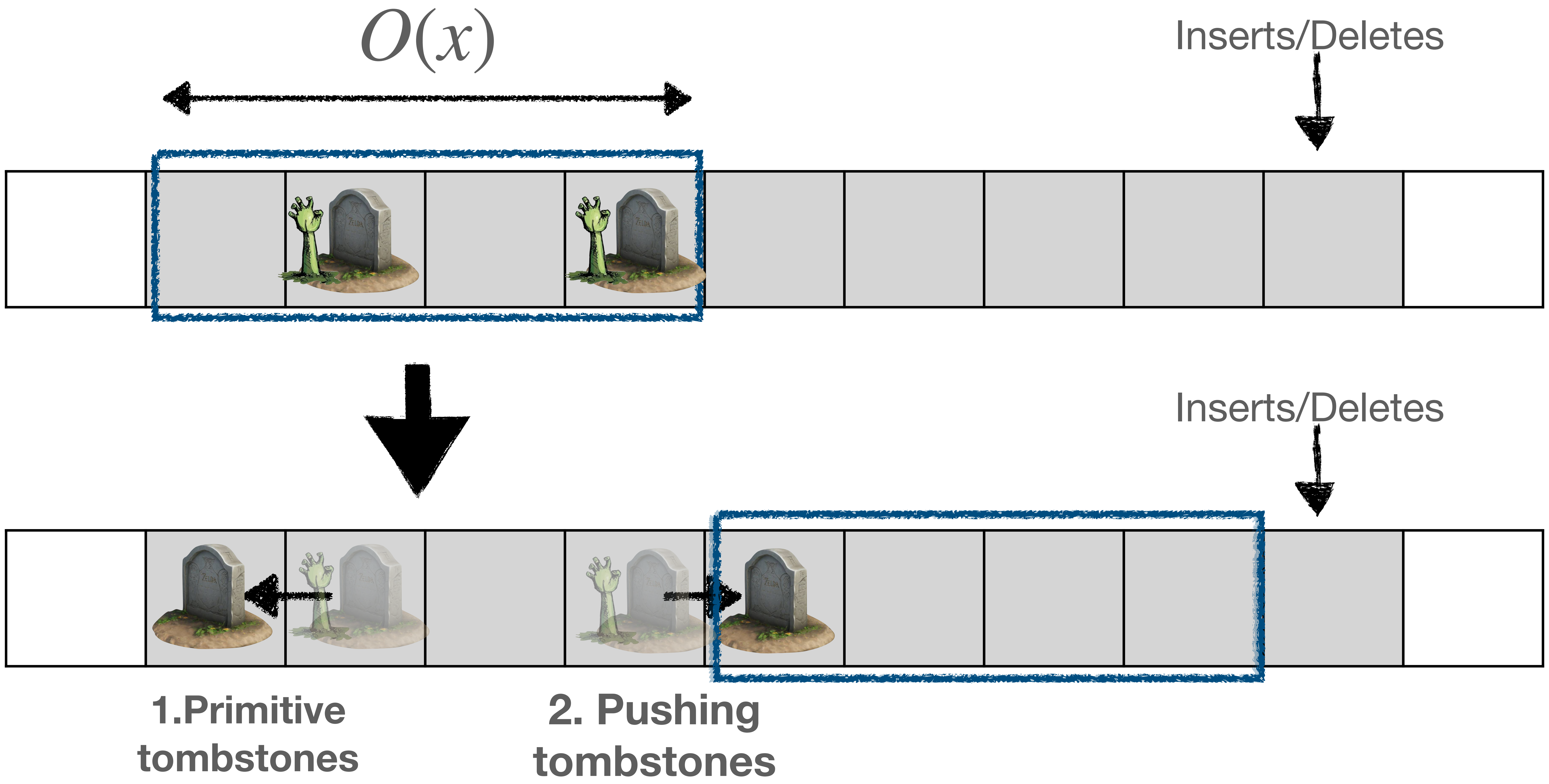
ZombieHT rebalances in windows



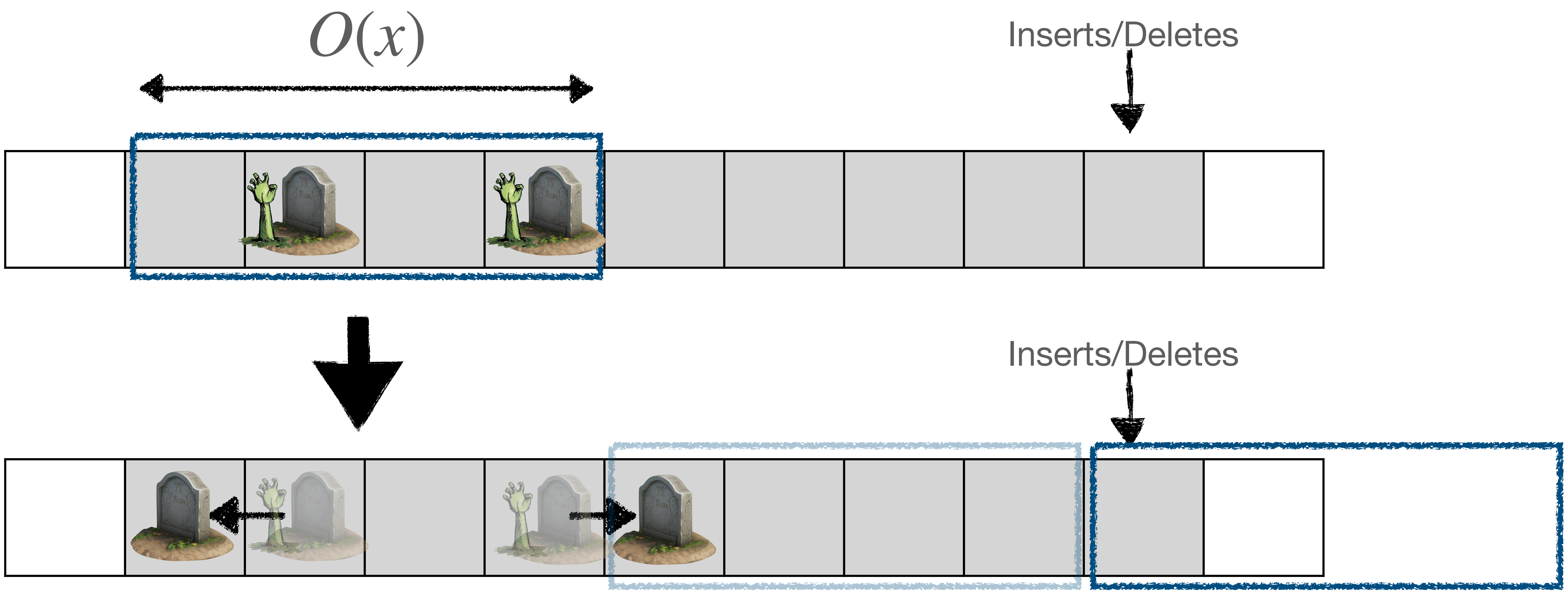
ZombieHT rebalances in windows



ZombieHT rebalances in windows



ZombieHT rebalances in windows

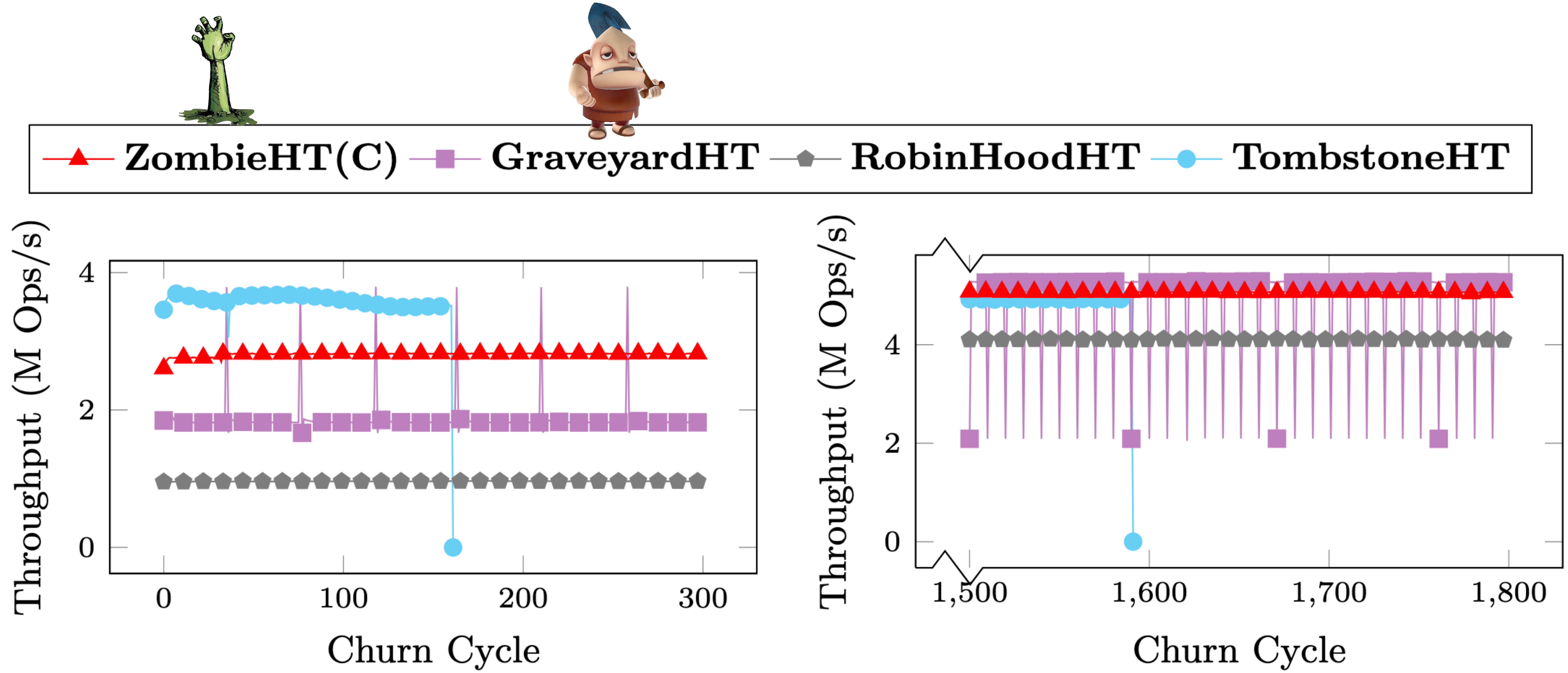


1.Primitive tombstones

2. Pushing tombstones

$O(x)$ **every operation!**

ZombieHT achieves consistent, high performance



(a) 50% updates and 50% queries.

(b) 5% updates and 95% queries.

ZombieHT takeaways

- $O(x)$ theoretical bound for each operation in the deamortization scheme
- Two variants: Compact (quotienting-based) and vectorized (Absl-based)
- Consistent performance with high space efficiency

Source code: https://github.com/saltsystemslab/GRHT