Syracuse University
College of Engineering
& Computer Science

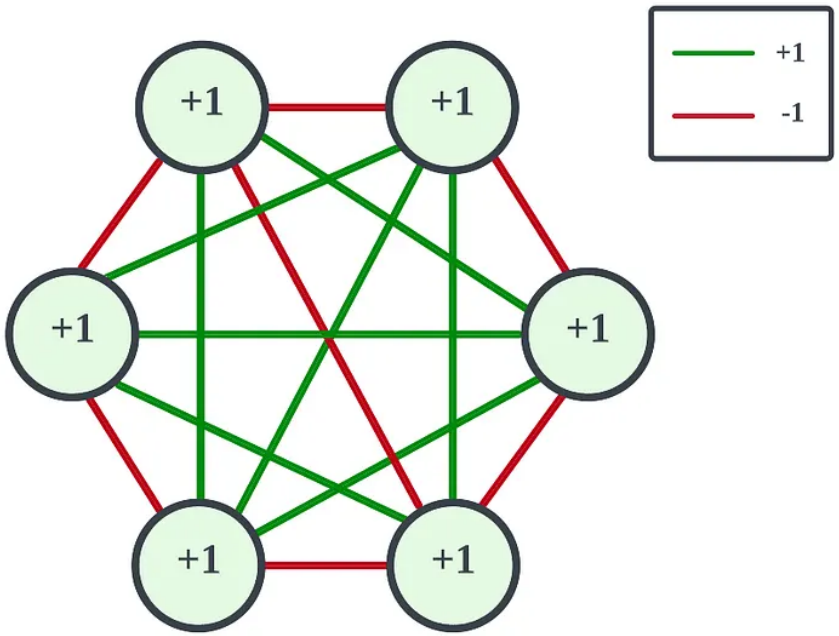
Linearithmic Clean-up for Vector-Symbolic Key-Value Memory with Kroneker Rotation Products

(Accepted by NeSy 2025)

Ruipeng Liu[1] Qinru Qiu[1] Simon Khan[2] Garrett E. Katz[1]

[1]Syracuse University [2]Air Force Research Laboratory

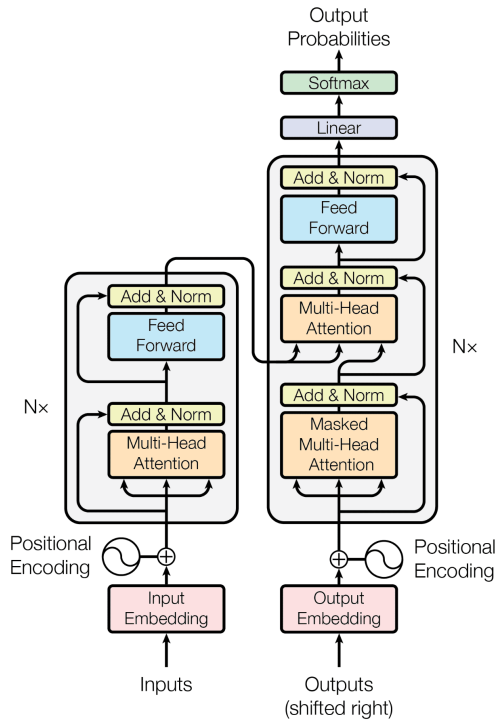
Differentiable Data Structures and Memory



Model 1

Hopfield Network [1], figure from here

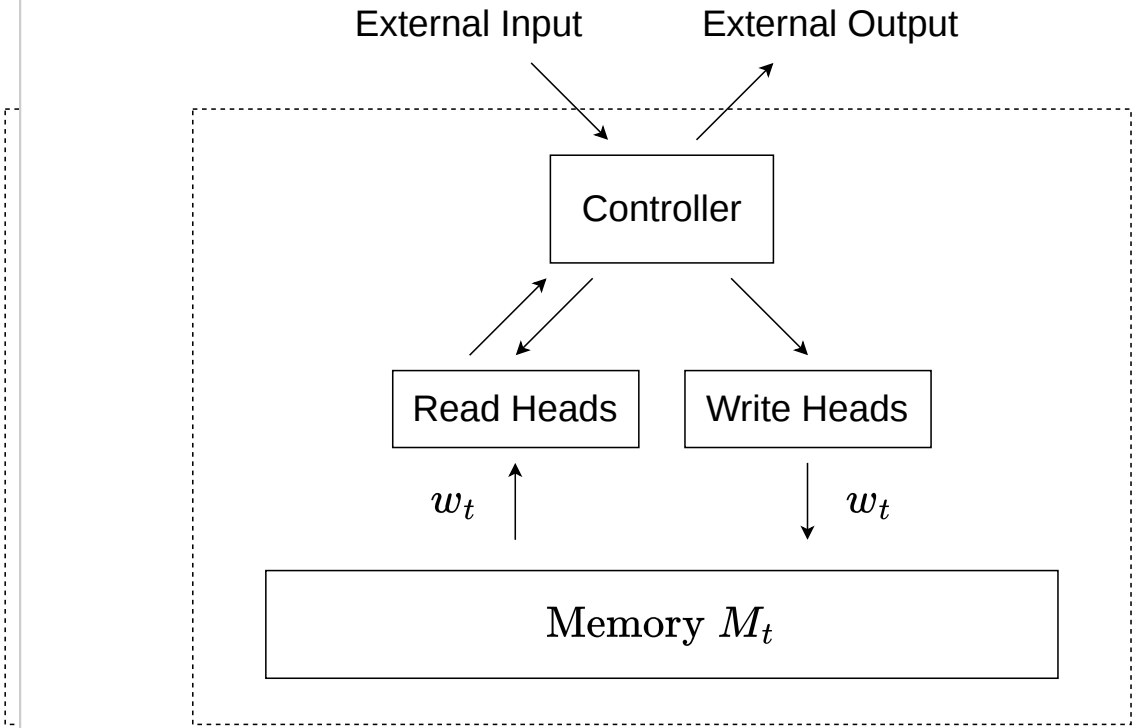
Differentiable Data Structures and Memory



Model 2

Transformer [2], figure from here

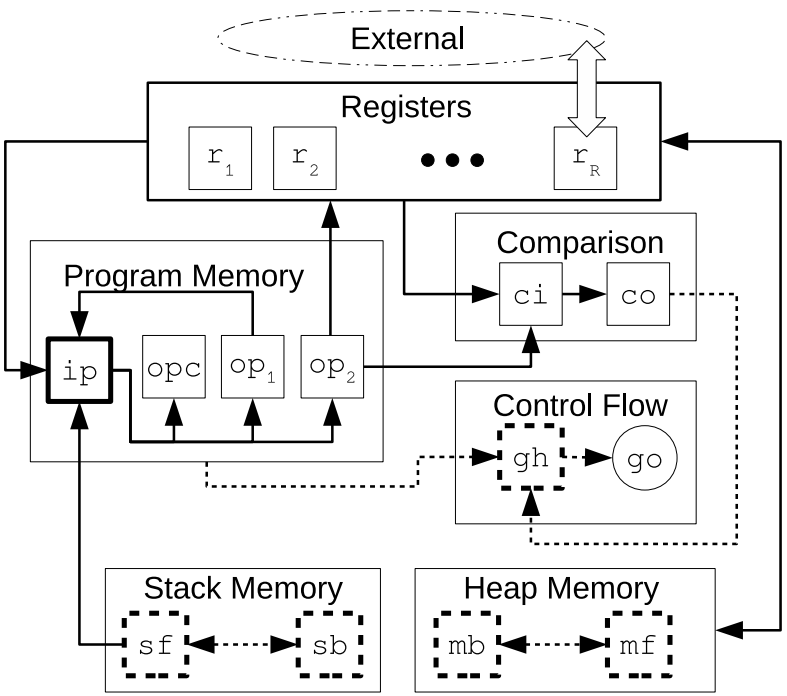
Differentiable Data Structures and Memory



Model 3

Neural Turing Machine [3]

Differentiable Data Structures and Memory



Model 4

Neural Virtual Machine [4]

Dive into NVM's Memory

Memory of NVM

Memory: $M \in \mathbb{R}^{N \times N}$

Read: $v \leftarrow \sigma(Ma)$

Write: $M \leftarrow M - \overbrace{\sigma(Ma)a^\top/N}^{\text{Erase}} + \overbrace{va^\top/N}^{\text{Write}}$

Memory of NVM

Memory: $M \in \mathbb{R}^{N \times N}$

Read: $v \leftarrow \sigma(Ma)$

Write: $M \leftarrow M - \overbrace{\sigma(Ma)a^\top/N}^{\text{Erase}} + \overbrace{va^\top/N}^{\text{Write}}$

Memory of NVM

Memory: $M \in \mathbb{R}^{N \times N}$

Read: $v \leftarrow \sigma(Ma)$

Write: $M \leftarrow M - \overbrace{\sigma(Ma)a^\top/N}^{\text{Erase}} + \overbrace{va^\top/N}^{\text{Write}}$

Memory of NVM

Memory: $M \in \mathbb{R}^{N \times N}$

Read: $v \leftarrow \sigma(Ma)$

Write: $M \leftarrow M - \overbrace{\sigma(Ma)a^\top/N}^{\text{Erase}} + \overbrace{va^\top/N}^{\text{Write}}$

Memory of NVM

Memory: $M \in \mathbb{R}^{N \times N}$

Read: $v \leftarrow \sigma(Ma)$

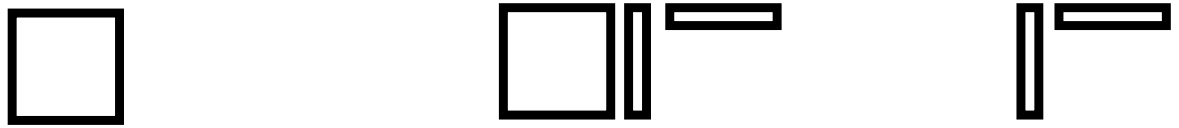
Write: $M \leftarrow M - \overbrace{\sigma(Ma)a^\top/N}^{\text{Erase}} + \overbrace{va^\top/N}^{\text{Write}}$

Memory of NVM

Memory: $M \in \mathbb{R}^{N \times N}$

Read: $v \leftarrow \sigma(Ma)$

Write: $M \leftarrow M - \overbrace{\sigma(Ma)a^\top/N}^{\text{Erase}} + \overbrace{va^\top/N}^{\text{Write}}$



Two Problems

Capacity is N

$\Downarrow$

Single-Pass Full-Capacity Learning Rule

$\Downarrow$

ICML'24 [5] (**Does Not Exist**)

Quadratic Runtime

$\Downarrow$

Vector-**S**ymbolic **A**rchitecture (VSA) formed Memory

$\Downarrow$

NeSy'25 [6] (This Talk)

Two Problems

Capacity is N

$\Downarrow$

Single-Pass Full-Capacity Learning Rule

$\Downarrow$

ICML'24 [5] (**Does Not Exist**)

Quadratic Runtime

$\Downarrow$

Vector-**S**ymbolic **A**rchitecture (VSA) formed Memory

$\Downarrow$

NeSy'25 [6] (This Talk)

Two Problems

Capacity is N

$\Downarrow$

Single-Pass Full-Capacity Learning Rule

$\Downarrow$

ICML'24 [5] (**Does Not Exist**)

Quadratic Runtime

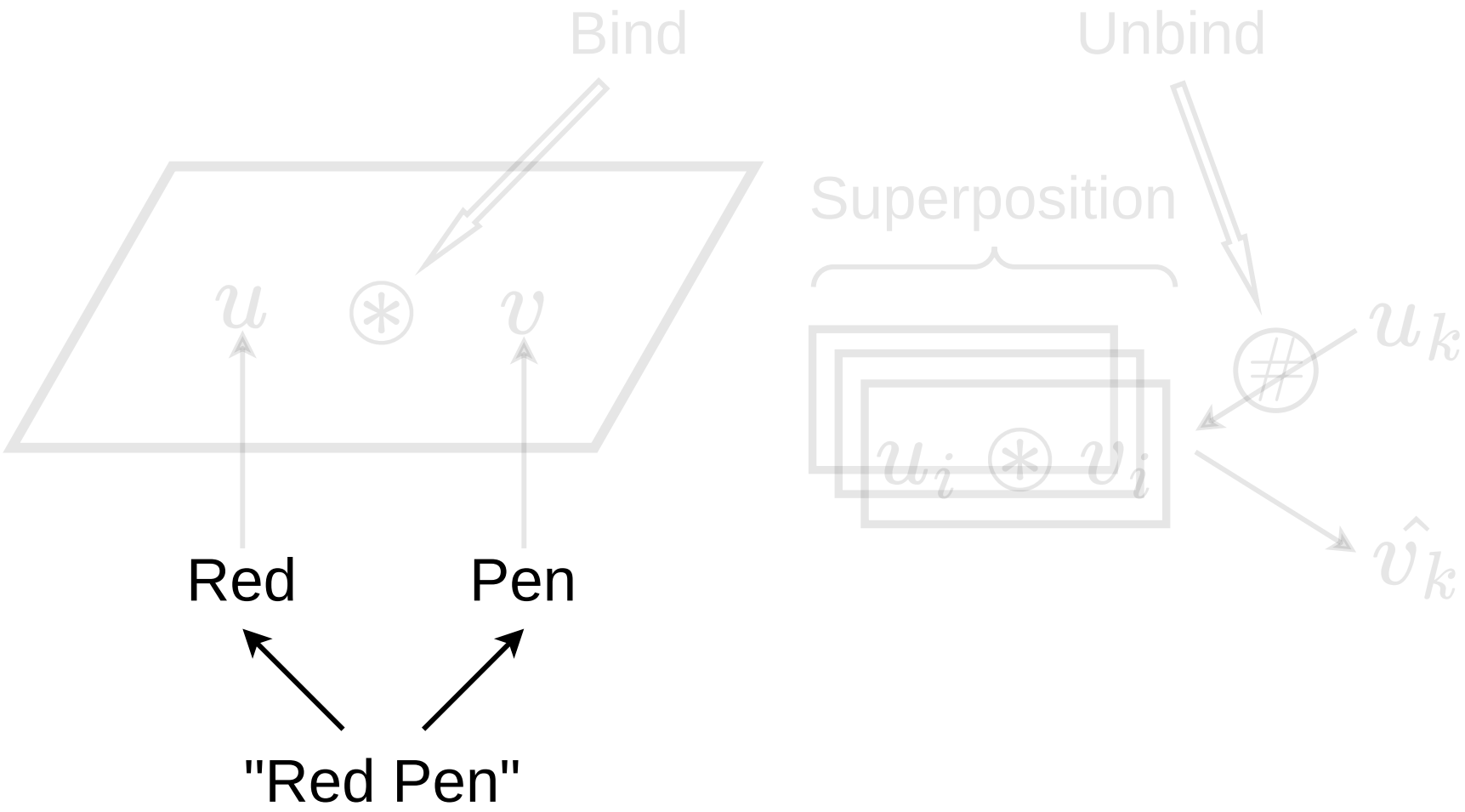
$\Downarrow$

Vector-**S**ymbolic **A**rchitecture (VSA) formed Memory

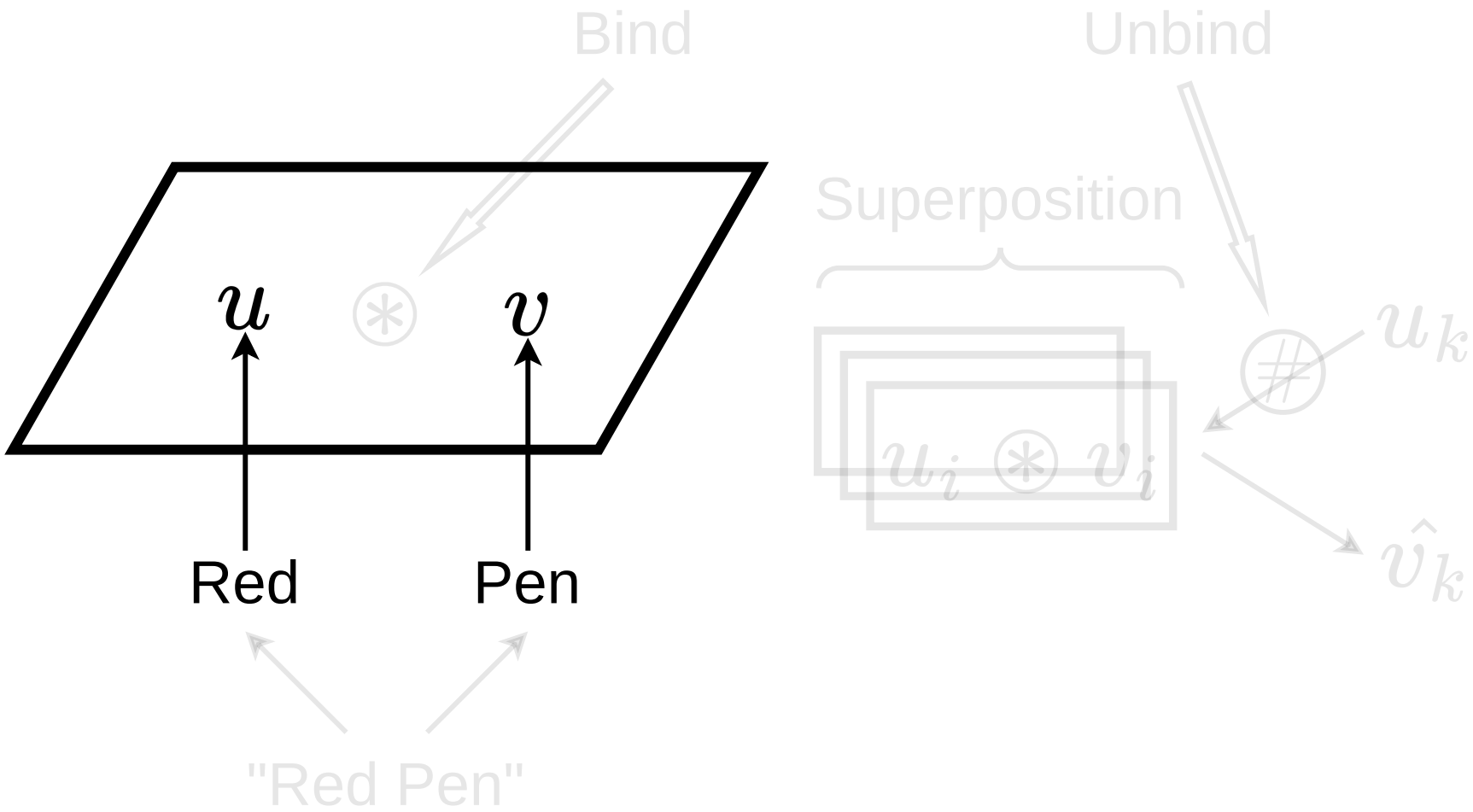
$\Downarrow$

NeSy'25 [6] (This Talk)

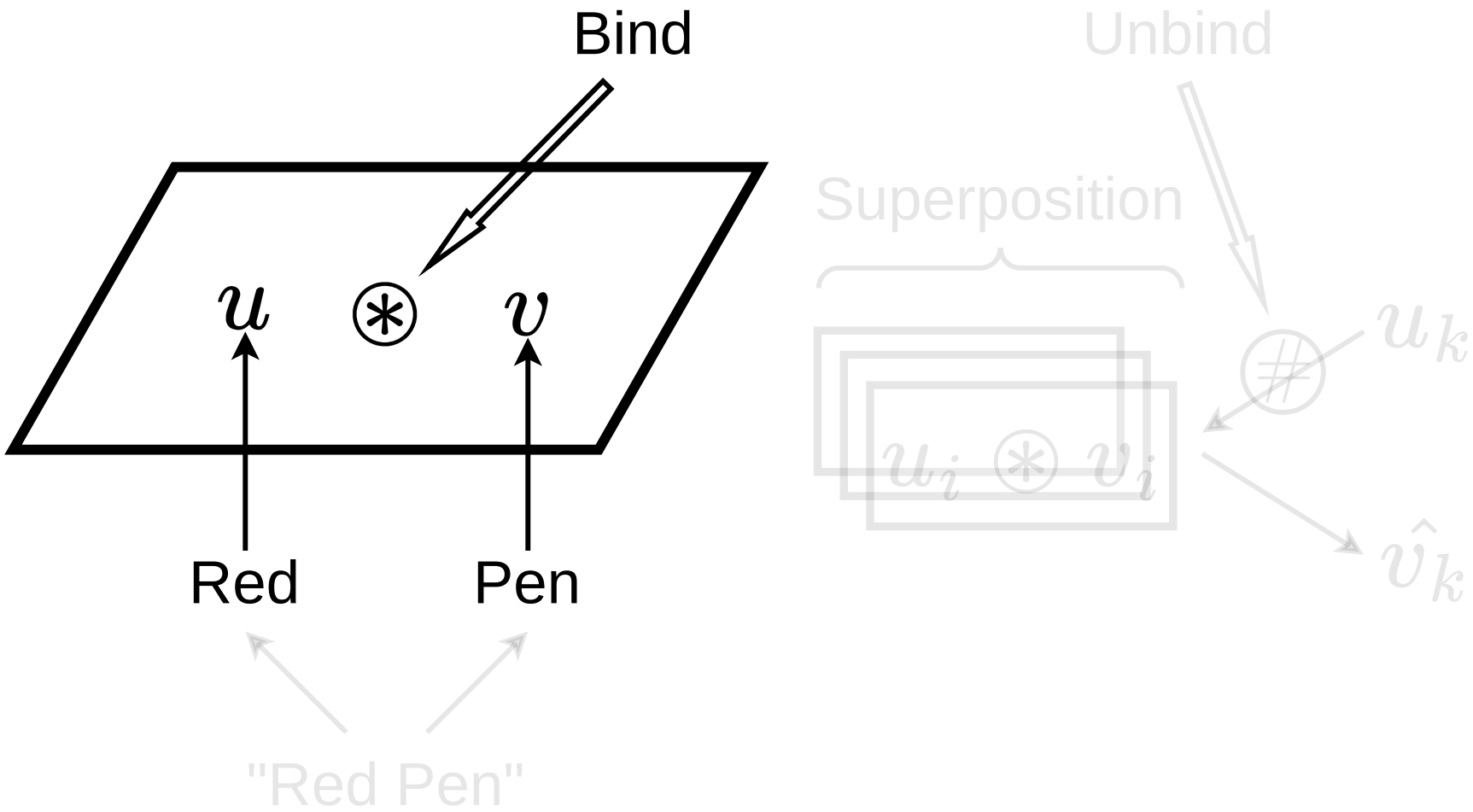
Vector Symbolic Architecture (VSA)



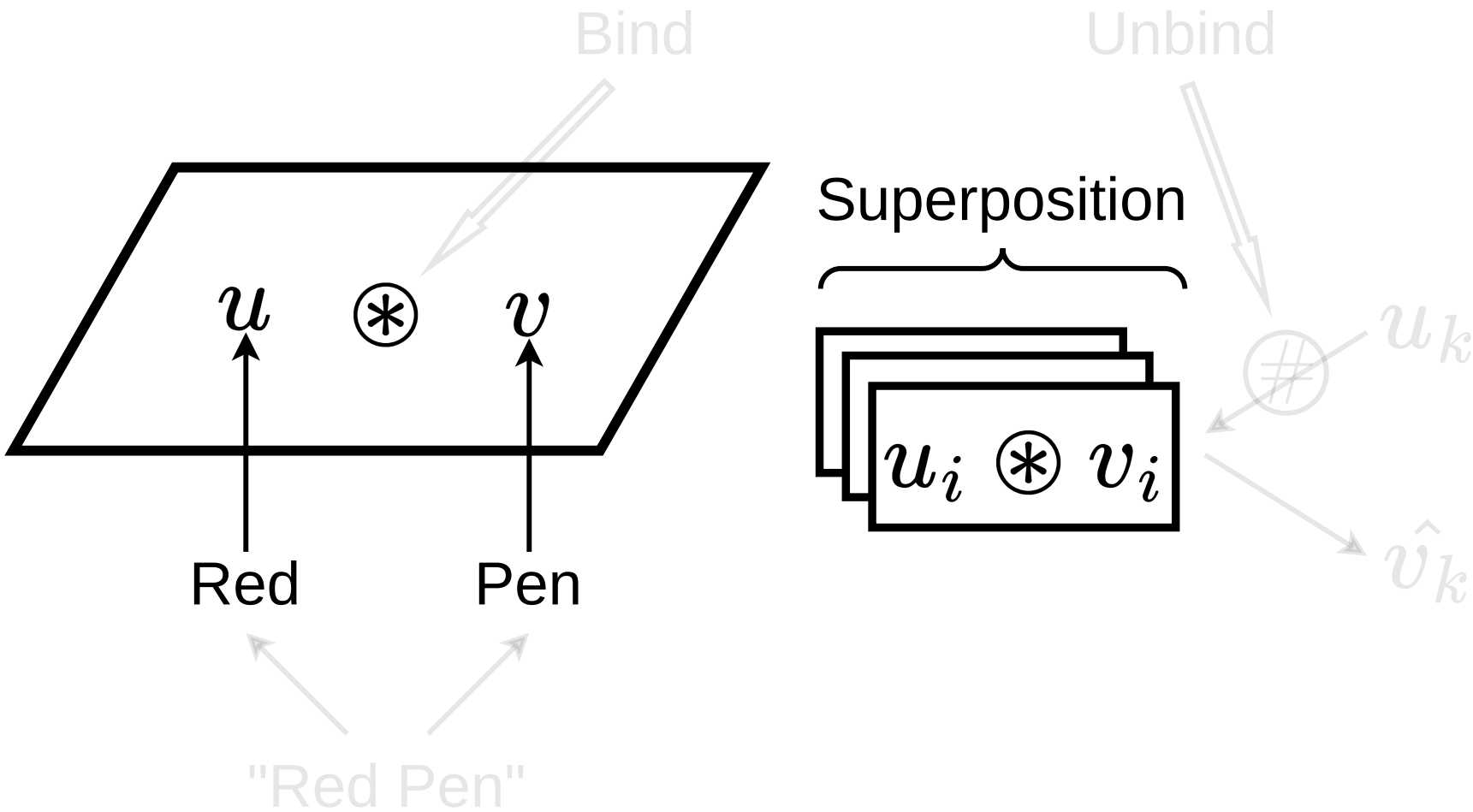
Vector Symbolic Architecture (VSA)



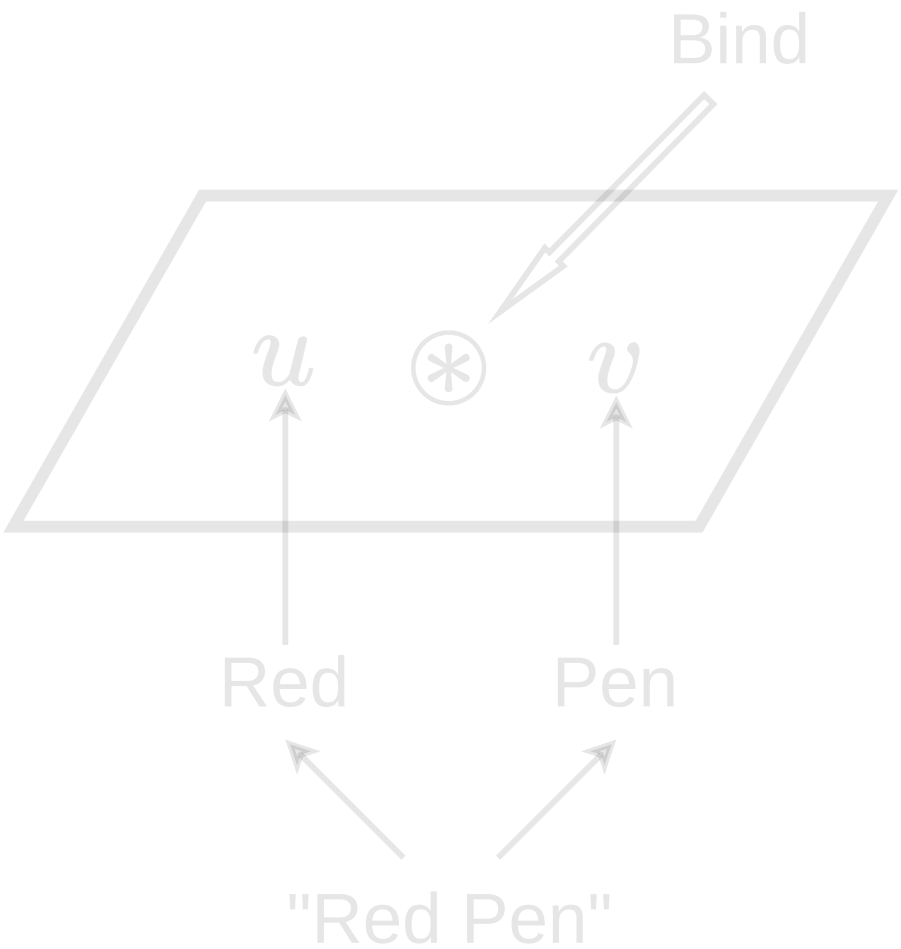
Vector Symbolic Architecture (VSA)

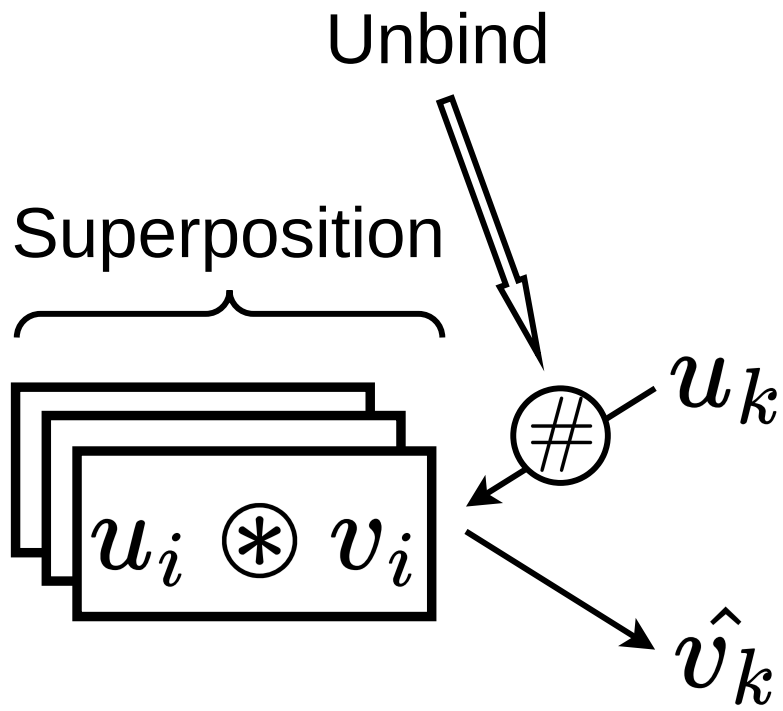


Vector Symbolic Architecture (VSA)

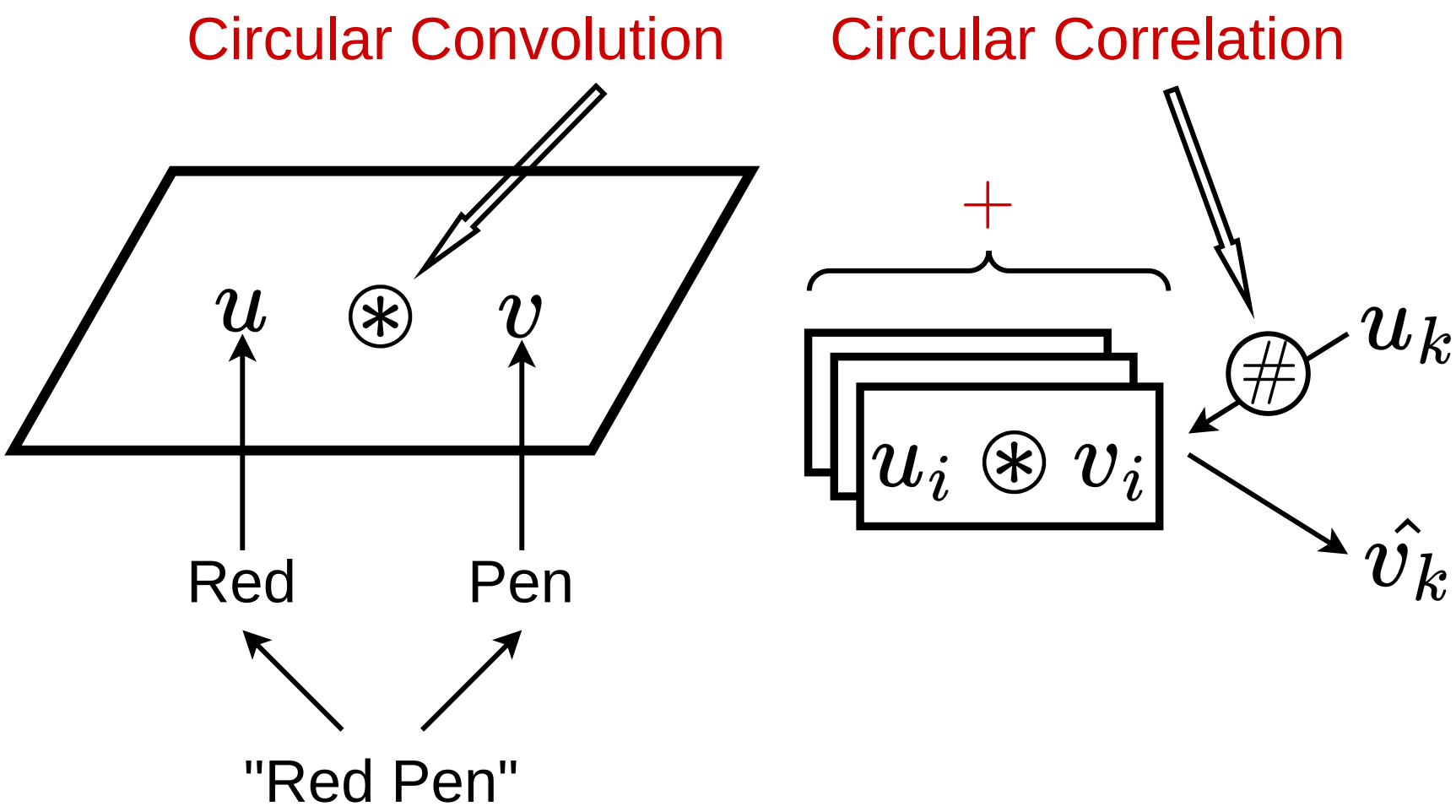


Vector Symbolic Architecture (VSA)

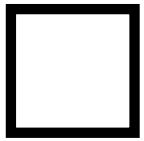


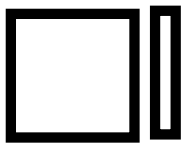


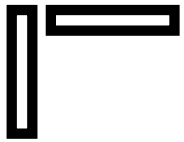
Holographic Reduced Representations (HRRs) [7]



Resulting In

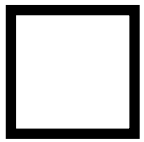
Memory: M  $\Longrightarrow$ $\mathcal{M}$

Read: $\sigma(Ma)$  $\Longrightarrow$ $\mathcal{M} \circledast a$

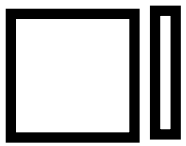
Write: $va^\top / N$  $\Longrightarrow$ $a \circledast v$

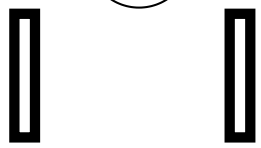
Resulting In

Memory: $M \Longrightarrow \mathcal{M}$

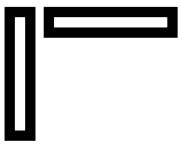


Read: $\sigma(Ma) \Longrightarrow \mathcal{M} \# a$





Write: $va^\top / N \Longrightarrow a \circledast v$



Perfect?

Issue During Reading

Memory $\mathcal{M} = \sum_i a_i \circledast v_i$

Read $\mathcal{M} \circledast a_k = v'_k$ ⟸ Contains Noise

Codebook

Clean-up $v_k^{cleaned} = \arg\max H v'_k$

Issue During Reading

Memory $\mathcal{M} = \sum_i a_i \circledast v_i$

Read $\mathcal{M} \circledast a_k = v'_k$ ⇐ Contains Noise

Codebook

Clean-up $v_k^{cleaned} = \arg\max H v'_k$

Issue During Reading

Memory $\mathcal{M} = \sum_i a_i \circledast v_i$

Read $\mathcal{M} \circledast a_k = v'_k$ ⇐ Contains Noise

Codebook

Clean-up $v_k^{cleaned} = \arg\max H v'_k$

Issue During Reading

Memory $\mathcal{M} = \sum_i a_i \circledast v_i$

Read $\mathcal{M} \circledast a_k = v'_k$ ⇐ Contains Noise

Codebook ↙

Clean-up $v_k^{cleaned} = \arg\max H v'_k$

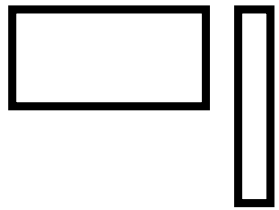
Issue During Reading

Memory $\mathcal{M} = \sum_i a_i \circledast v_i$

Read $\mathcal{M} \circledast a_k = v'_k$ ⇐ Contains Noise

Codebook ↙

Clean-up $v_k^{cleaned} = \arg\max H v'_k$



Kroneker **Ro**tation **P**roduct Codebook (`krop`)

Codebook based on **Hadamard Matrices** using Sylvester's construction and **Kroneker Rotation Products**:

$$\tilde{H}^{(0)} = [1], \quad \tilde{H}^{(k+1)} = \begin{bmatrix} \tilde{H}^{(k)}\cos(\theta_k) & \tilde{H}^{(k)}\sin(\theta_k) \\ \tilde{H}^{(k)}\sin(\theta_k) & -\tilde{H}^{(k)}\cos(\theta_k) \end{bmatrix}$$

$$\Rightarrow \quad \tilde{H}^{(K)} = \bigotimes_{k=1}^{K} \begin{bmatrix} \cos(\theta_{K-k}) & \sin(\theta_{K-k}) \\ \sin(\theta_{K-k}) & -\cos(\theta_{K-k}) \end{bmatrix}$$

where θ_k is sampled uniformly from $(0, 2\pi)$.

$\mathcal{O}(\log N)$ Storage Space

$\log N$ Storage Space

$$\overbrace{\hphantom{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}}^{N}$$

$$\tilde{H}^{(k)} = \begin{bmatrix} \tilde{H}^{(k-1)}\cos(\theta_{k-1}) & \tilde{H}^{(k-1)}\sin(\theta_{k-1}) \\ \tilde{H}^{(k-1)}\sin(\theta_{k-1}) & -\tilde{H}^{(k-1)}\cos(\theta_{k-1}) \end{bmatrix}$$

$$\underbrace{\theta_0, \theta_1, \cdots, \theta_{k-1}}_{\mathcal{O}(\log N)}$$

log N Storage Space

$$\tilde{H}^{(k)} = \overbrace{\left[\begin{array}{cc} \tilde{H}^{(k-1)}\cos(\theta_{k-1}) & \tilde{H}^{(k-1)}\sin(\theta_{k-1}) \\ \tilde{H}^{(k-1)}\sin(\theta_{k-1}) & -\tilde{H}^{(k-1)}\cos(\theta_{k-1}) \end{array}\right]}^{N}$$

$$\underbrace{\theta_0, \theta_1, \cdots, \theta_{k-1}}_{\mathcal{O}(\log N)}$$

log N Storage Space

$$\tilde{H}^{(k)} = \overbrace{\begin{bmatrix} \tilde{H}^{(k-1)}\cos(\theta_{k-1}) & \tilde{H}^{(k-1)}\sin(\theta_{k-1}) \\ \tilde{H}^{(k-1)}\sin(\theta_{k-1}) & -\tilde{H}^{(k-1)}\cos(\theta_{k-1}) \end{bmatrix}}^{N}$$

$$\underbrace{\theta_0, \theta_1, \cdots, \theta_{k-1}}_{\mathcal{O}(\log N)}$$

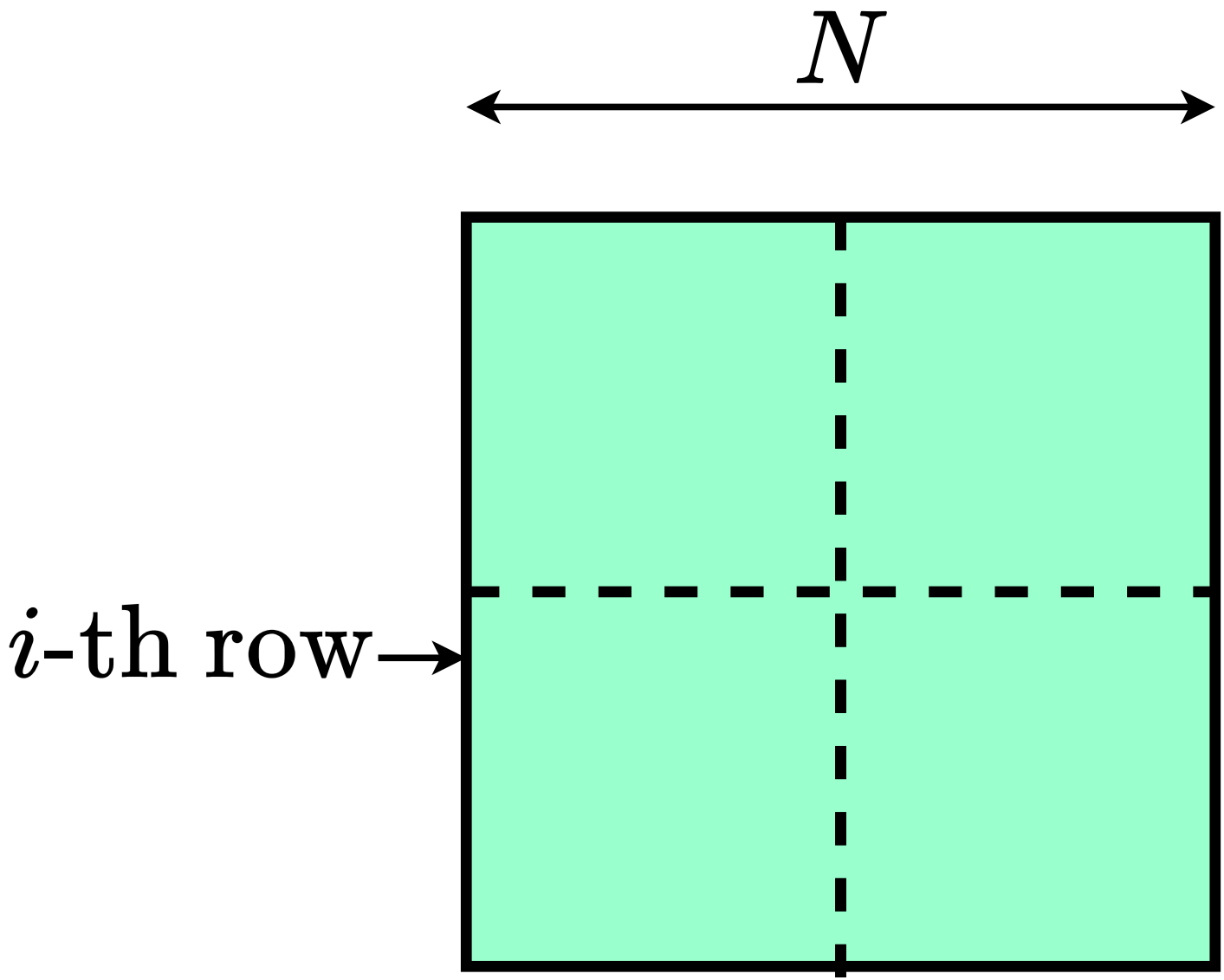
log N Storage Space

$$\tilde{H}^{(k)} = \overbrace{\begin{bmatrix} \tilde{H}^{(k-1)}\cos(\theta_{k-1}) & \tilde{H}^{(k-1)}\sin(\theta_{k-1}) \\ \tilde{H}^{(k-1)}\sin(\theta_{k-1}) & -\tilde{H}^{(k-1)}\cos(\theta_{k-1}) \end{bmatrix}}^{N}$$

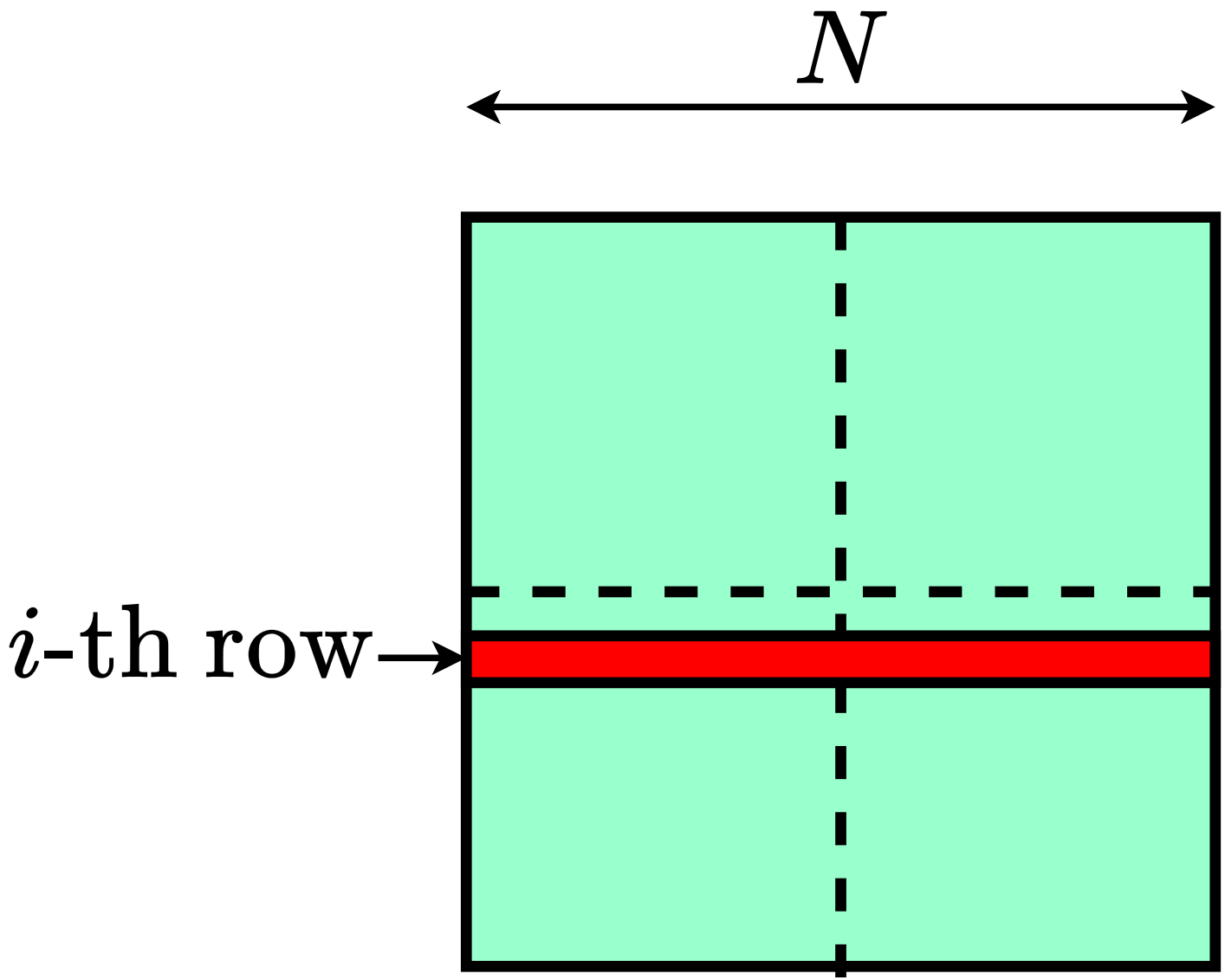
$$\underbrace{\theta_0, \theta_1, \cdots, \theta_{k-1}}_{\mathcal{O}(\log N)}$$

Linear Time Vector Reconstruction

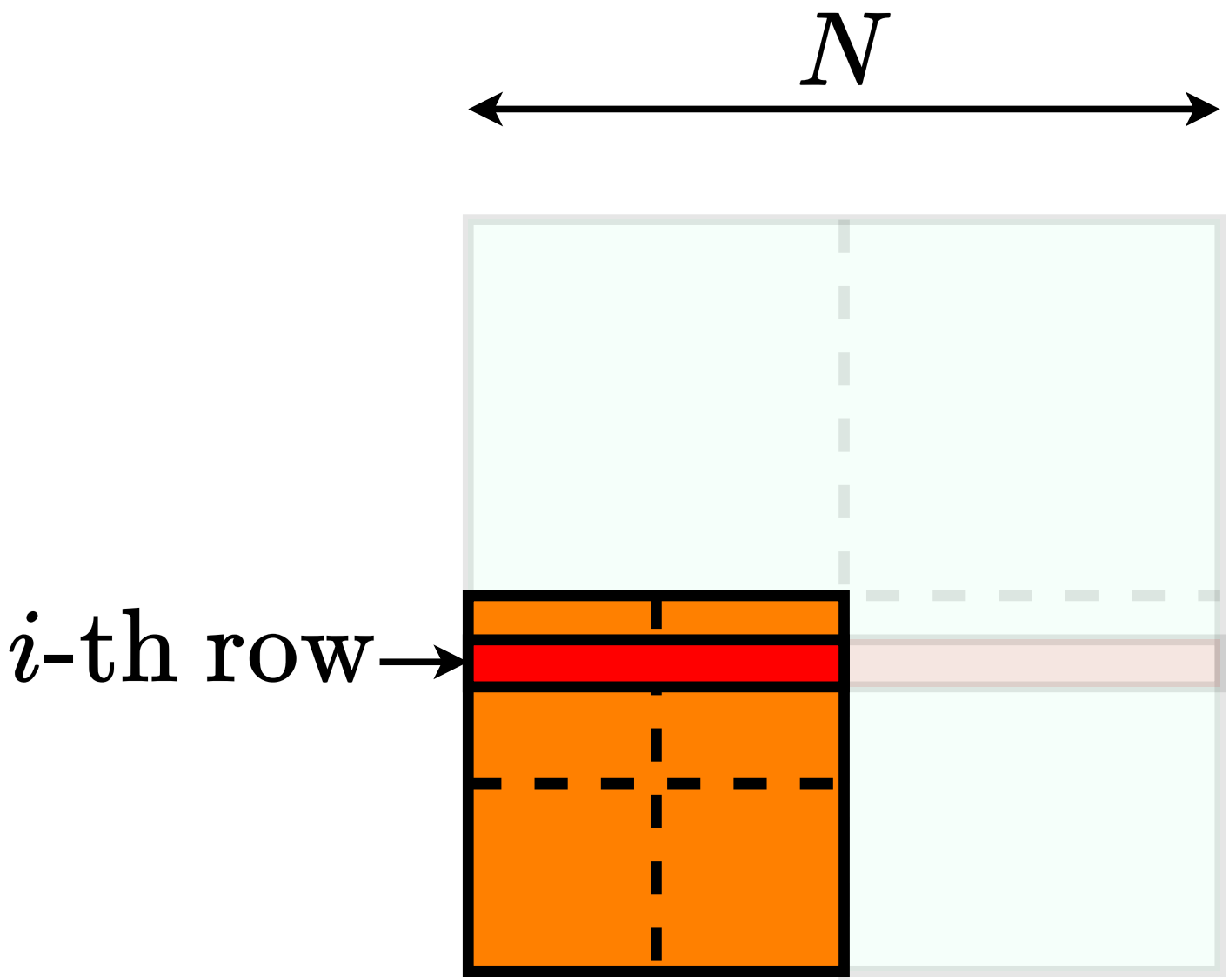
Linear Time Vector Reconstruction



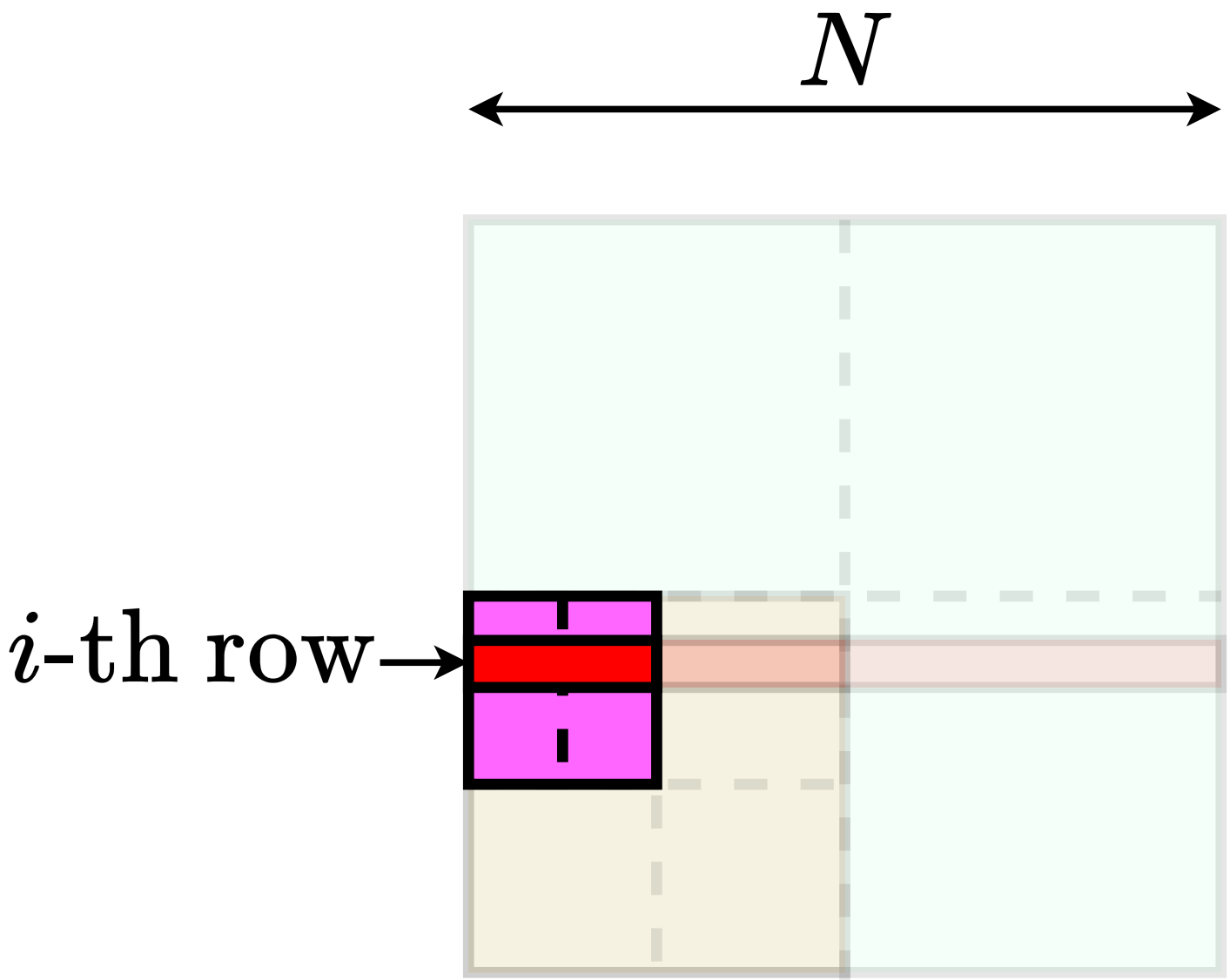
Linear Time Vector Reconstruction



Linear Time Vector Reconstruction

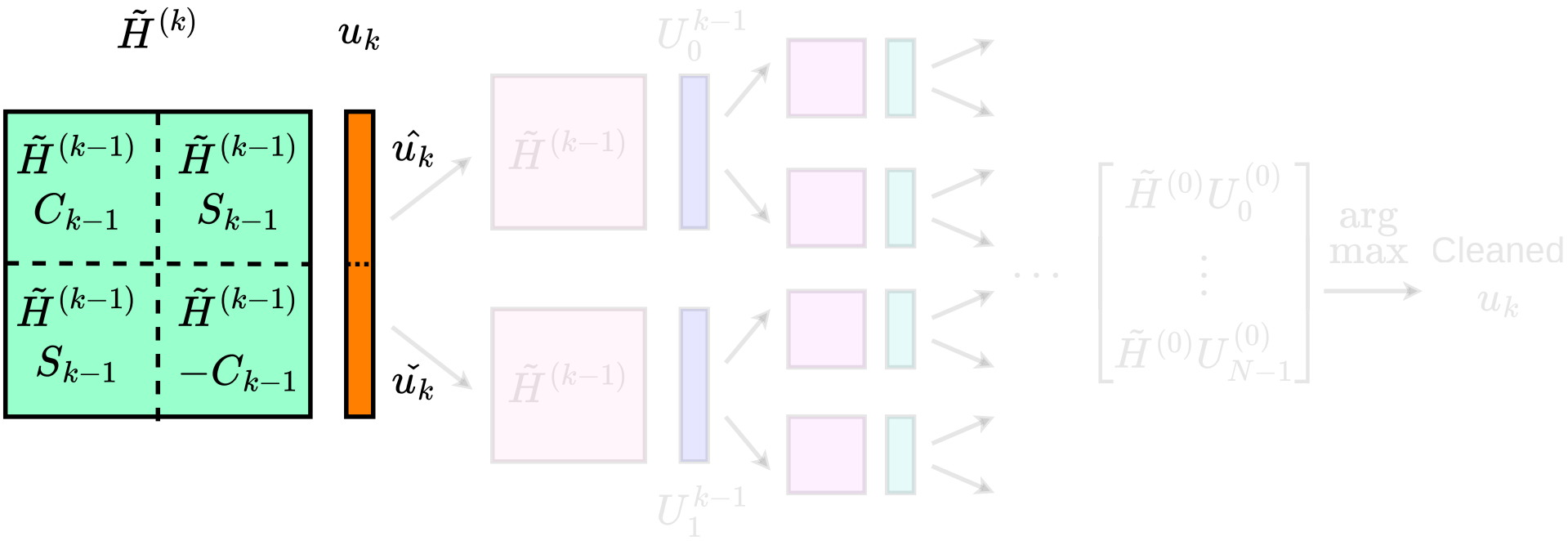


Linear Time Vector Reconstruction

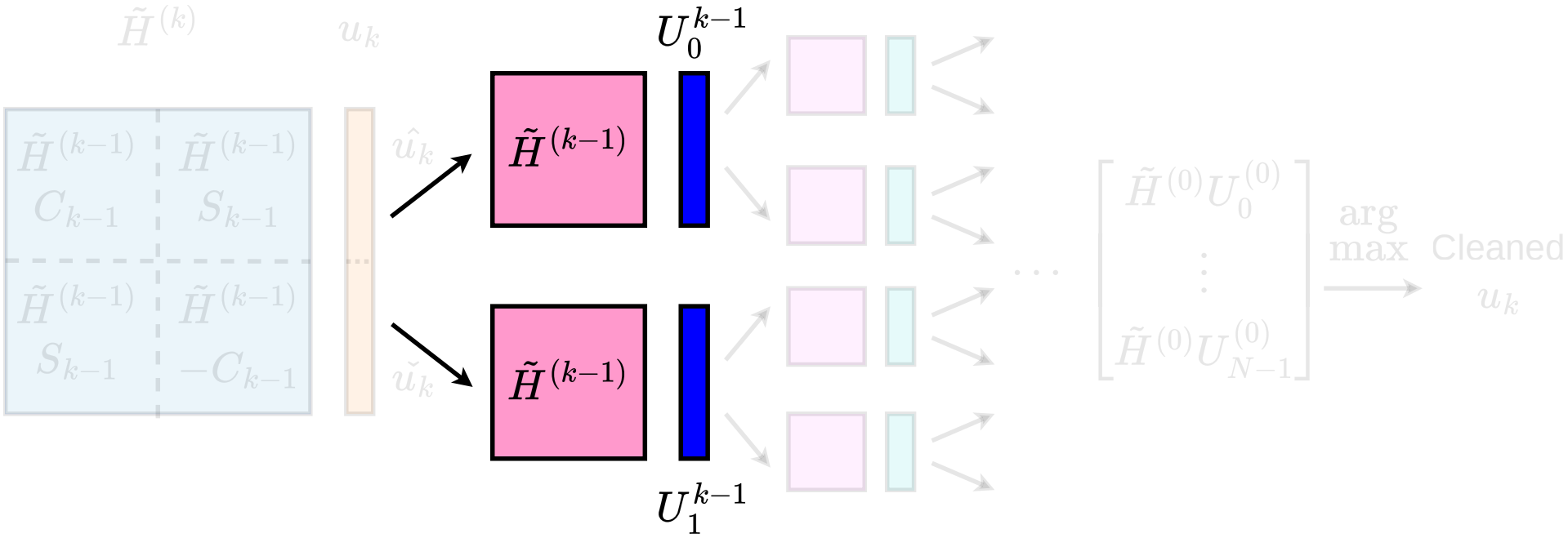


$\mathcal{O}(N \log N)$ Clean-up Process

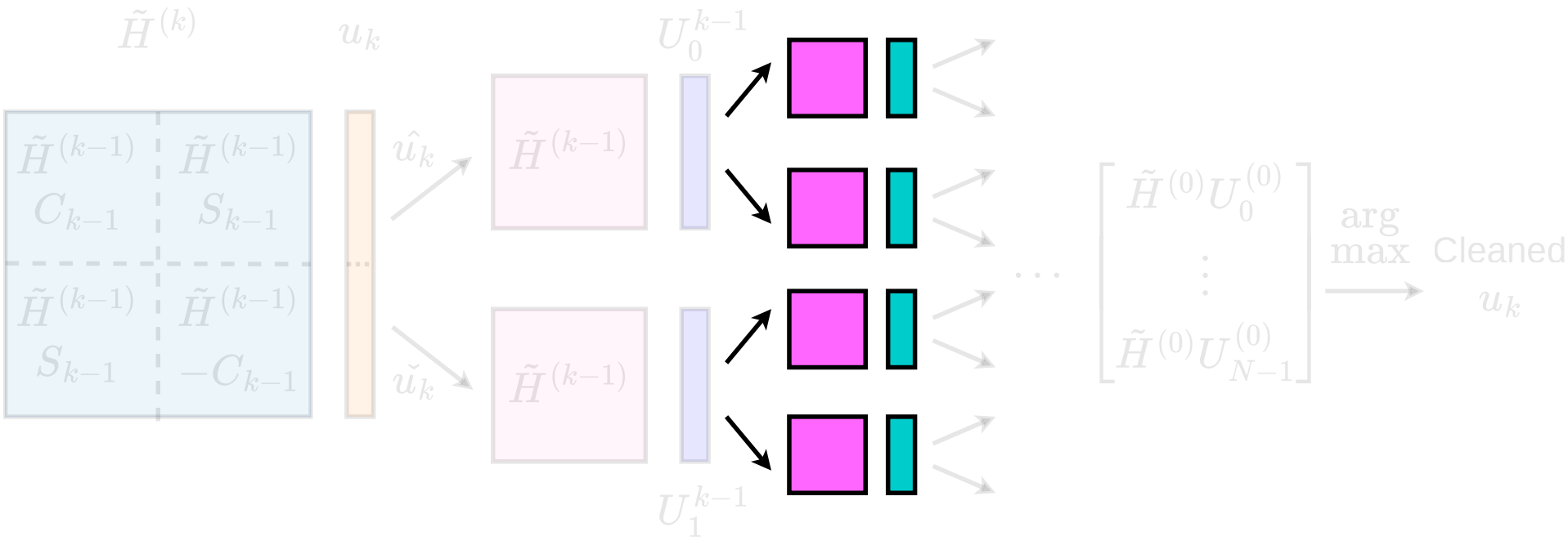
Speeding Up the Clean-up Process



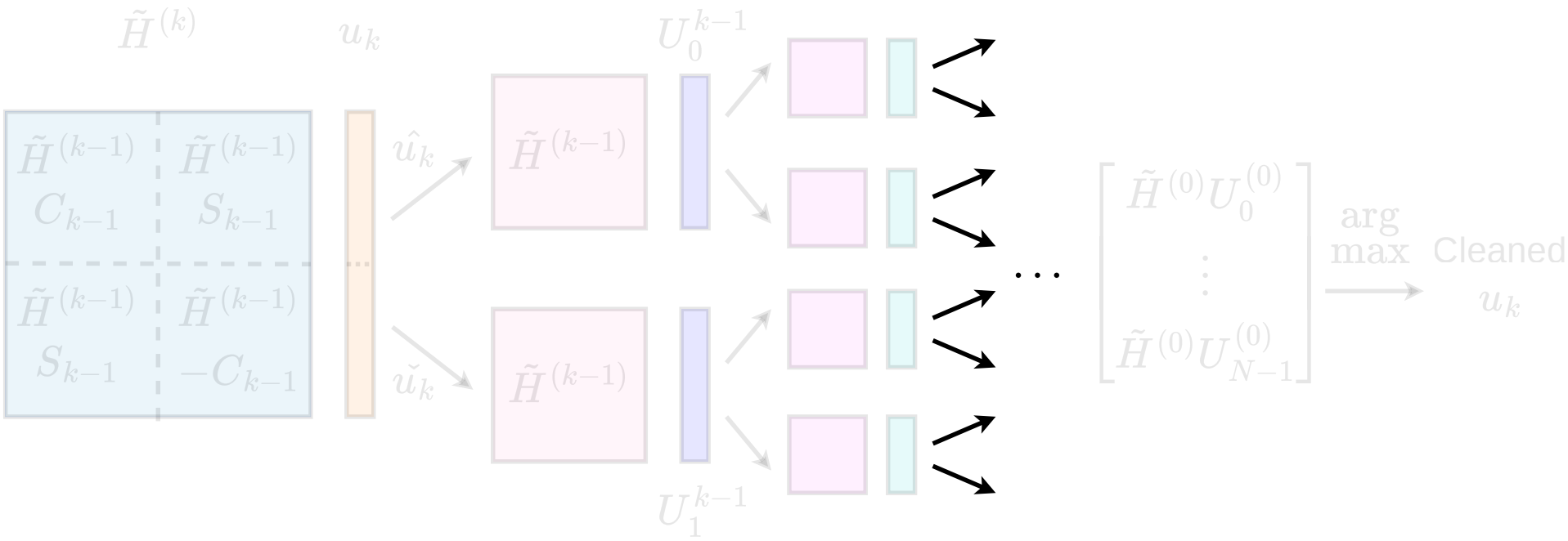
Speeding Up the Clean-up Process



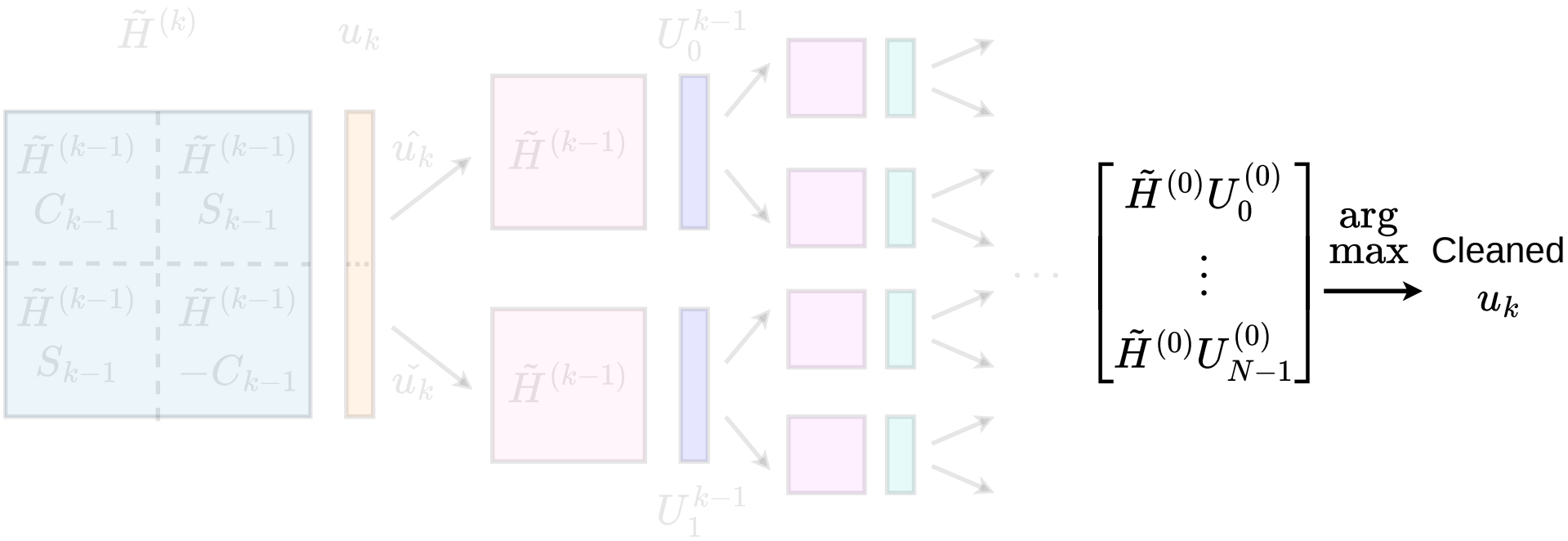
Speeding Up the Clean-up Process



Speeding Up the Clean-up Process



Speeding Up the Clean-up Process



Experiments

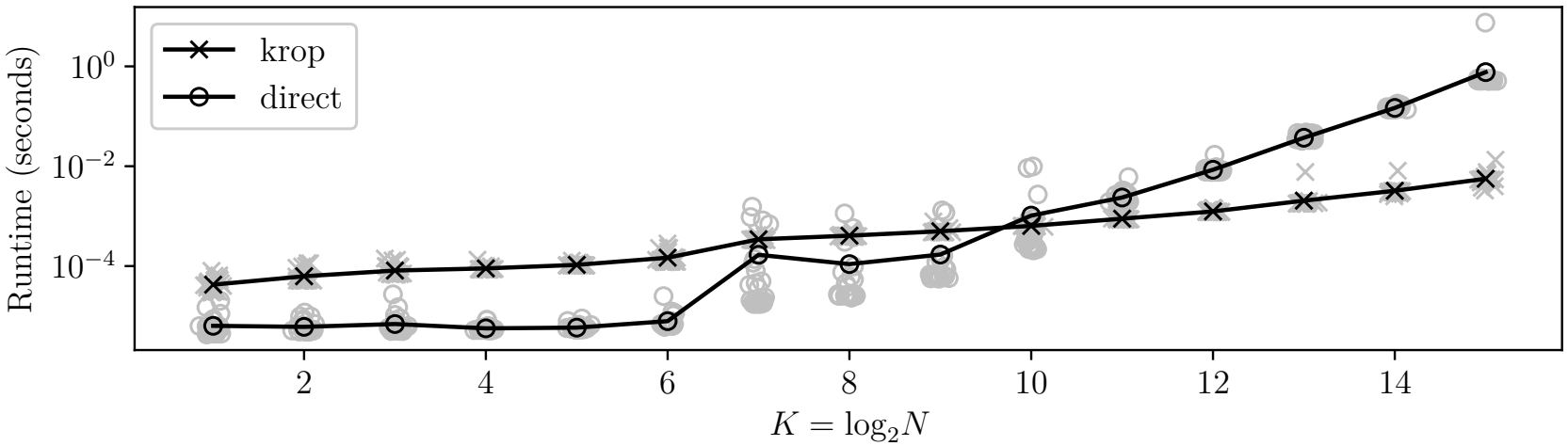
Clean-up Efficiency

Comparing clean-up runtime

How much faster?

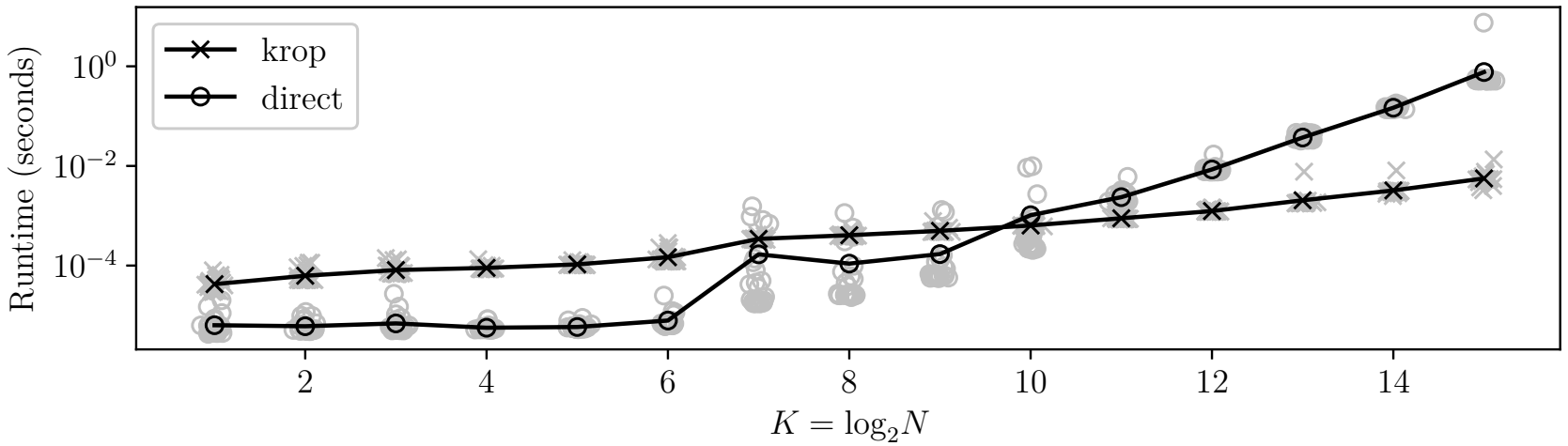
Comparing clean-up runtime

How much faster?



Comparing clean-up runtime

How much faster?

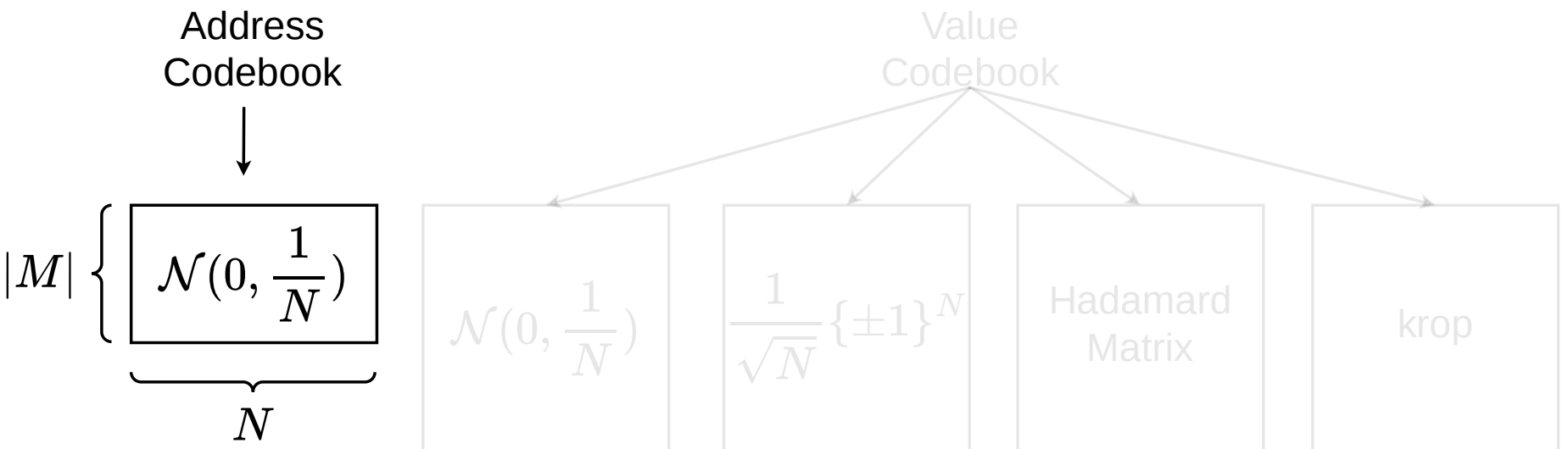


`krop` scales better.

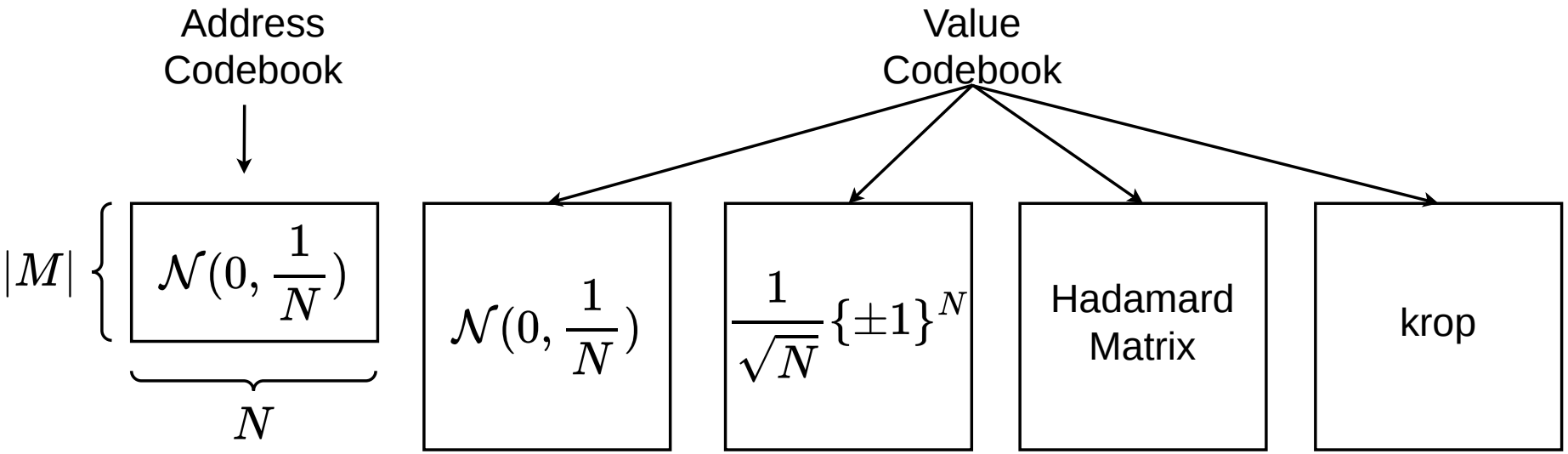
Memory Capacity Test

Is a stored item recovered correctly?

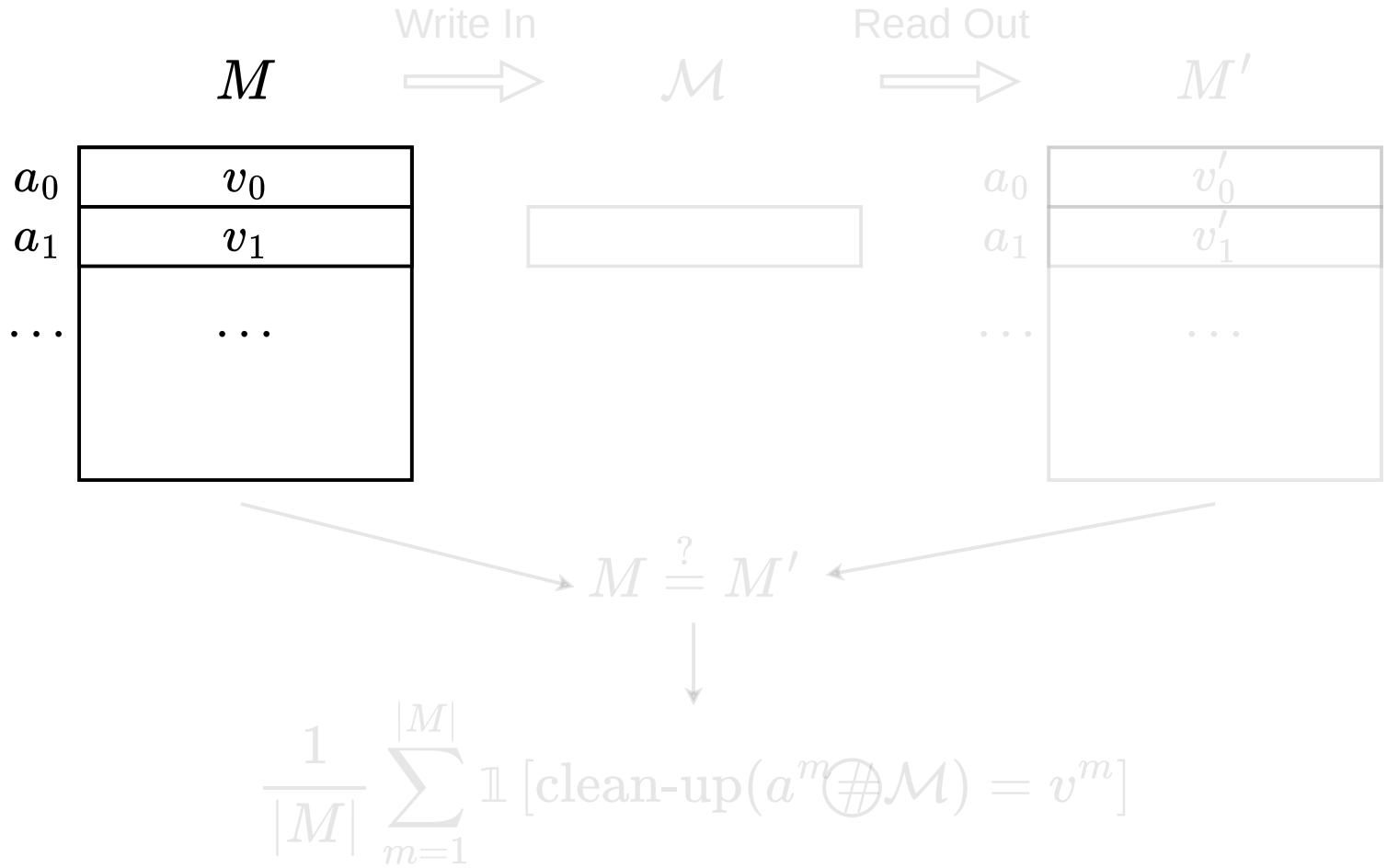
Codebooks


Address
Codebook
$|M|$
$\mathcal{N}(0, \frac{1}{N})$
N
Value
Codebook
$\mathcal{N}(0, \frac{1}{N})$
$\frac{1}{\sqrt{N}}\{\pm 1\}^N$
Hadamard
Matrix
krop

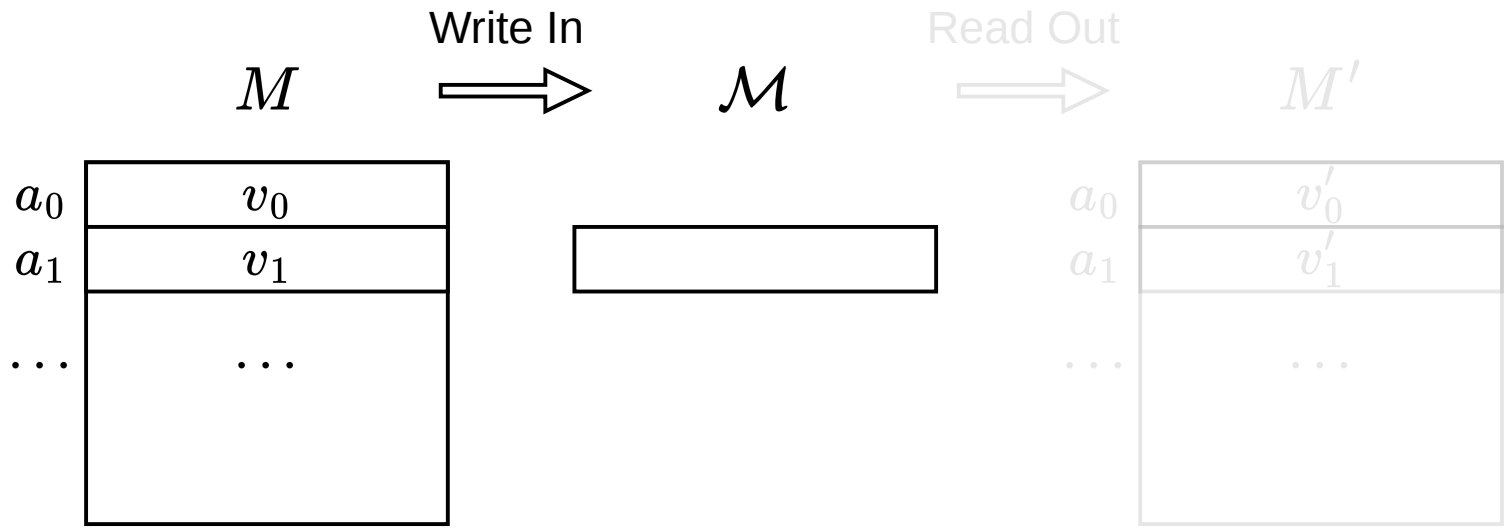
Codebooks


Address
Codebook
Value
Codebook
$|M|$
$\mathcal{N}(0, \frac{1}{N})$
N
$\mathcal{N}(0, \frac{1}{N})$
$\frac{1}{\sqrt{N}}\{\pm 1\}^N$
Hadamard
Matrix
krop

Memory Capacity Test



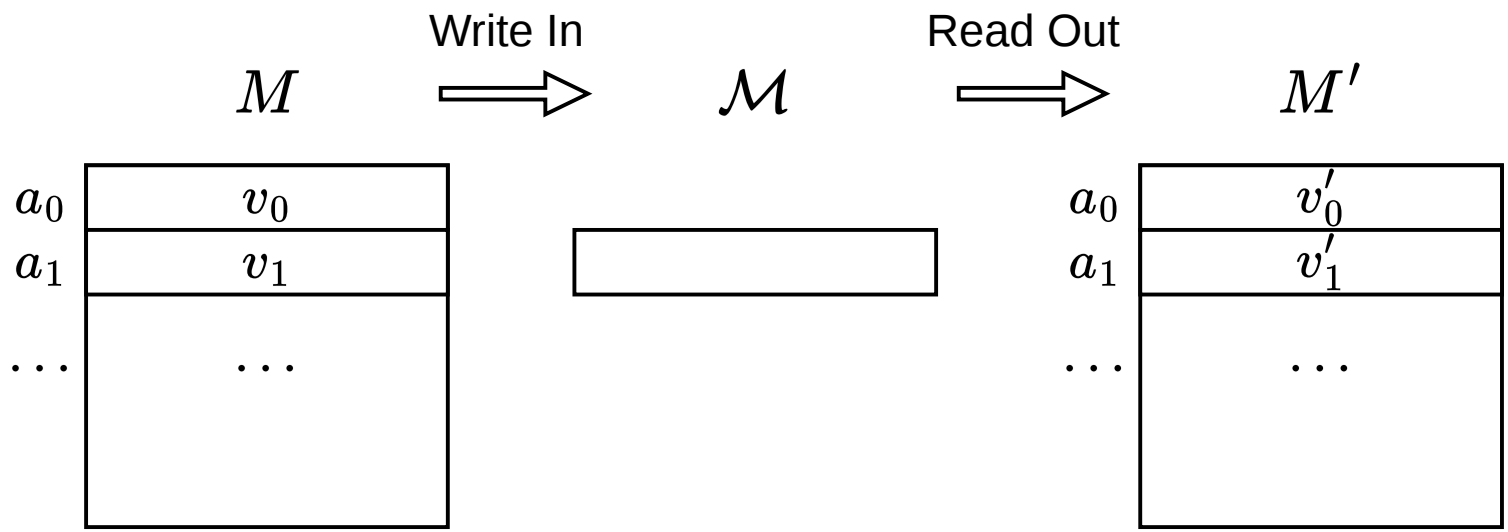
Memory Capacity Test



$$M \stackrel{?}{=} M'$$

$$\frac{1}{|M|}\sum_{m=1}^{|M|}\mathbb{1}\left[\text{clean-up}(a^m \circledast \mathcal{M}) = v^m\right]$$

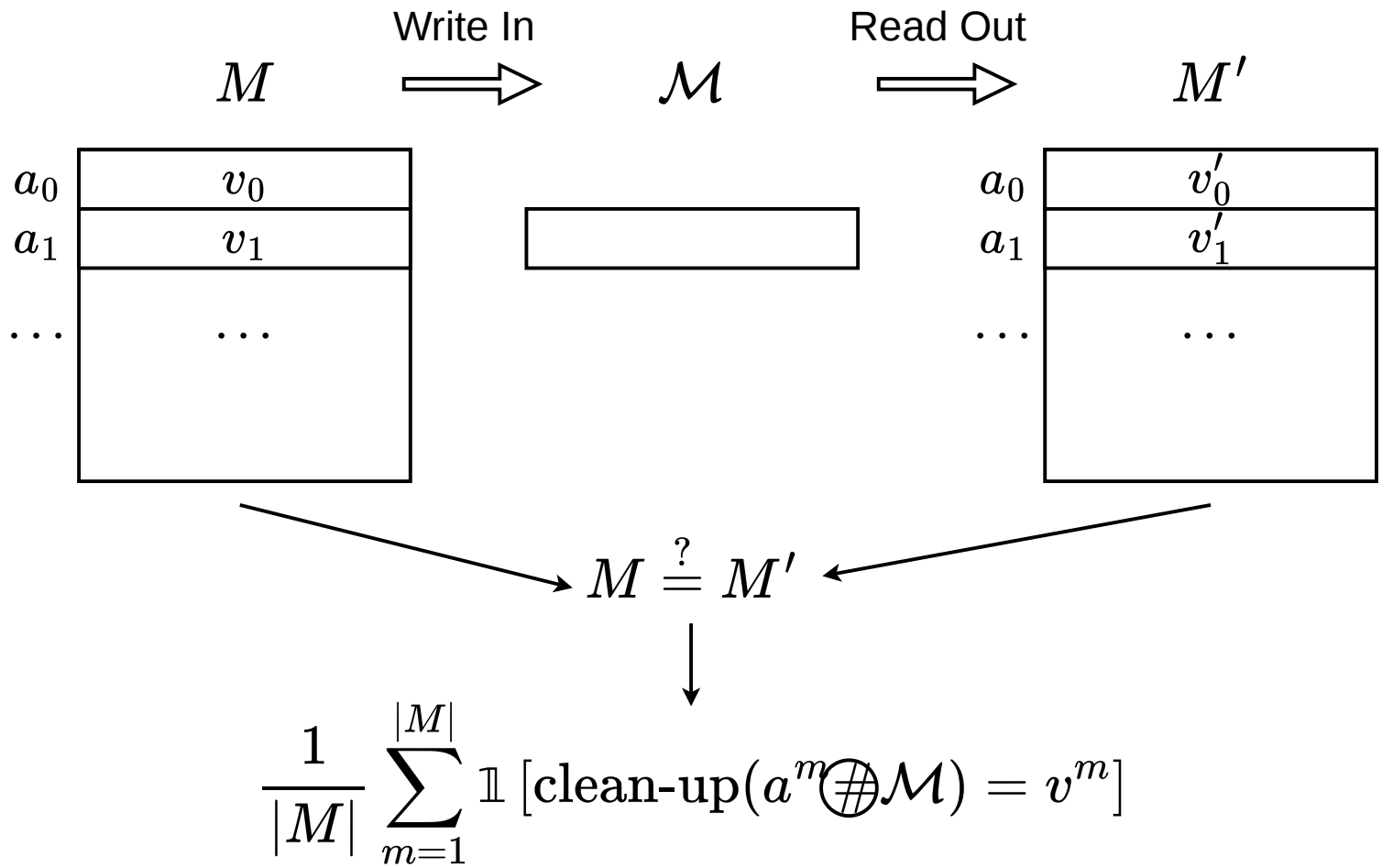
Memory Capacity Test



$$M \stackrel{?}{=} M'$$

$$\frac{1}{|M|}\sum_{m=1}^{|M|}\mathbb{1}\left[\text{clean-up}(a^m \circledast \mathcal{M}) = v^m\right]$$

Memory Capacity Test



Memory Capacity Result

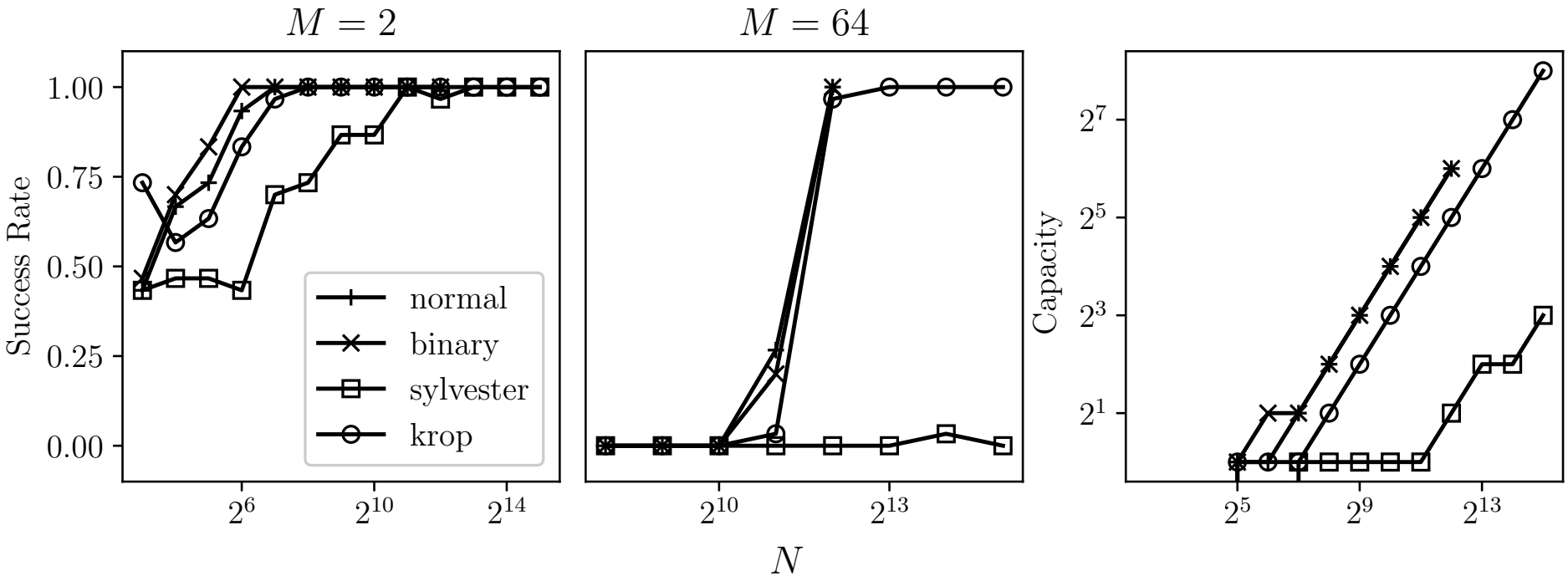
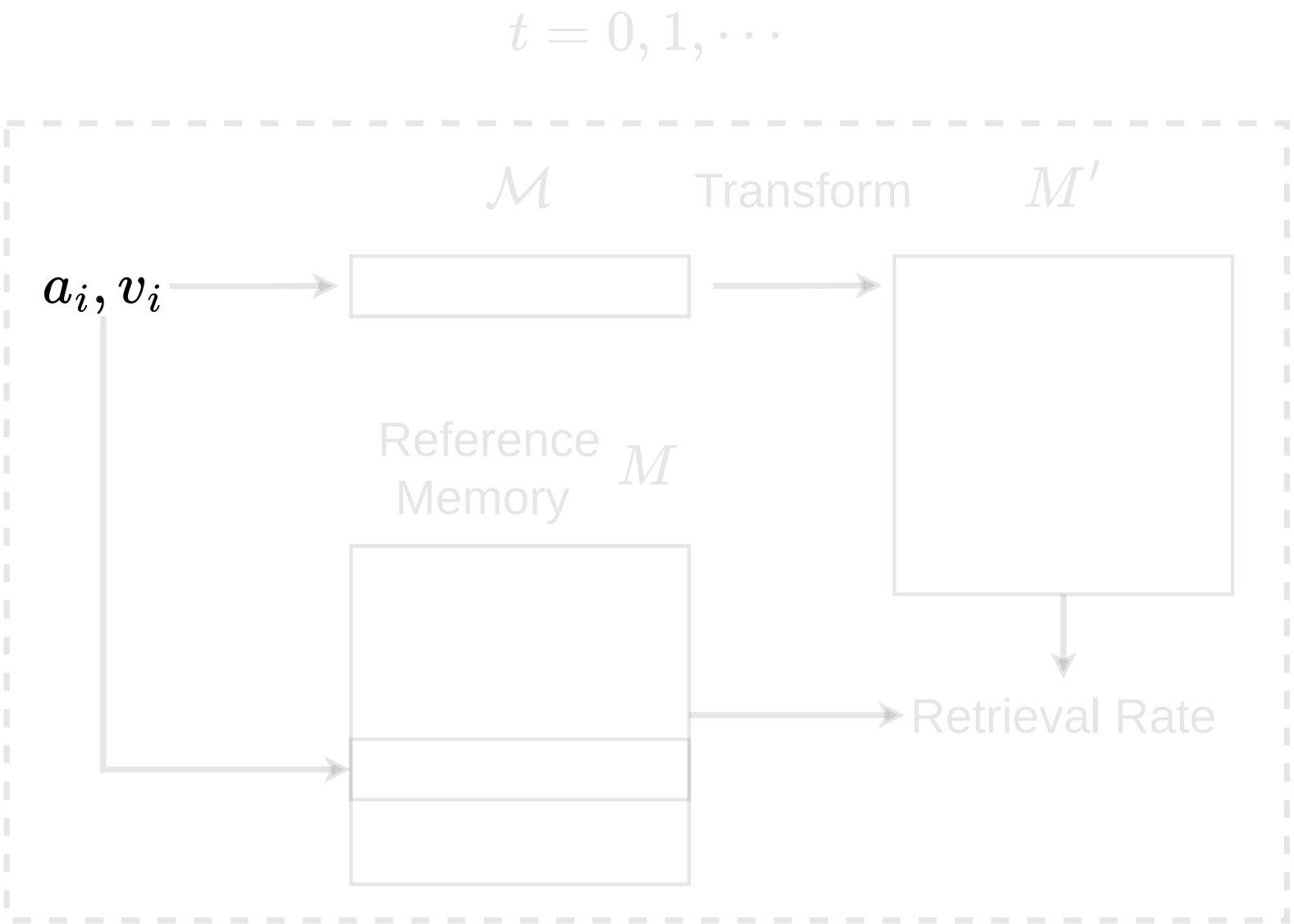


Figure: Success rates (left) and memory capacity (right) for each kind of value codebook.

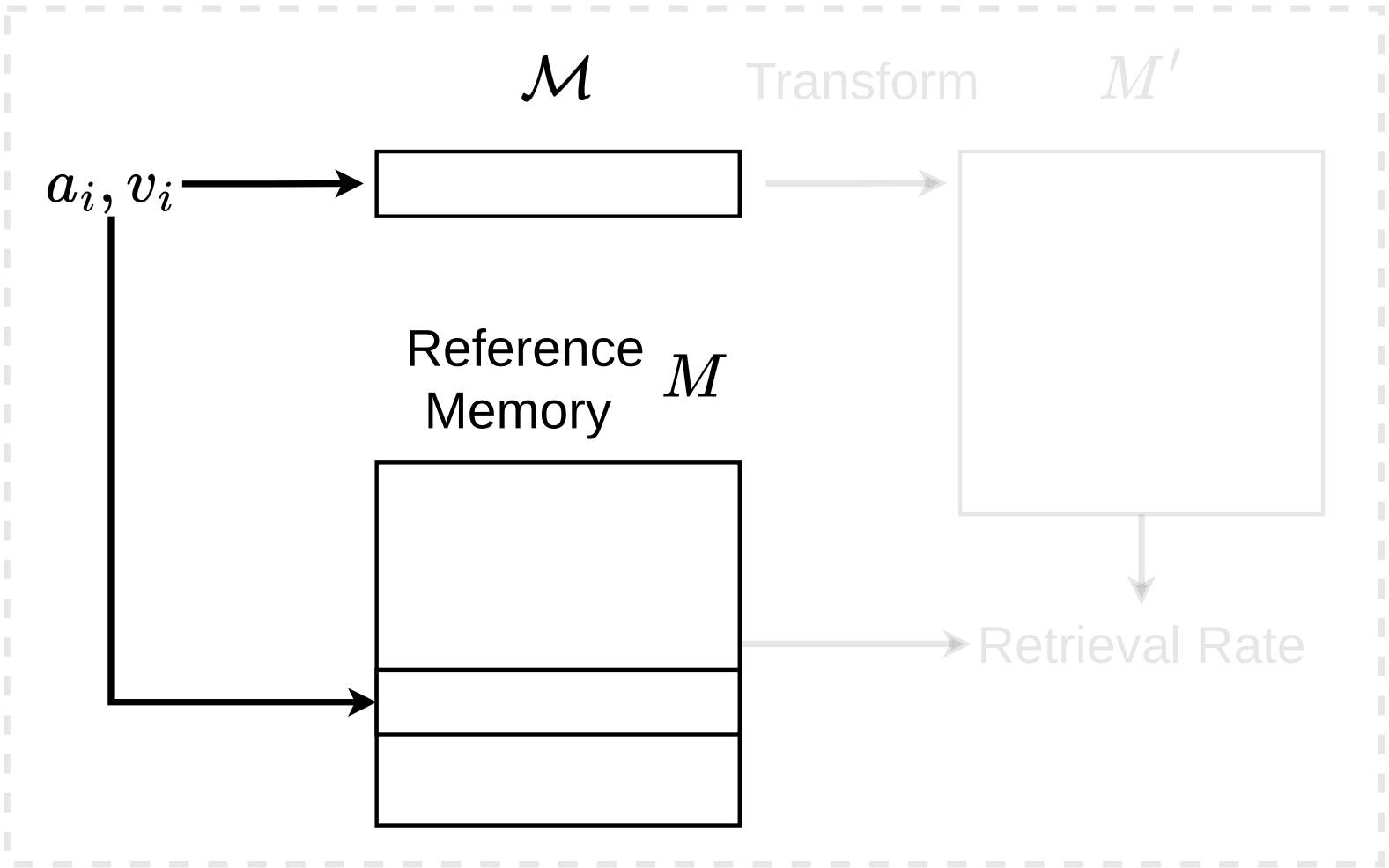
Mutable Key-Value Memory Test

Does read-out error accumulate **over time**?

Mutable Key-Value Memory

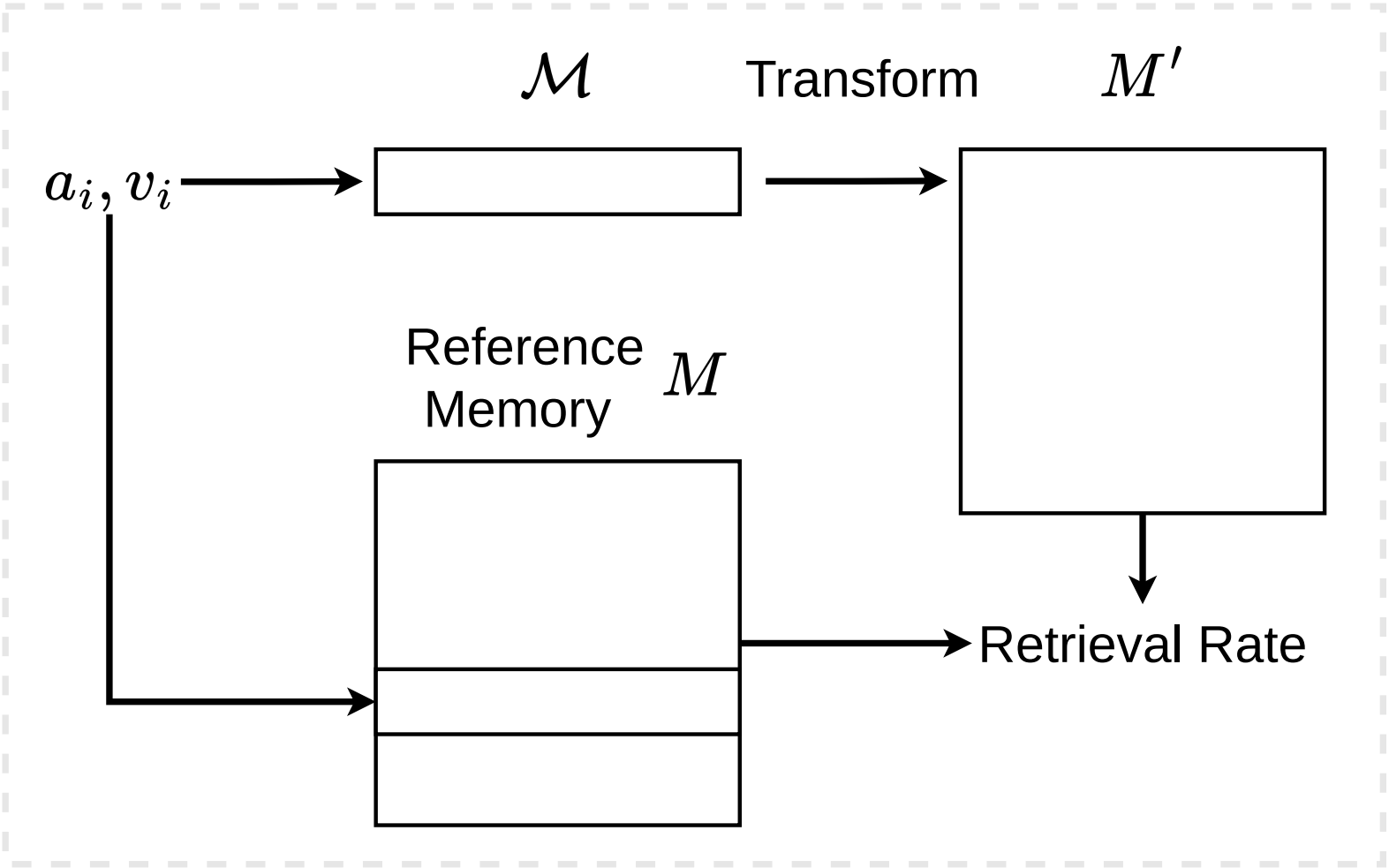


Mutable Key-Value Memory



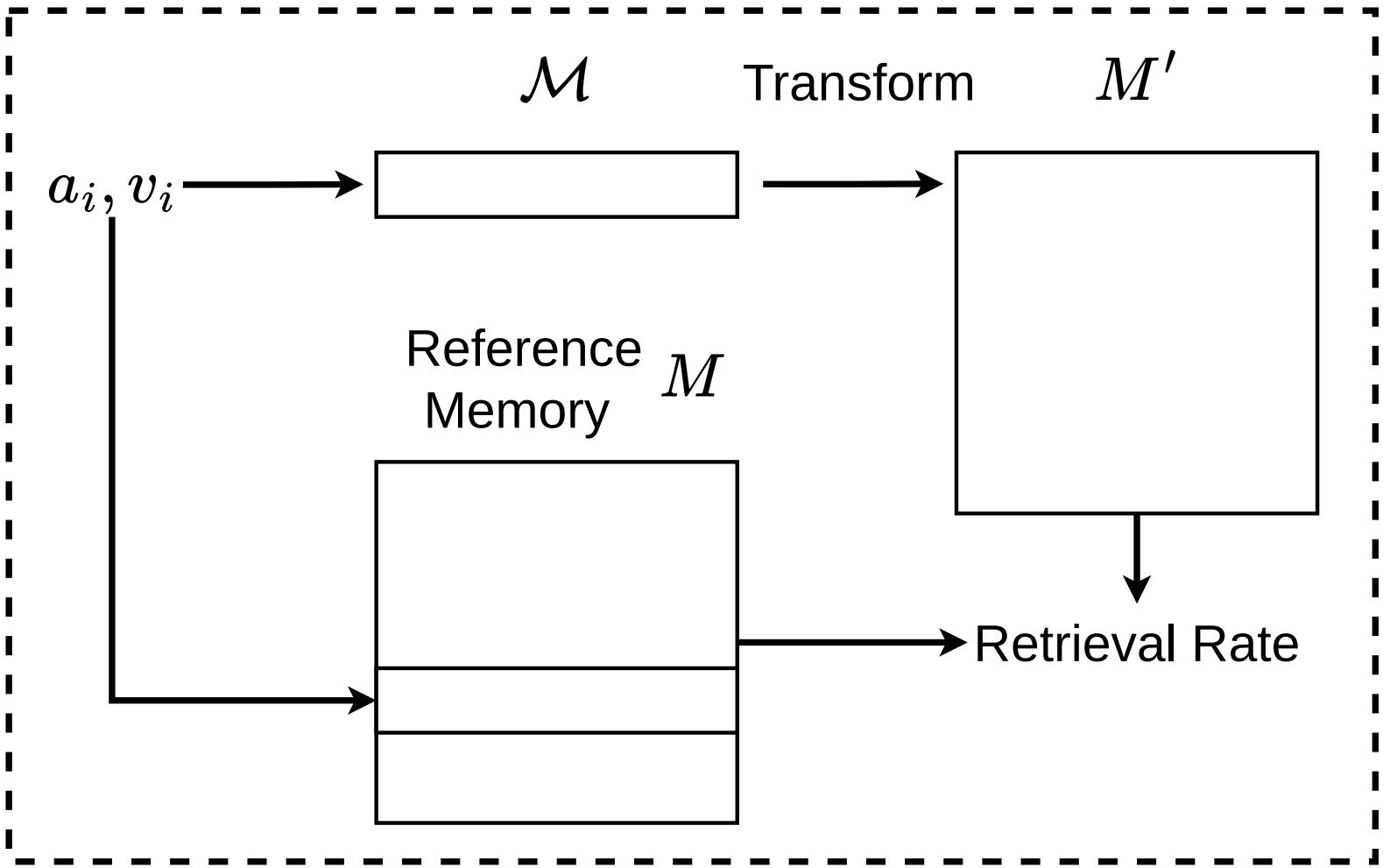
Mutable Key-Value Memory

$t = 0, 1, \cdots$



Mutable Key-Value Memory

$$t = 0, 1, \cdots$$



Mutable Key-Value Memory Result

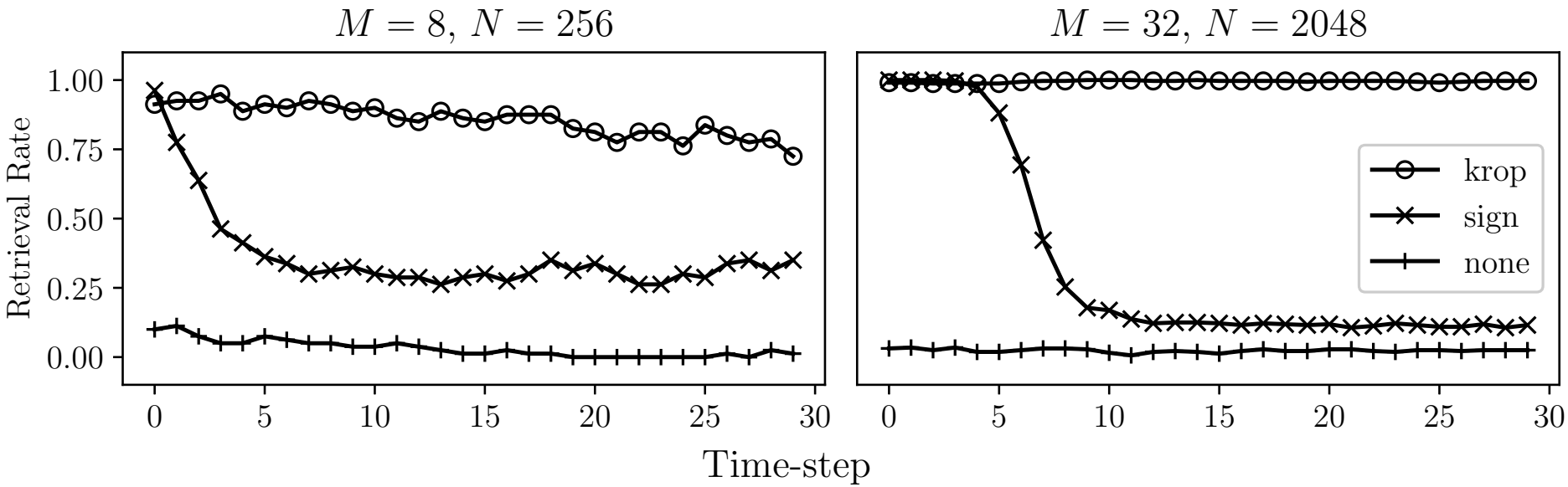


Figure: Examples of mutable VSA memory retrieval rates by time-step.

Mutable Key-Value Memory Result

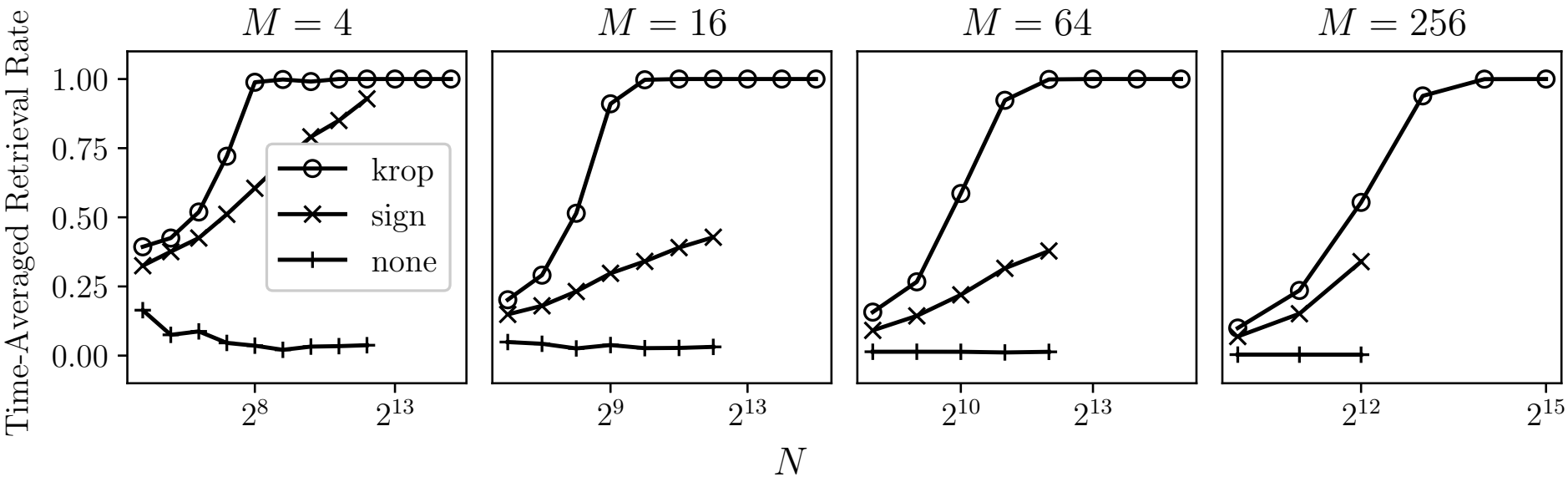
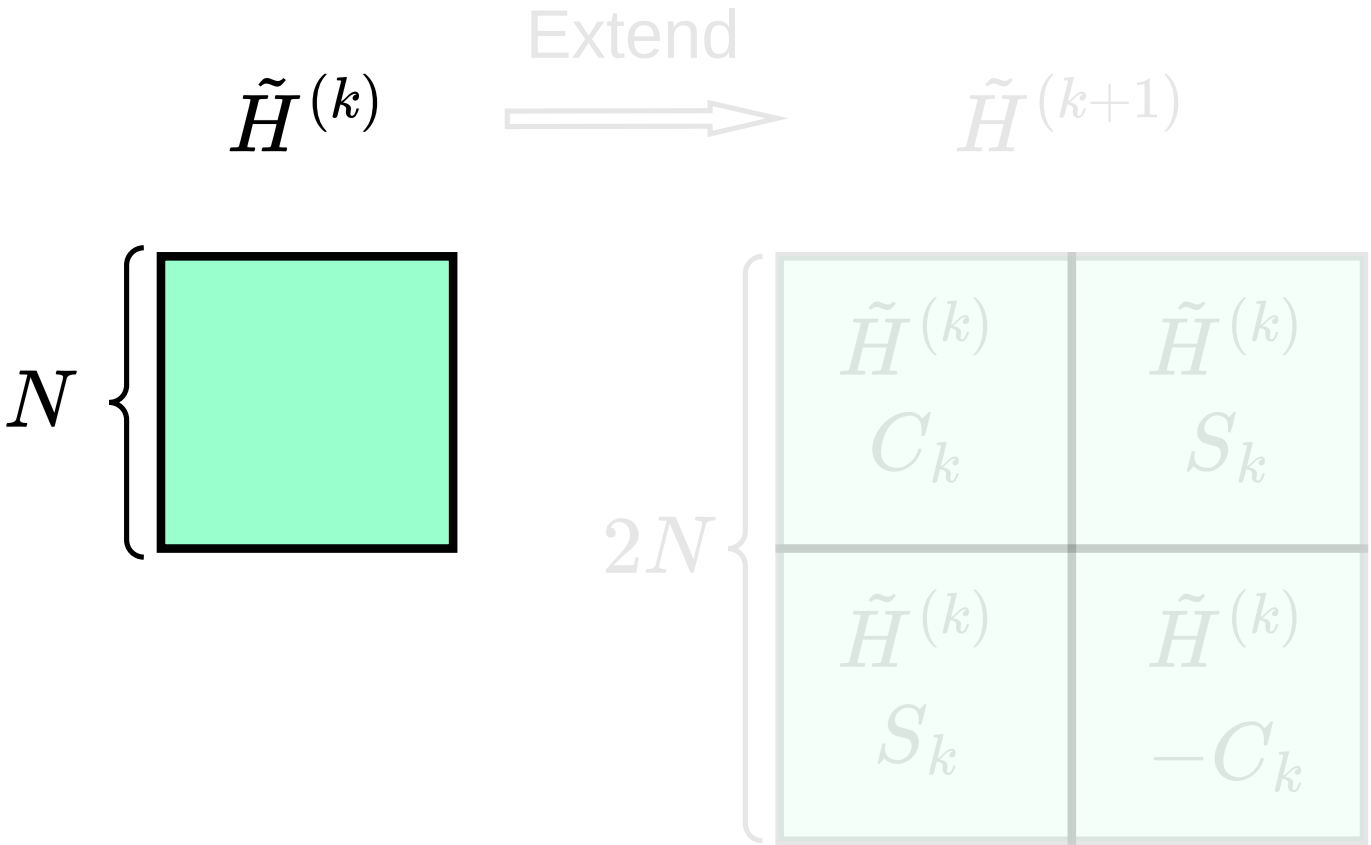


Figure: Mutable VSA memory retrieval rates averaged over 10 independent trials and 30 read-write steps.

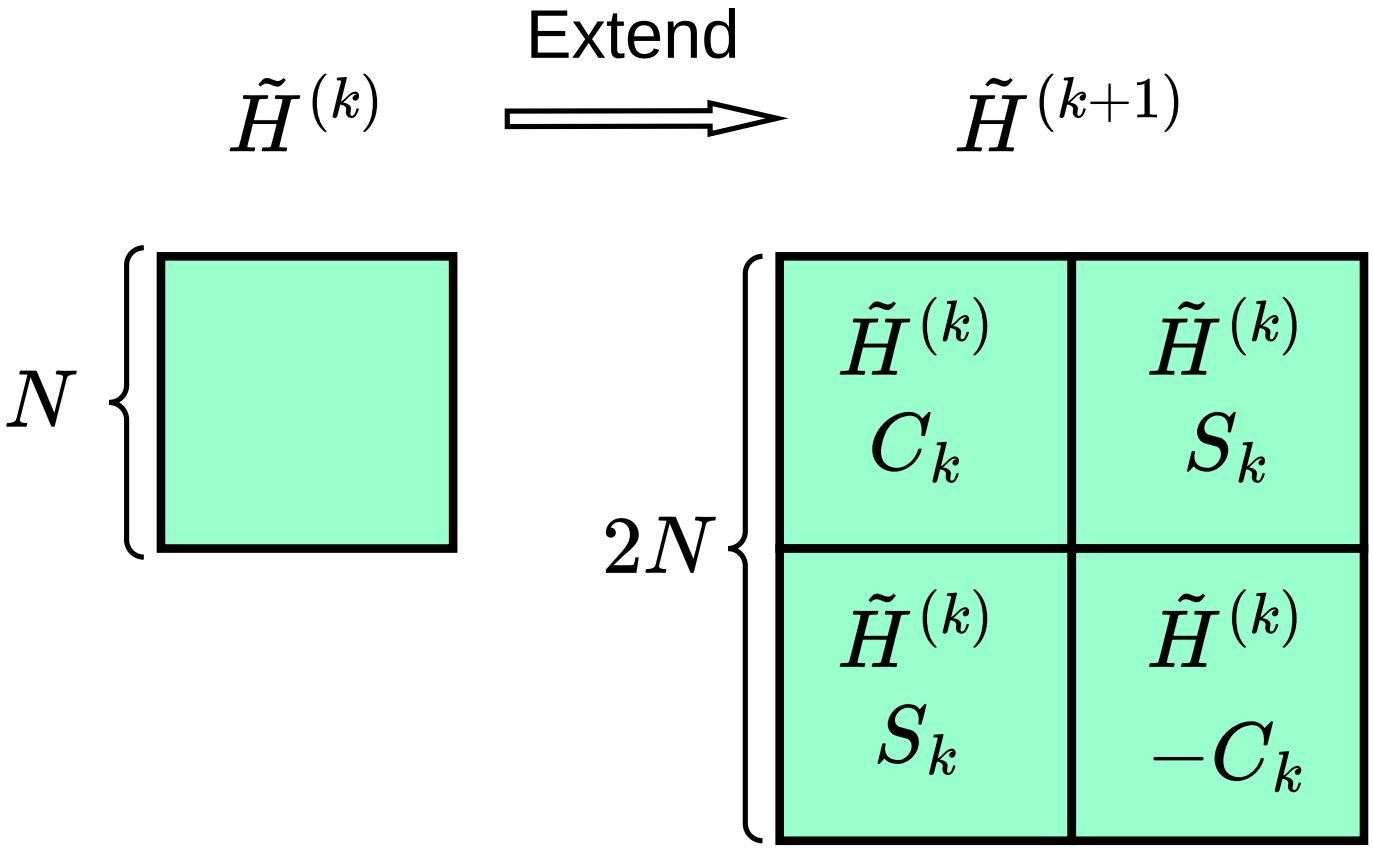
Limitation

`krop` improves clean-up efficiency, but...

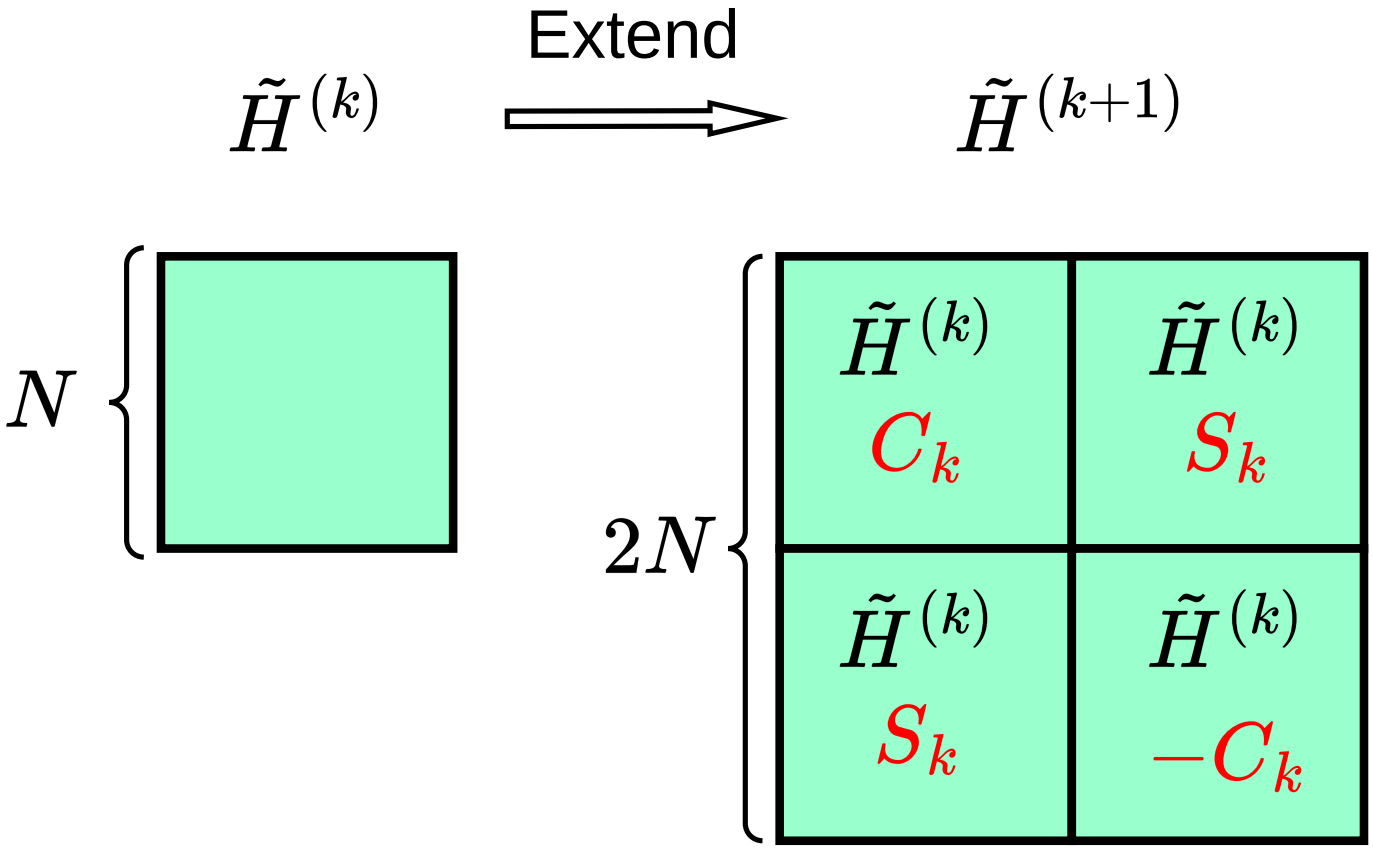
Distributional Miss Match



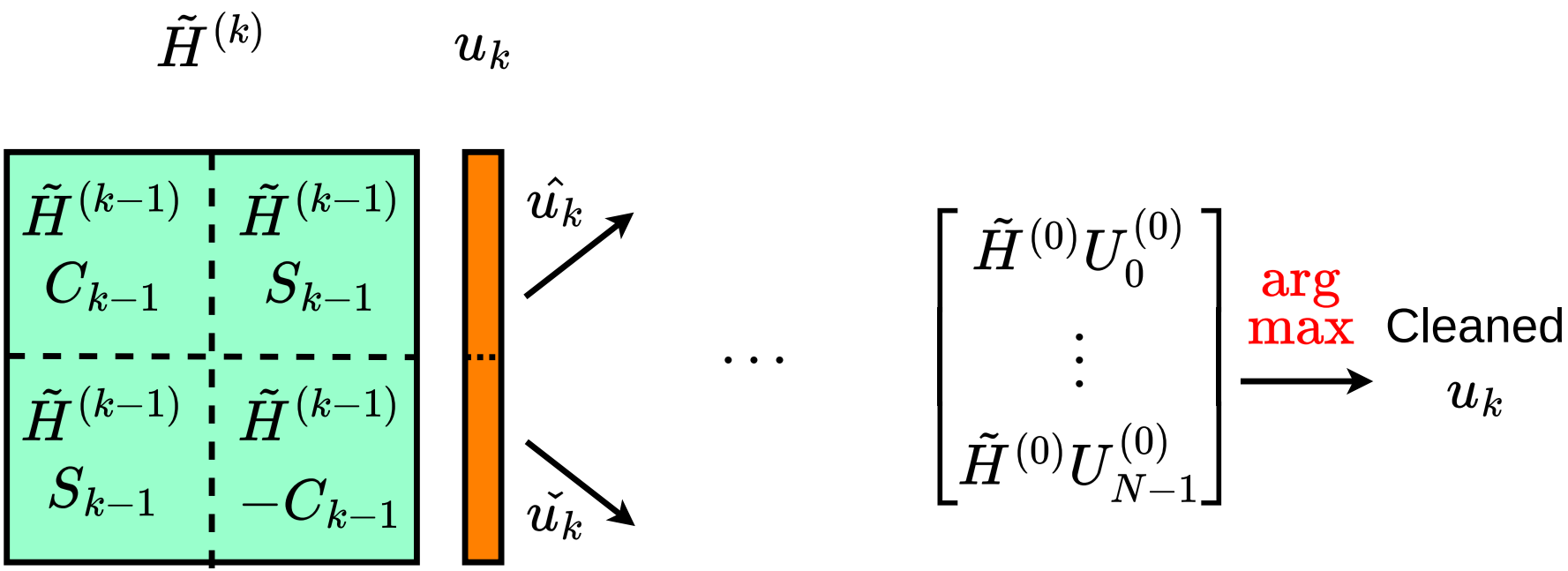
Distributional Miss Match



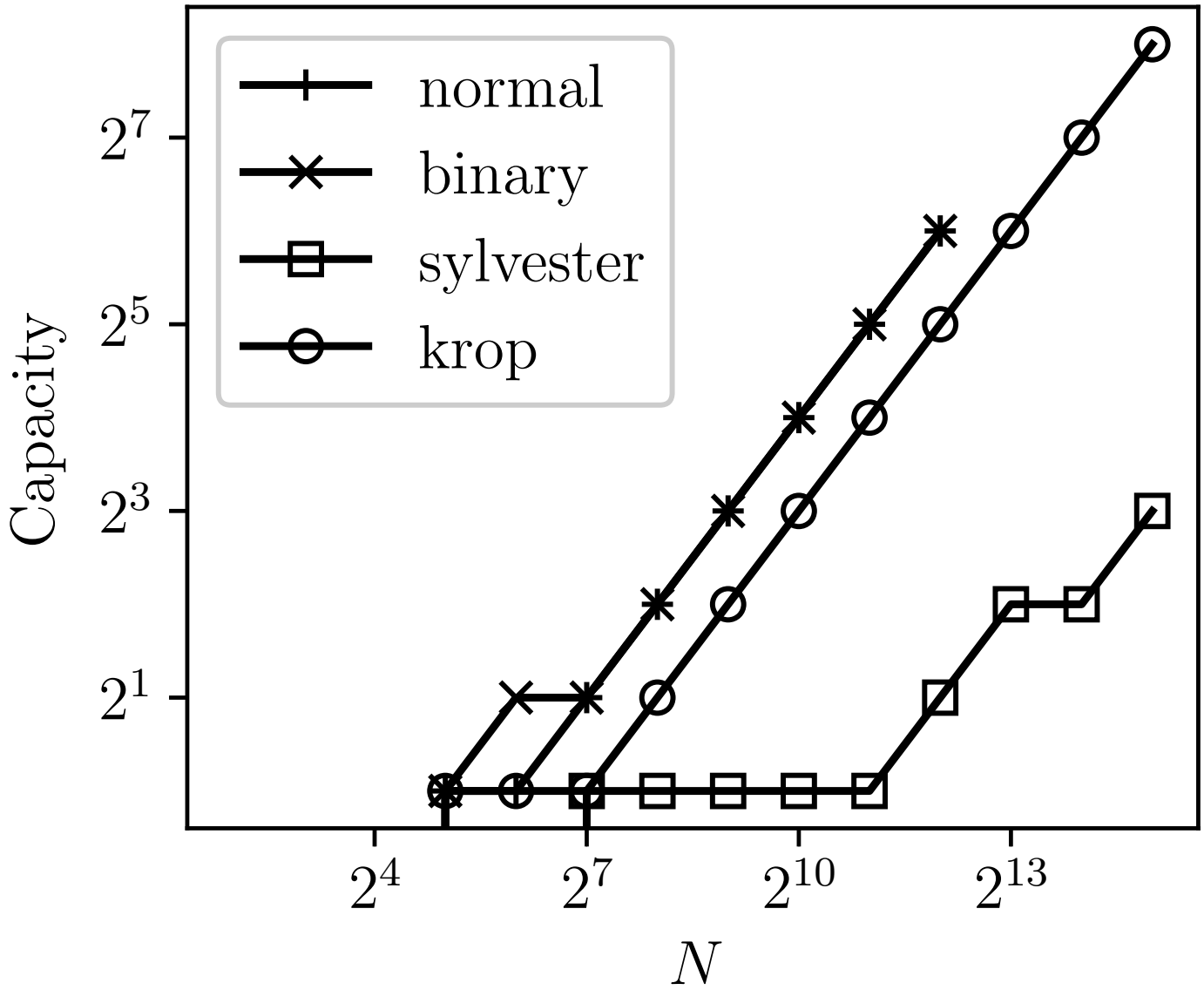
Distributional Miss Match



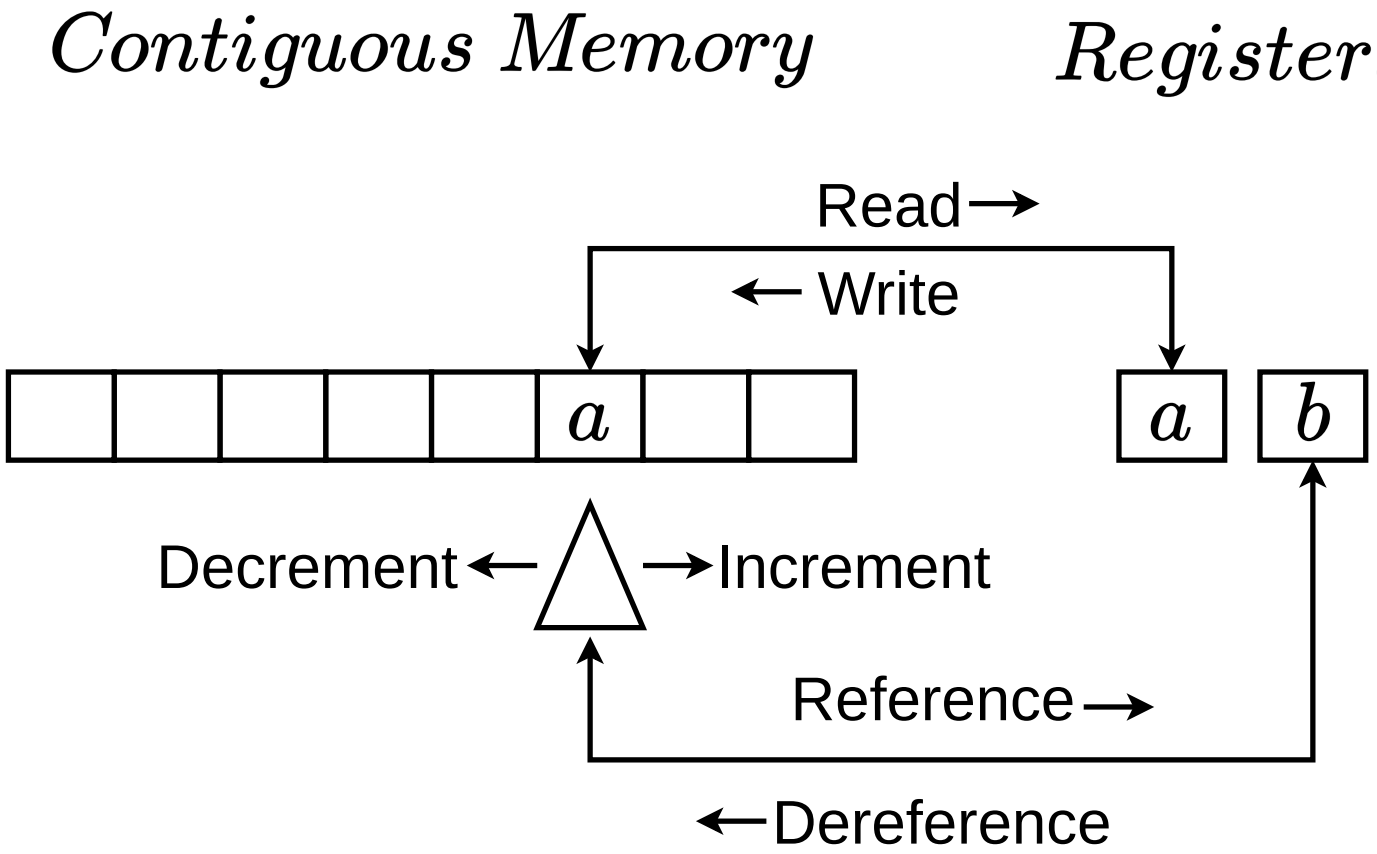
Non-differentiability



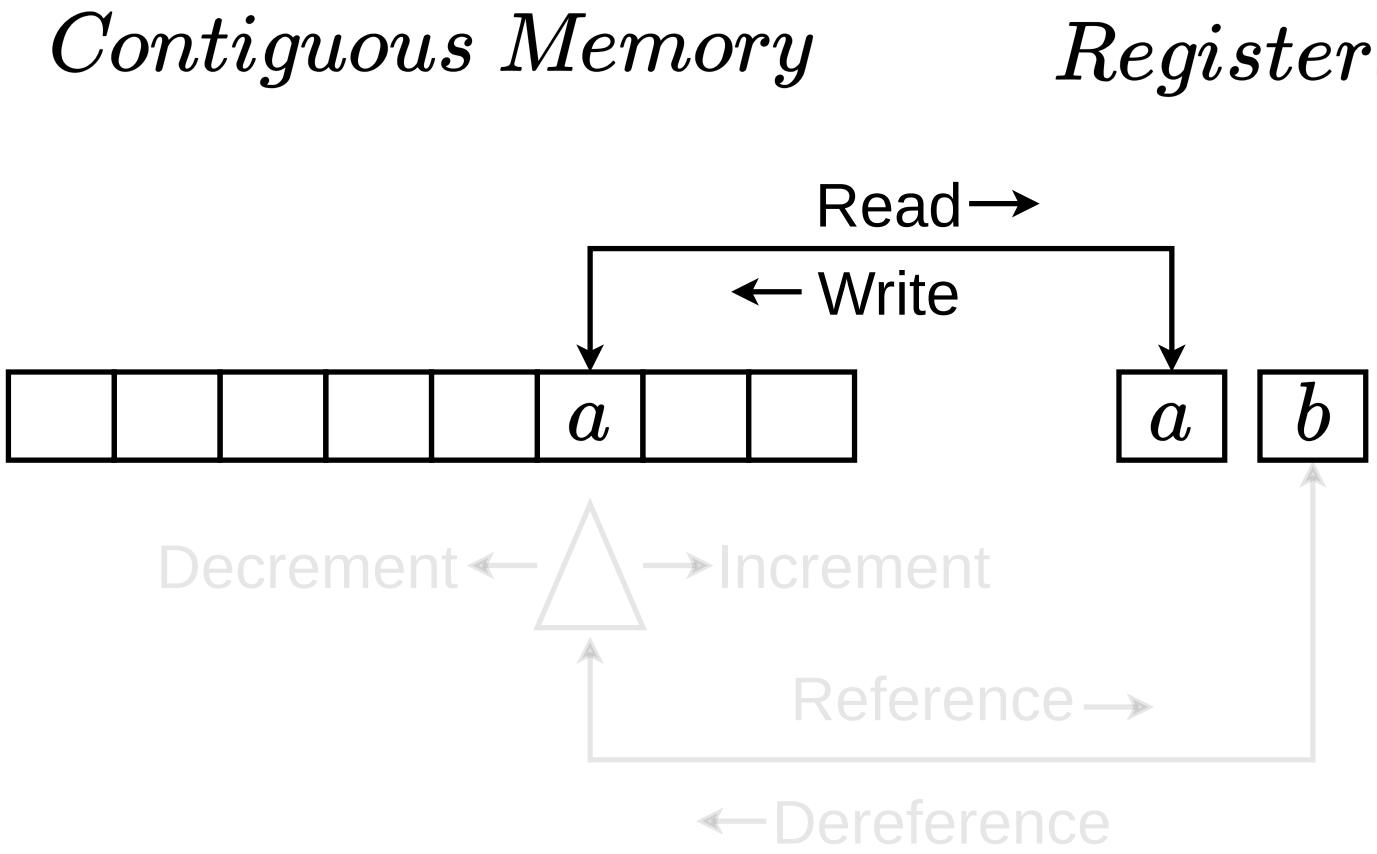
Required Large Dimension



Pointer Issue



Pointer Issue



Conclusion

krop is:

A codebook with linearithmic clean-up for VSA key-value memory.

Conclusion

krop is:

A codebook with linearithmic clean-up for VSA key-value memory.

It provides:

Compact codebook storage and efficient vector reconstruction.

Conclusion

`krop` is:

A codebook with linearithmic clean-up for VSA key-value memory.

It provides:

Compact codebook storage and efficient vector reconstruction.

Remaining limitations:

Distribution mismatch, non-differentiability, high-dimensional requirements and typing issue.

Thank You

Questions and Discussion

Contact: rliu02@syr.edu

Ruipeng Liu

Garrett Katz

References


J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities." *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.


A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.


A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," 2014. [Online]. Available: https://arxiv.org/abs/1410.5401


G. E. Katz, G. P. Davis, R. J. Gentili, and J. A. Reggia, "A programmable neural virtual machine based on a fast store-erase learning rule," *Neural Networks*, vol. 119, pp. 10–30, 2019.


R. Liu, B. He, N. Tahir, and G. E. Katz, "On the feasibility of single-pass full-capacity learning in linear threshold neurons with binary input vectors," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR, 21–27 Jul 2024, pp. 31 119–31 130. [Online]. Available: https://proceedings.mlr.press/v235/liu24x.html

References

R. Liu, Q. Qiu, S. Khan, and G. E. Katz, "Linearithmic clean-up for vector-symbolic key-value memory with kroneker rotation products," in *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning*, ser. Proceedings of Machine Learning Research, L. H. Gilpin, E. Giunchiglia, P. Hitzler, and E. van Krieken, Eds., vol. 284. PMLR, 08–10 Sep 2025, pp. 1107–1118. [Online]. Available: https://proceedings.mlr.press/v284/liu25b.html

T. A. Plate, "Holographic reduced representations," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 623–641, 1995.