

State Space Models 101

Shubham Gupta



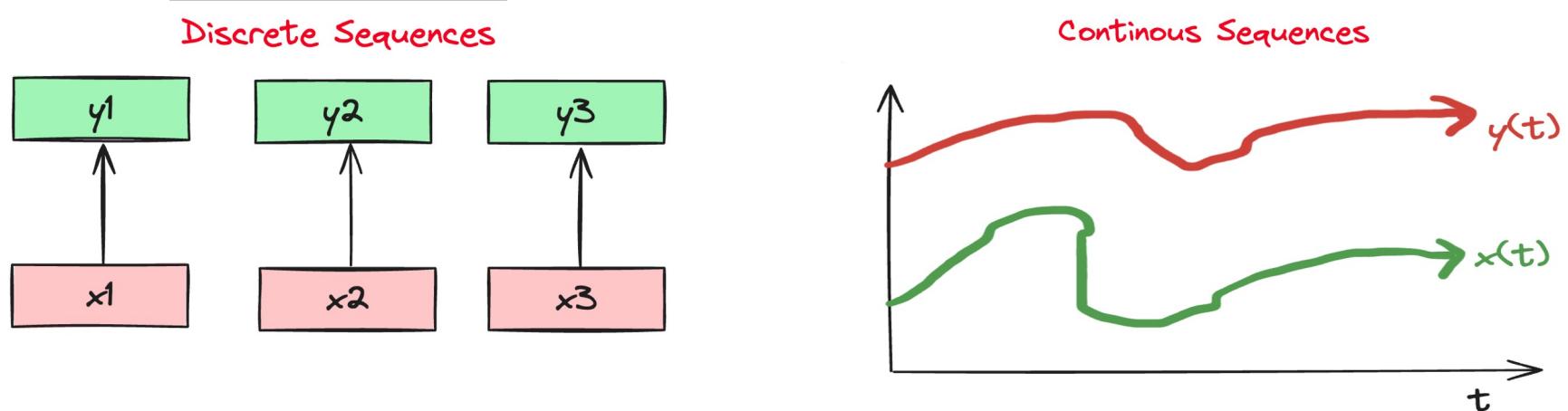
in/shubhamgupta2208



shubhamg.in

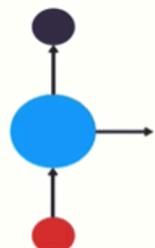
Sequence Modelling: Recap

Goal: Map an input sequence to an output sequence



Sequence Modelling: Models

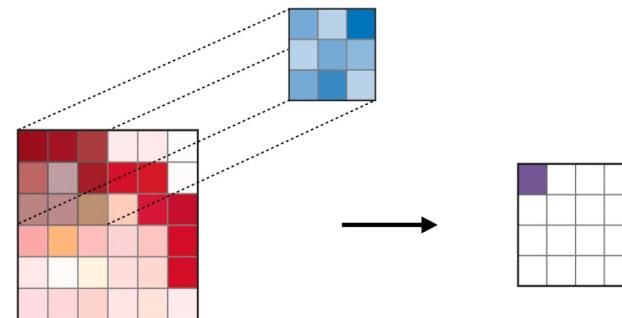
RNN



Source: Medium

Training: $O(N)$ ||
Inference: $O(N)$
Performance:

CNN

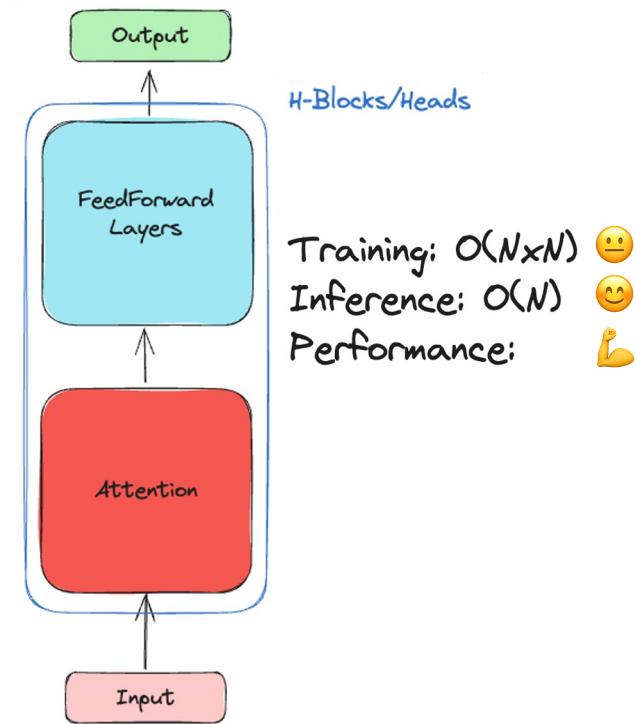
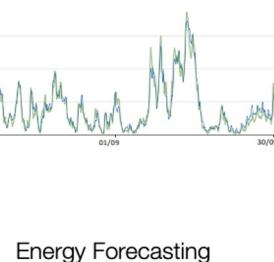
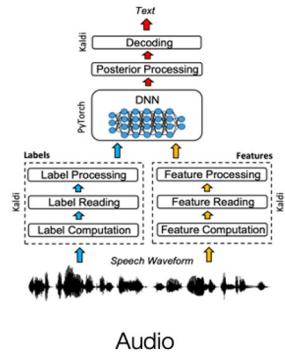
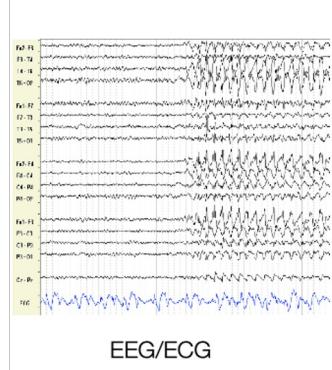


Source: Medium

Training: $O(N)$ ||
Inference: $O(N)$
Performance:

Sequence Modelling: Transformers

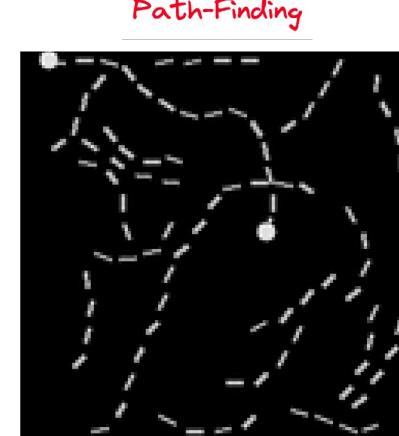
- SoTA on sequence modelling tasks
- Struggle to scale over long sequences
- Why do we care?
 - Enable new capabilities
 - Model other sequential data



Long Range Arena Benchmark

- LRA [Yi Tay et al., 2020]
- Measure long-context model quality
- Multiple input modalities
- Total tasks: 6
- Input sequence length: 1k-16k
- Transformer Performance:
 - Avg. Accuracy: **52%** 🤦
 - Unable to solve Path-X

ListOps
[MAX 4 3 [MIN 2 3] 1 0 [MEDIAN 1 5 8 9, 2]]
Output: 5



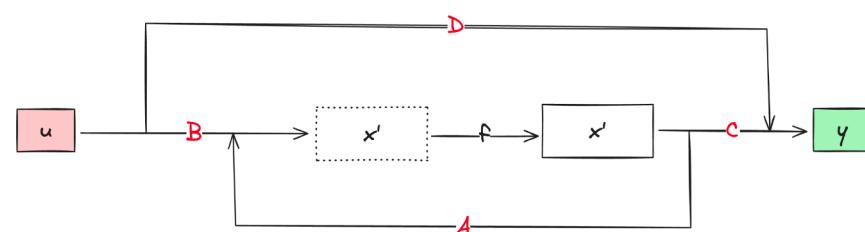
Ideal Model

Training: $O(N)$ 😊 ||
Inference: $O(N)$ 😊
Performance: 💪

Obtain best of all
models for long-
context?

State Space Models

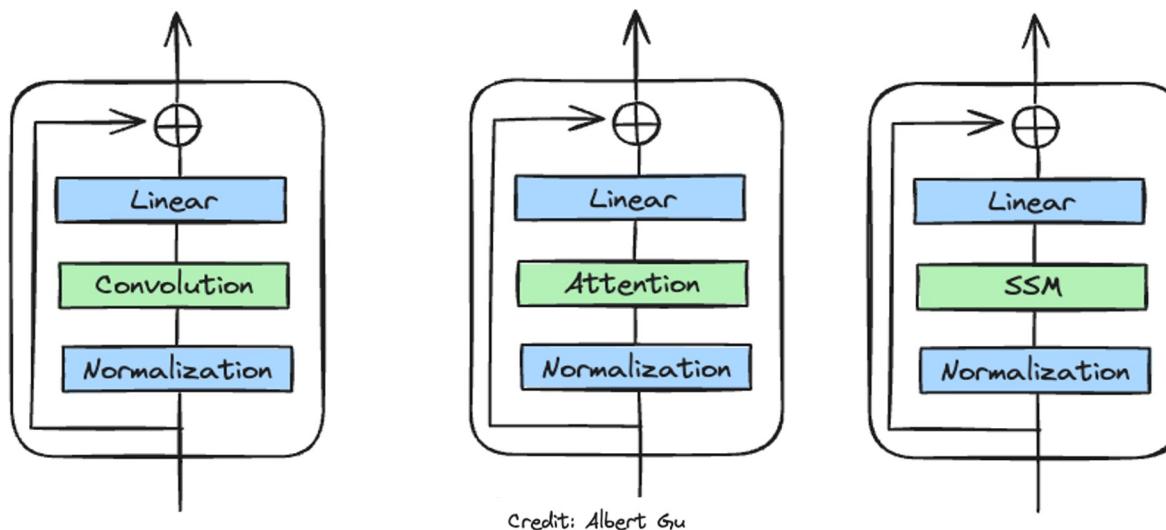
- Kalman, 1960
- Used in control theory/signal processing
- Modelled as *continuous-timed* differential equation
- 1-layer, Linear Model
- Time - Invariant



$$x'(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx'(t) + Du(t)$$

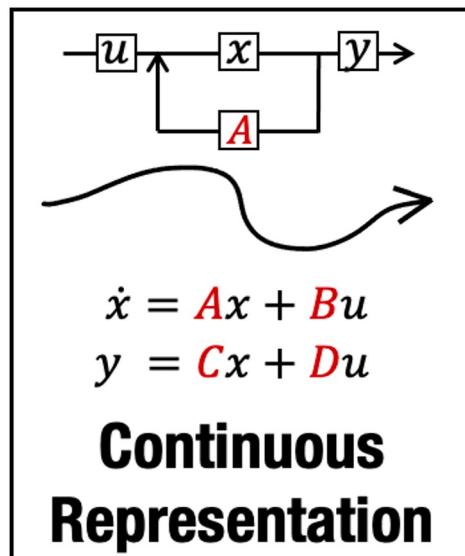
Deep SSMs

- Deep, non-linear model
- Deterministic transformation

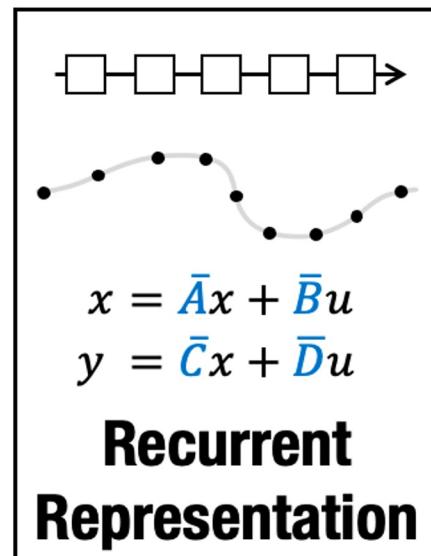


SSM Properties

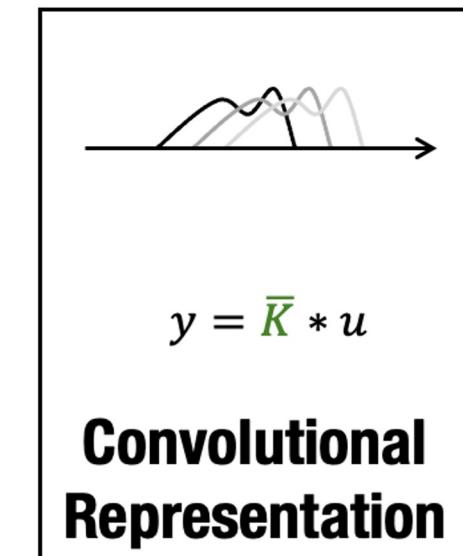
Operates on signals/sequences



Efficient online computation



Efficient parallelizable computation



Discretize
→

Unroll
→

Source: Albert Gu

SSM: Recurrent

$$x'(t) = \mathbf{A}x(t) + \mathbf{B}u(t)$$

$$y(t) = \mathbf{C}x'(t) + \mathbf{D}u(t)$$

Discretize:

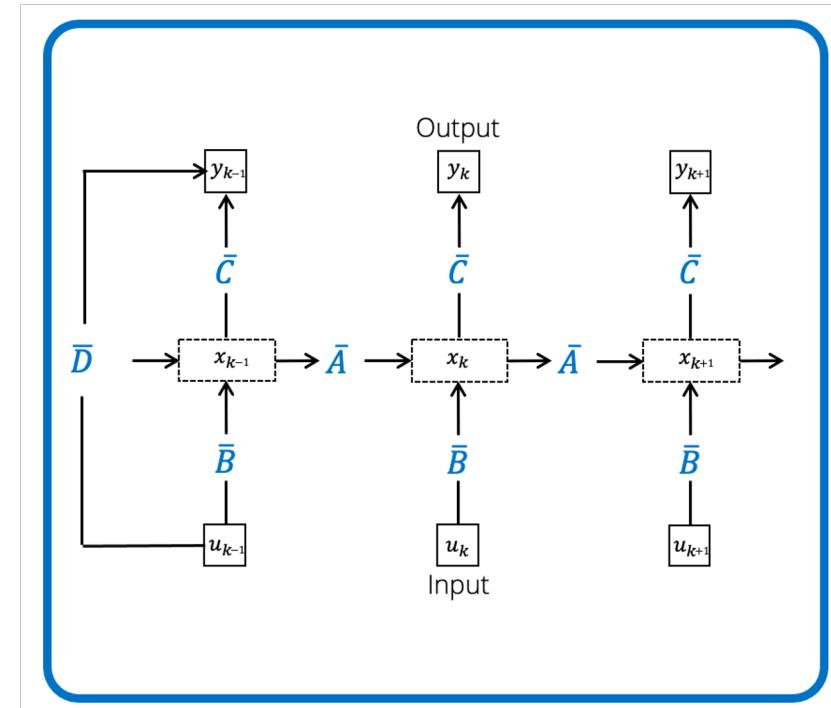
$$\bar{\mathbf{A}} = I + \Delta A$$

Recurrent "hidden state"

$$x_k = \bar{\mathbf{A}}x_{k-1} + \bar{\mathbf{B}}u_k$$

Out Projection

$$y_k = \bar{\mathbf{C}}x_k + \bar{\mathbf{D}}u_k$$



Source: Albert Gu

SSM: Convolutional

$$x_k = \bar{A}x_{k-1} + \bar{B}u_k \quad y_k = \bar{C}x_k + \bar{D}u_k$$

Expand the terms

$$x_0 = \bar{B}u_0 \quad x_1 = \bar{A}\bar{B}u_0 + \bar{B}u_1 \quad x_2 = \bar{A}^2\bar{B}u_0 + \bar{A}\bar{B}u_1 + \bar{B}u_2$$

Output will be a linear projection of state

$$y_0 = \bar{C}\bar{B}u_0 + \bar{D}u_0$$

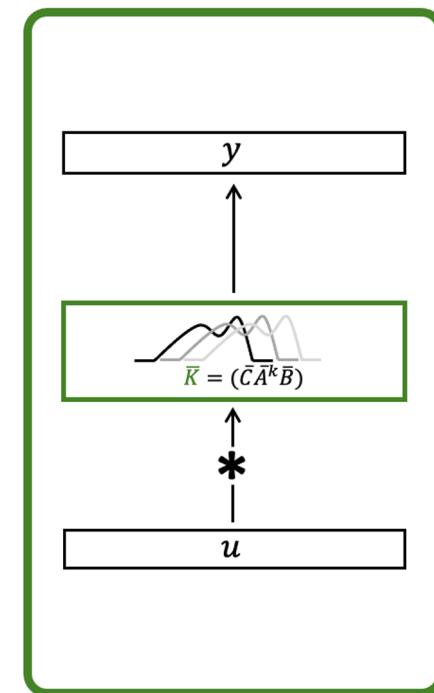
$$y_1 = \bar{C}\bar{A}\bar{B}u_0 + \bar{C}\bar{B}u_1 + \bar{D}u_1$$

$$y_2 = \bar{C}\bar{A}^2\bar{B}u_0 + \bar{C}\bar{A}\bar{B}u_1 + \bar{C}\bar{B}u_2 + \bar{D}u_2$$

No non-linearity => Simple computation

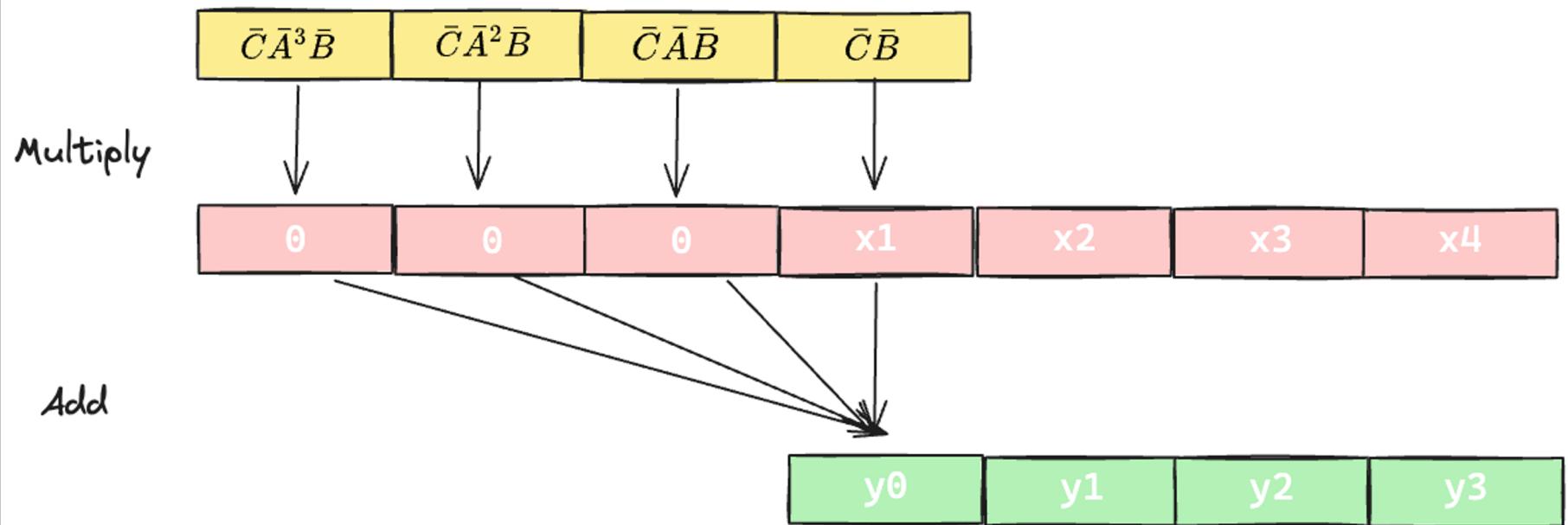
Extracting common coefficients, we get the SSM Kernel

$$\bar{K} = (\bar{C}\bar{A}^i\bar{B})_{i \in L}$$

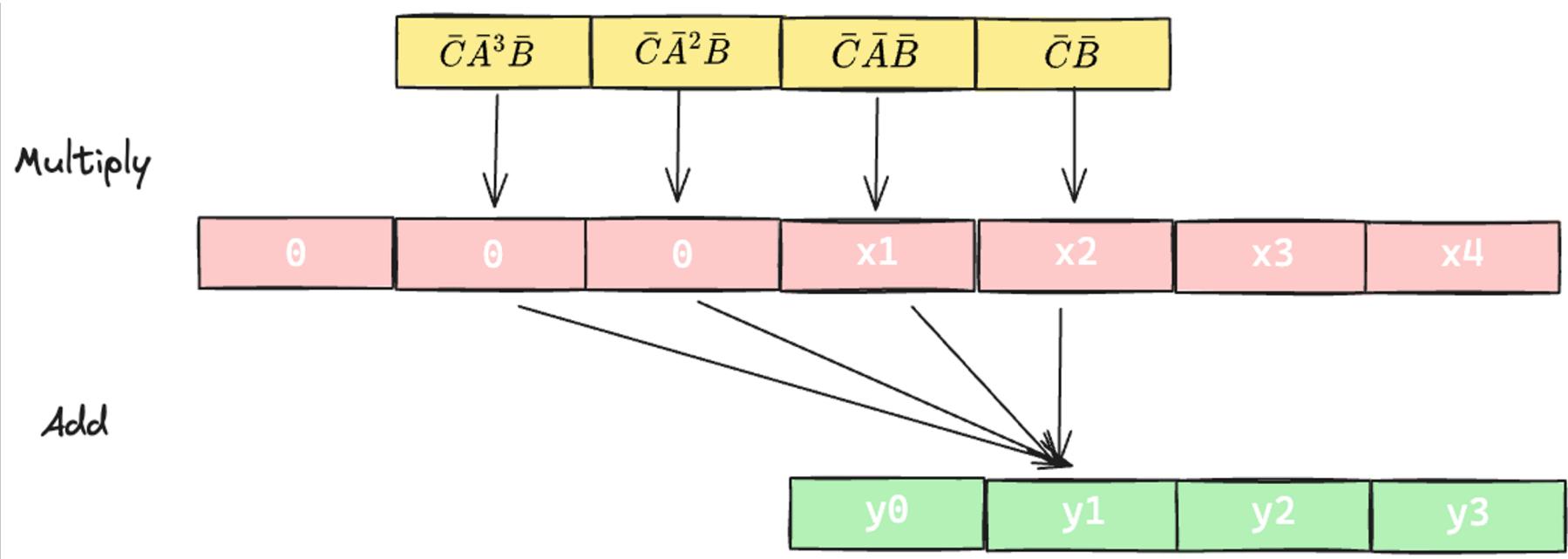


Source: Albert Gu

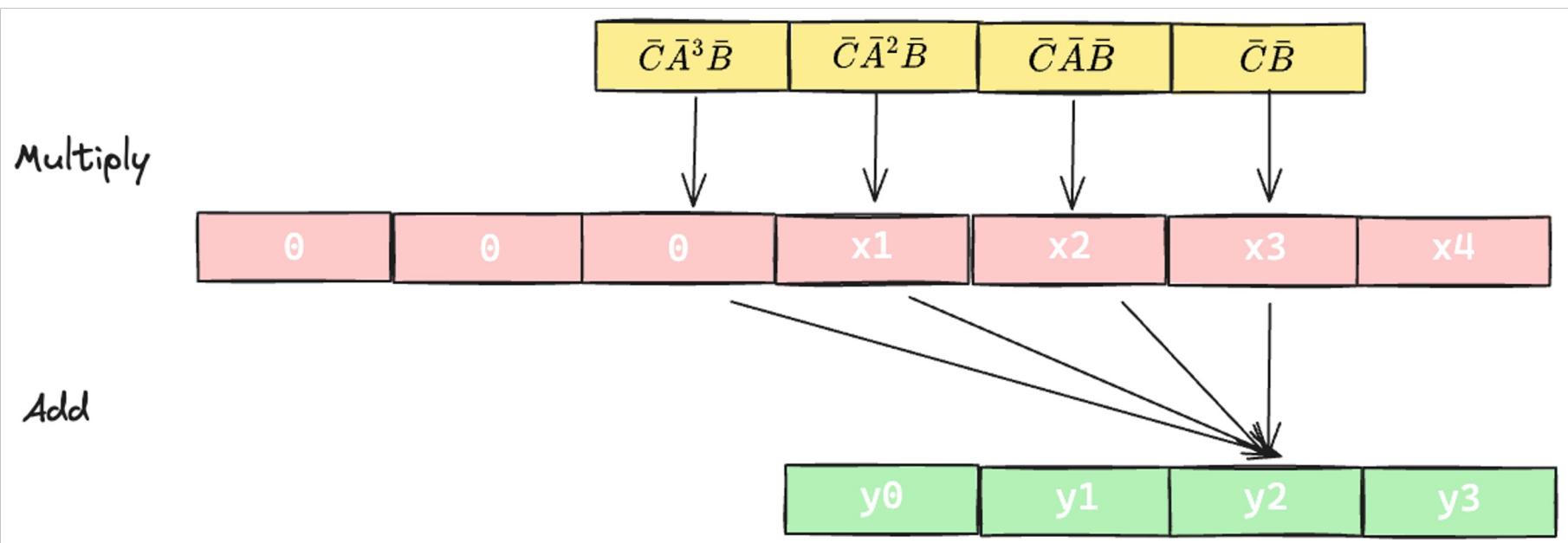
SSM: Convolutional Kernel



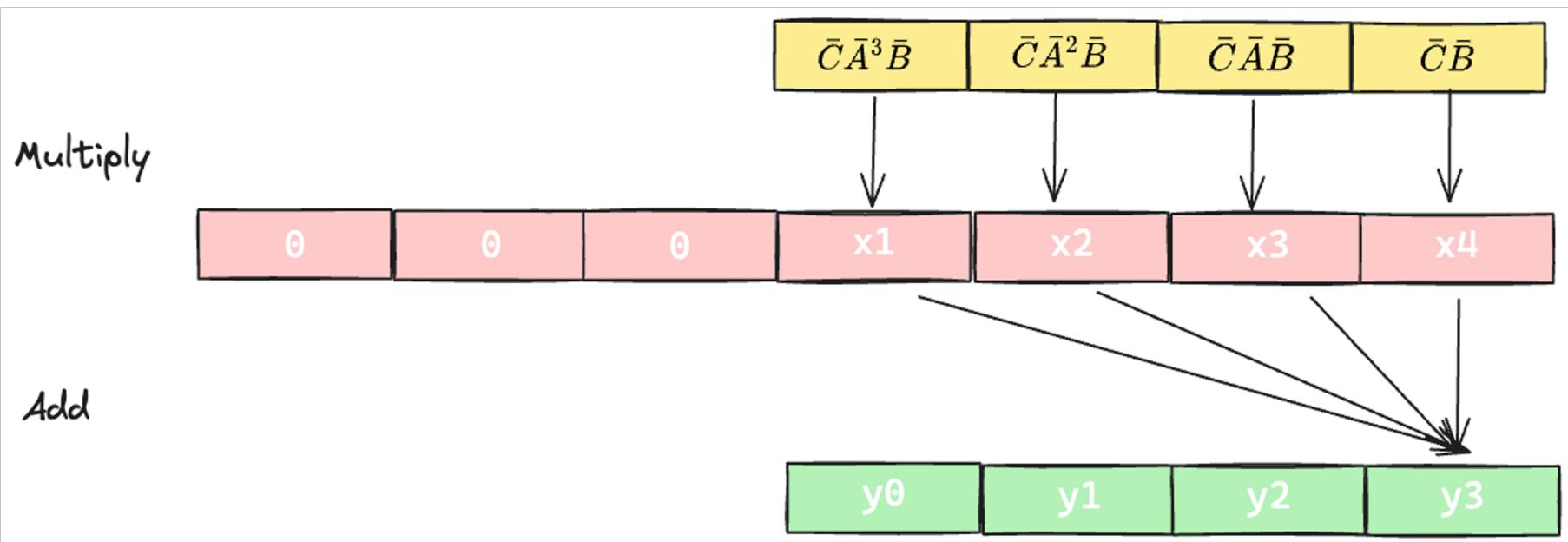
SSM: Convolutional Kernel



SSM: Convolutional Kernel



SSM: Convolutional Kernel



Deep SSM: Challenges

➤ Modelling Challenge

- SSMs inherit problems of CNN,RNN on LRA
- Random init A
 - 60% acc sequential MNIST 😞

➤ Computation Challenge

- SSM has nice properties if \bar{A} and \bar{K} are known
- Computing them is **hard!**

➤ Computing the Kernel

- A power  -> vanishing gradient?
- A power  -> $O(D^2N)$ computation
 - Ideal -> $O(N)$

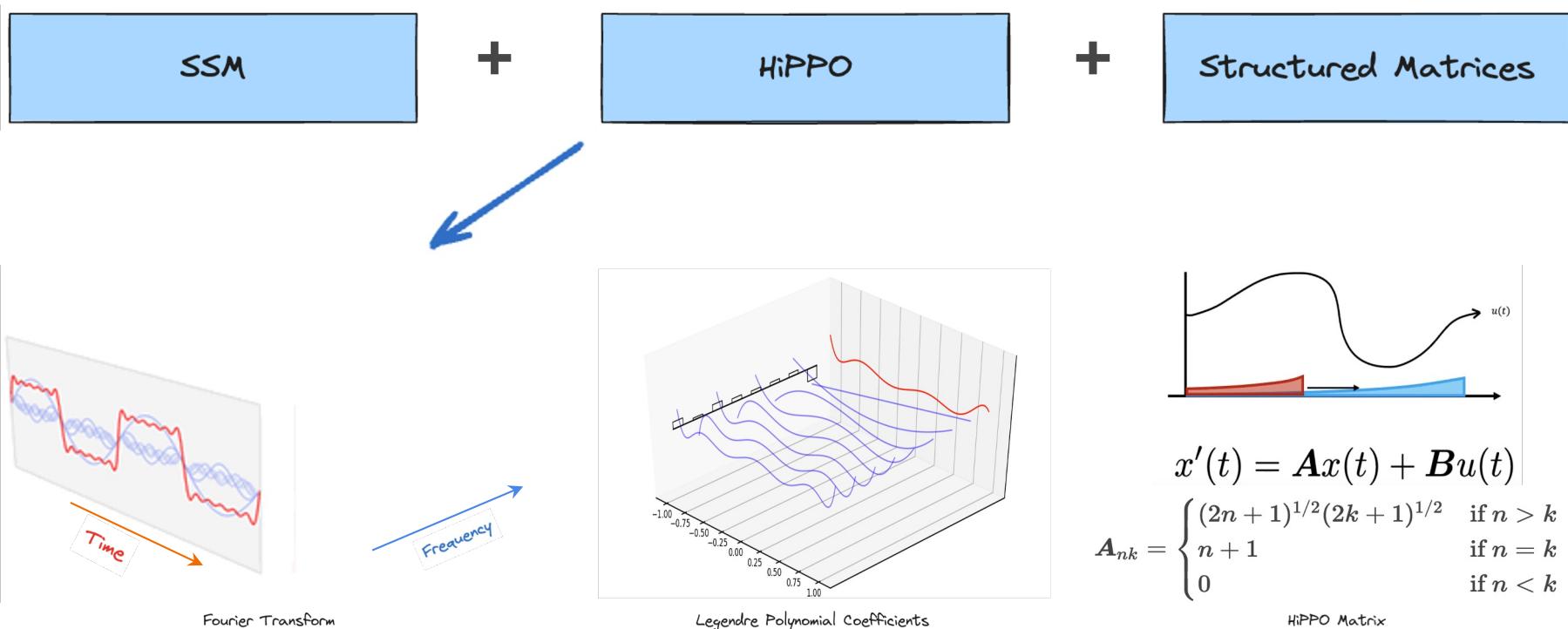
$$x_k = \bar{A}x_{k-1} + \bar{B}u_k$$

$$y_k = \bar{C}x_k + \bar{D}u_k$$

$$\bar{K} = (\bar{C}\bar{A}^i\bar{B})_{i \in L}$$

D - SSM Hidden Dimension

S4: Structured SSM



S4: LRA Results

MODEL	LISTOPS	TEXT	RETRIEVAL	IMAGE	PATHFINDER	PATH-X	AVG
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Reformer	<u>37.27</u>	56.10	53.40	38.07	68.50	✗	50.56
BigBird	36.05	64.02	59.29	40.83	74.87	✗	54.17
Linear Trans.	16.13	<u>65.90</u>	53.09	42.34	75.30	✗	50.46
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.18
FNet	35.33	65.11	59.61	38.67	<u>77.80</u>	✗	54.42
Nyströmformer	37.15	65.52	<u>79.56</u>	41.58	70.94	✗	57.46
Luna-256	37.25	64.57	79.29	<u>47.38</u>	77.72	✗	<u>59.37</u>
S4	59.60	86.82	90.90	88.65	94.20	96.35	86.09

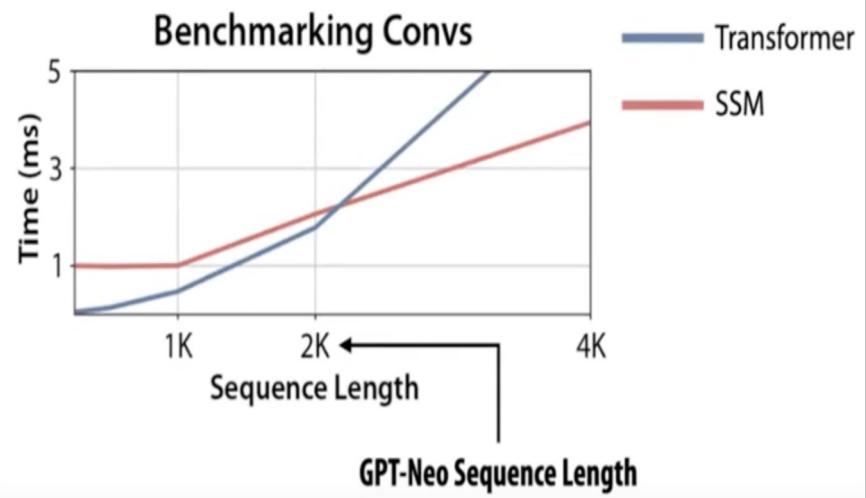
S4: LRA Results

S4: Problems

Language Modelling

Model	PPL(OWT)
Transformer	20.6
S4D	24.9
GSS	24.0

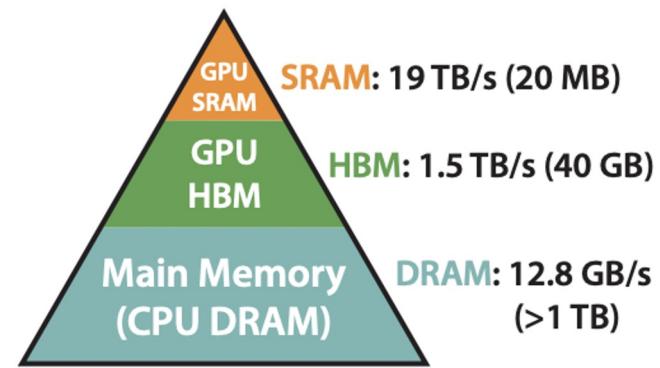
Short Sequence Efficiency



Source: Albert Gu

Background: GPU Memory

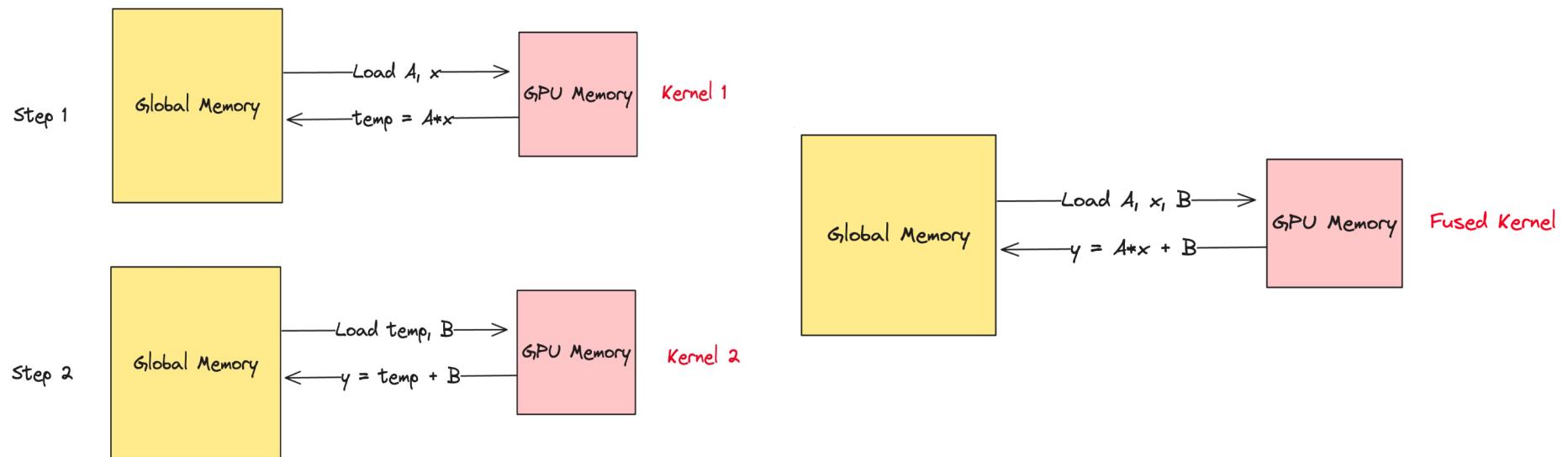
- Components:
 - DRAM
 - HBM
 - SRAM
- Problems:
 - Size
 - Speed
- Optimizations:
 - Kernel Fusion
 - Recomputation
 - Parallelism



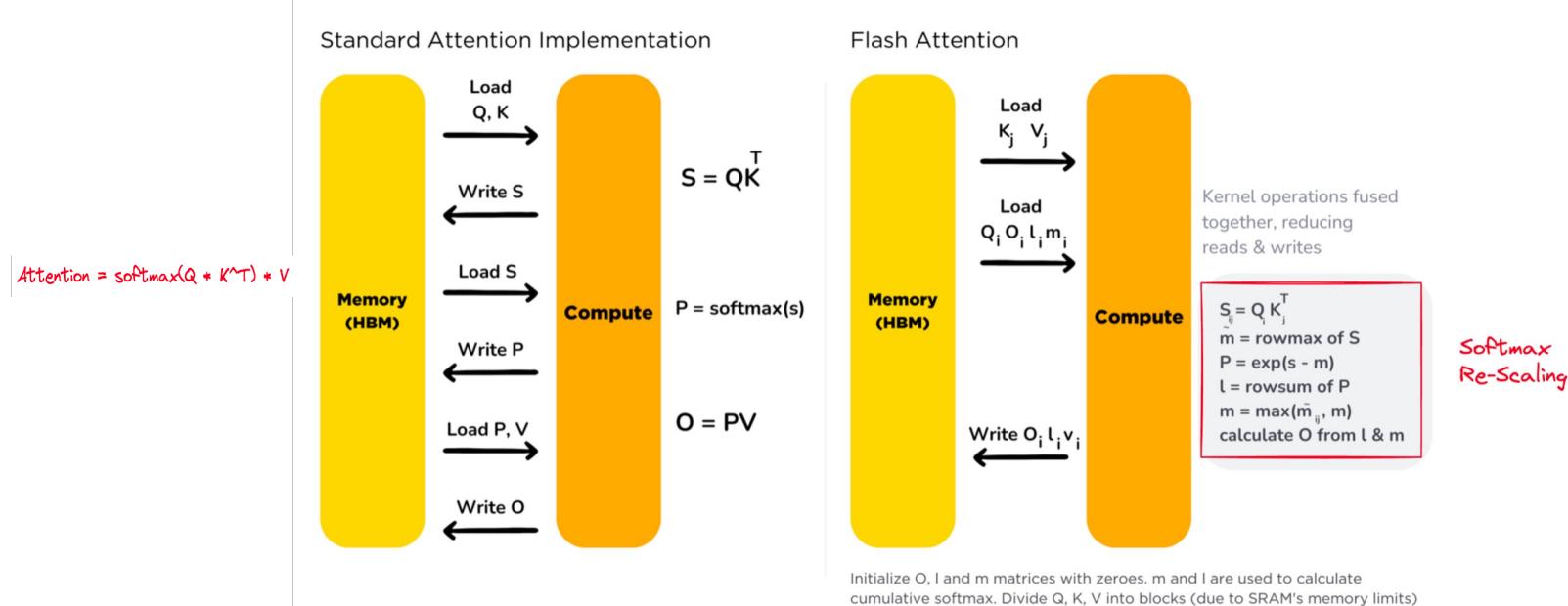
	A100 40GB PCIe	A100 80GB PCIe	A100 40GB SXM	A100 80GB SXM
GPU Memory	40GB HBM2	80GB HBM2e	40GB HBM2	80GB HBM2e
GPU Memory Bandwidth	1,555GB/s	1,935GB/s	1,555GB/s	2,039GB/s

GPU: Kernel Fusion

$$y = Ax + B$$

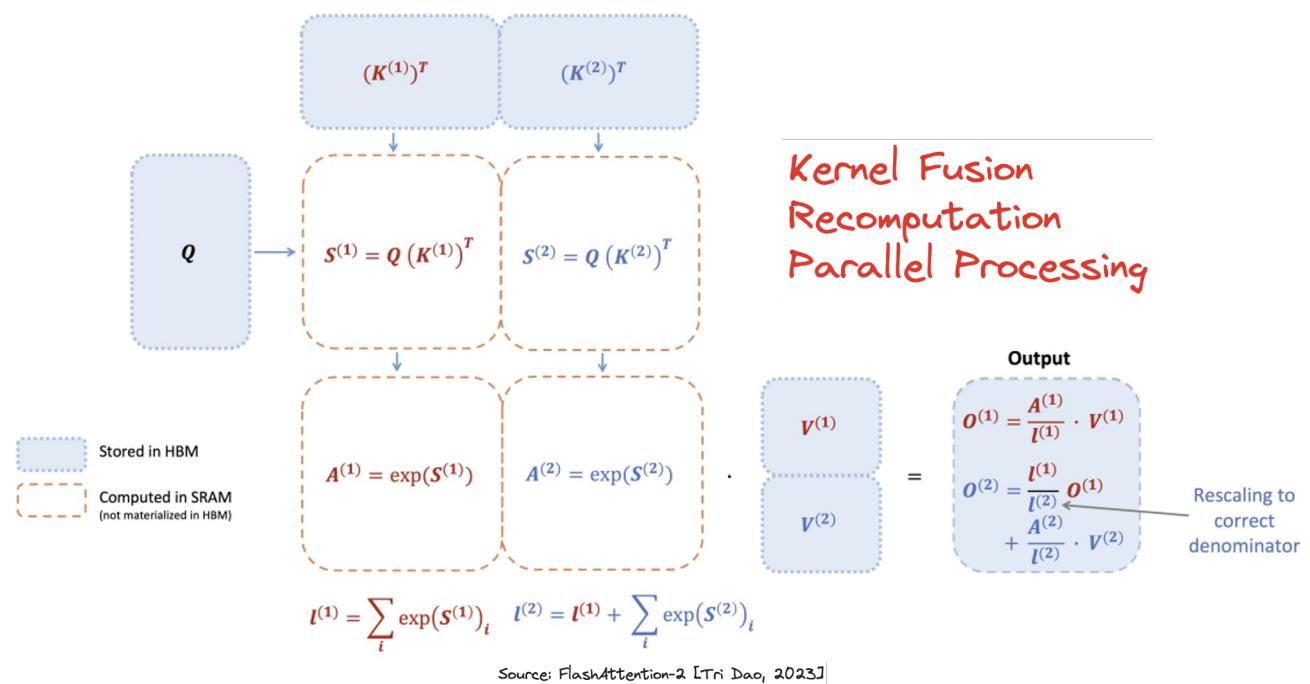


Flash Attention: Kernel Fusion



Source: HuggingFace

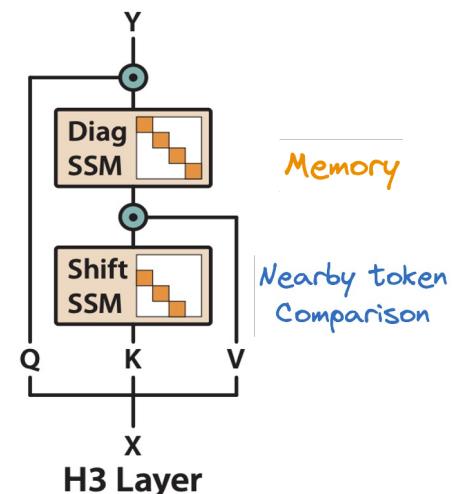
Flash Attention: Implementation



Hungry-Hungry HiPOOs (H3)

- Improve SSM on synthetic tasks
- Combine two SSMs in a single network
- Replaces all Transformer layers, except two
- Trained using FlashConv + Fusion

Task	Input	Output
Induction Head	a b c d e l - f g h i ... x y z l	f
Associative Recall	a 2 c 4 b 3 d 1 a	2



H3: Results

Language Modelling

Model	PPL(OWT)
Transformer	20.6
S4D	24.9
GSS	24.0
H3	21.0
H3 (with two attention blocks)	19.6

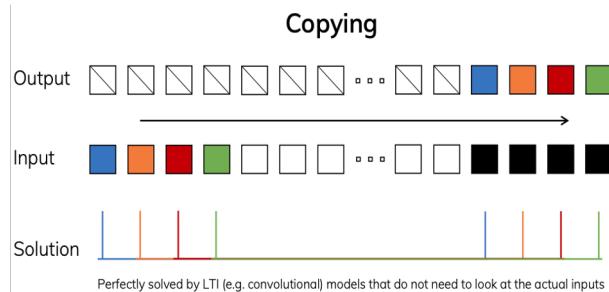
Source: HazyResearch

H3: Scaling Results

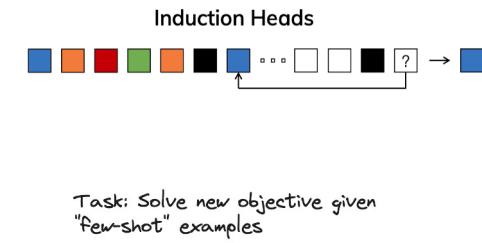
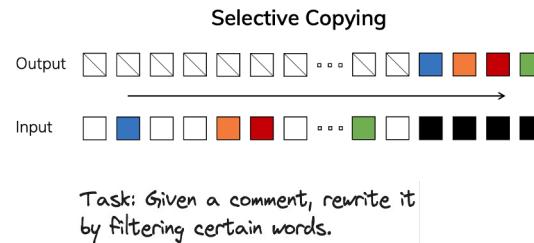
Model	Pile	OpenWebText	WikiText103
GPT-2 small (125M)	19.0*	22.6	29.9
GPT-Neo-125M	9.4	22.6	26.3
Hybrid H3-125M	8.8	20.9	23.7
GPT-2 medium (355M)	13.9*	17.0	21.8
Hybrid H3-355M	7.1	15.9	16.9
GPT-2 XL (1.5B)	12.4*	12.9	17.0
GPT-Neo-1.3B	6.2	13.1	13.3
Hybrid H3-1.3B	6.0	12.4	12.5
GPT-Neo-2.7B	5.7	11.7	11.5
Hybrid H3-2.7B	5.4	11.0	10.6

Model Perplexity [Dan Fu et al., 2022]

S4: Additional Problems



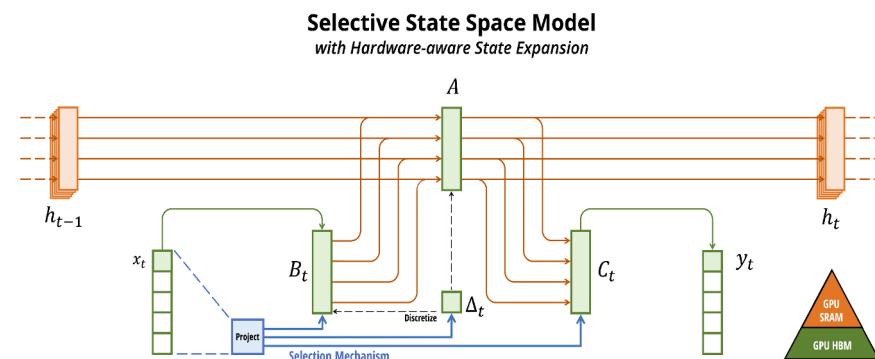
Task: Rewrite input one token at a time, but time-shifted



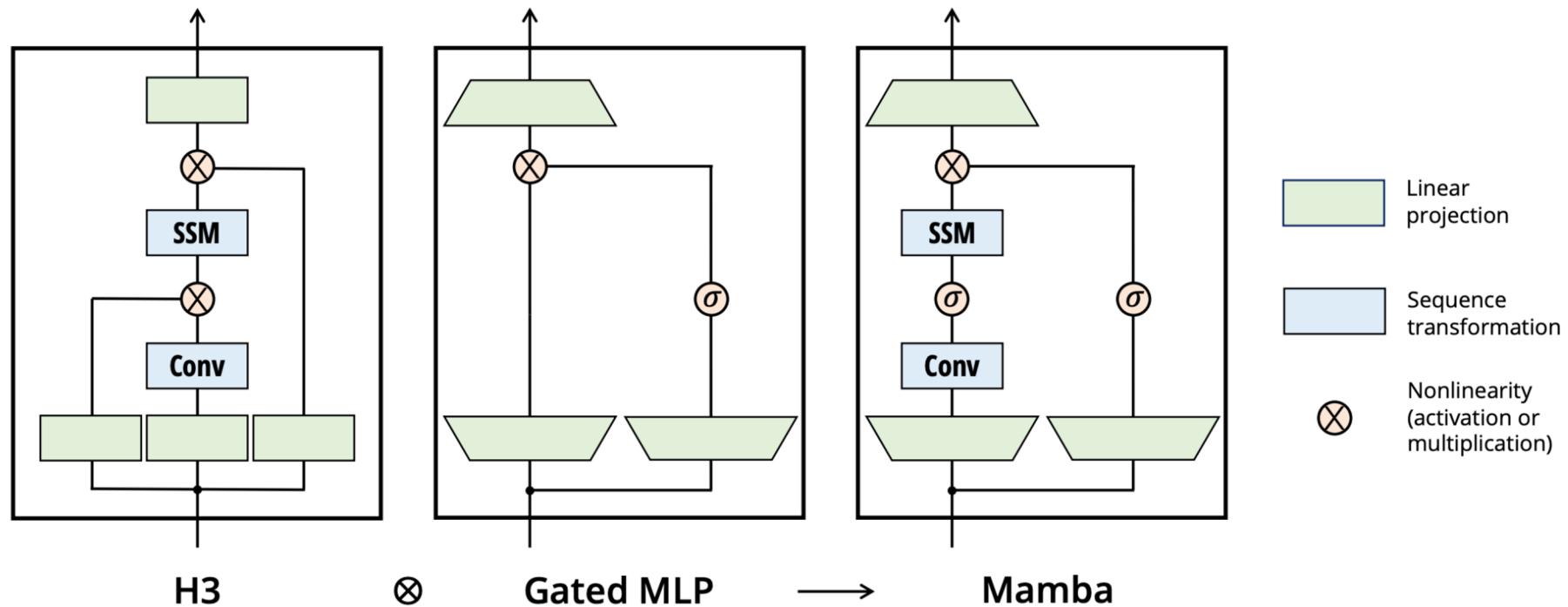
\bar{A} , \bar{B} and \bar{C} are independent of input

Mamba(S6): Selective SSM

- SSSSSS...🐍
- Improves SSM performance on copying tasks
- Handles input data that is varying in time
- Only supports recurrent form



Mamba: Architecture



Source: Mamba [Albert Gu, Tri Dao 2023]

Mamba: Implementation

Algorithm 1 SSM (S4)

Input: $x : (B, L, D)$
Output: $y : (B, L, D)$
1: $A : (D, N) \leftarrow \text{Parameter}$
 ▷ Represents structured $N \times N$ matrix
2: $B : (D, N) \leftarrow \text{Parameter}$
3: $C : (D, N) \leftarrow \text{Parameter}$
4: $\Delta : (D) \leftarrow \tau_\Delta(\text{Parameter})$
5: $\bar{A}, \bar{B} : (D, N) \leftarrow \text{discretize}(\Delta, A, B)$
6: $y \leftarrow \text{SSM}(\bar{A}, \bar{B}, C)(x)$
 ▷ Time-invariant: recurrence or convolution
7: **return** y

Batch size: B
Prompt token size: L
Token dimensions: D

Algorithm 2 SSM + Selection (S6)

Input: $x : (B, L, D)$
Output: $y : (B, L, D)$
1: $A : (D, N) \leftarrow \text{Parameter}$
 ▷ Represents structured $N \times N$ matrix
2: $B : (B, L, N) \leftarrow s_B(x)$
3: $C : (B, L, N) \leftarrow s_C(x)$
4: $\Delta : (B, L, D) \leftarrow \tau_\Delta(\text{Parameter} + s_\Delta(x))$
5: $\bar{A}, \bar{B} : (B, L, D, N) \leftarrow \text{discretize}(\Delta, A, B)$
6: $y \leftarrow \text{SSM}(\bar{A}, \bar{B}, C)(x)$
 ▷ Time-varying: recurrence (*scan*) only
7: **return** y

Linear layers

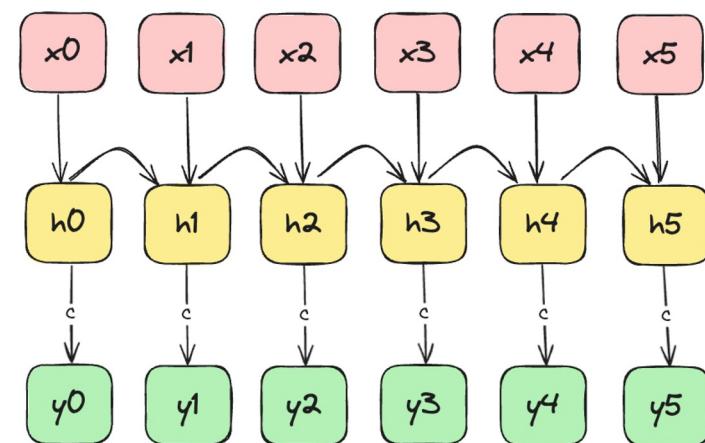
No parallel training 🤢

Source: Mamba [Albert Gu, Tri Dao 2023]

Mamba: Scan Operation

Input	1	2	3	4	5	6	7	8
Output	1	3	6	10	15	21	28	36

Current val = Sum of previous val + input

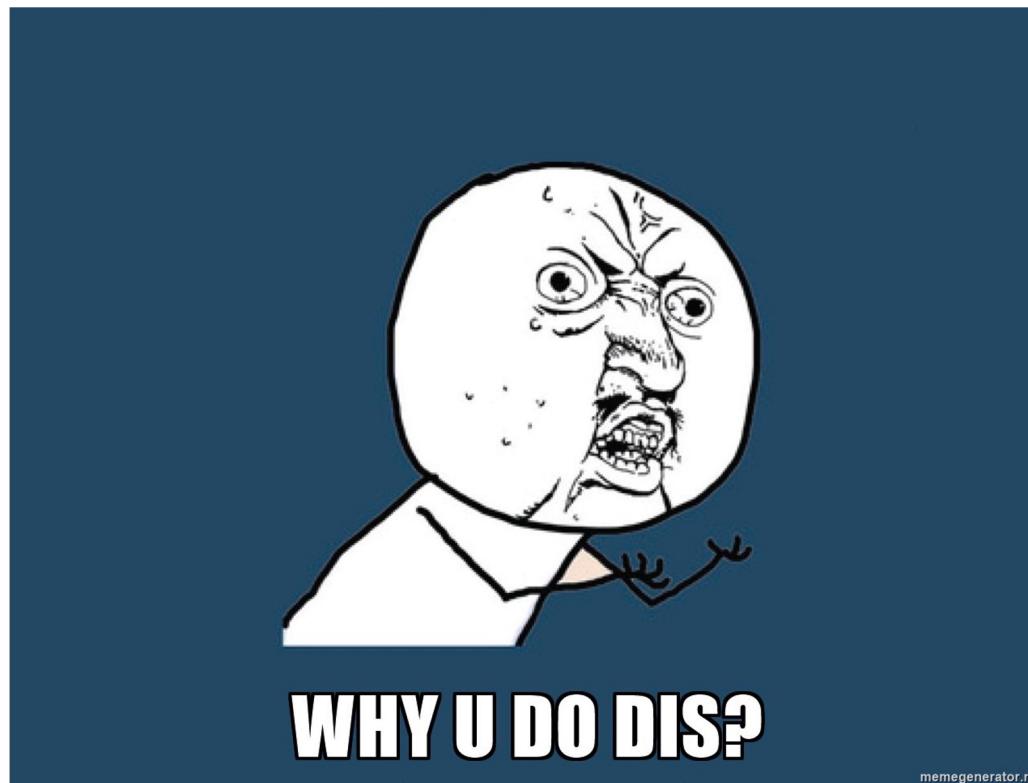


$$x_t = \bar{A}x_{t-1} + \bar{B}u_t$$

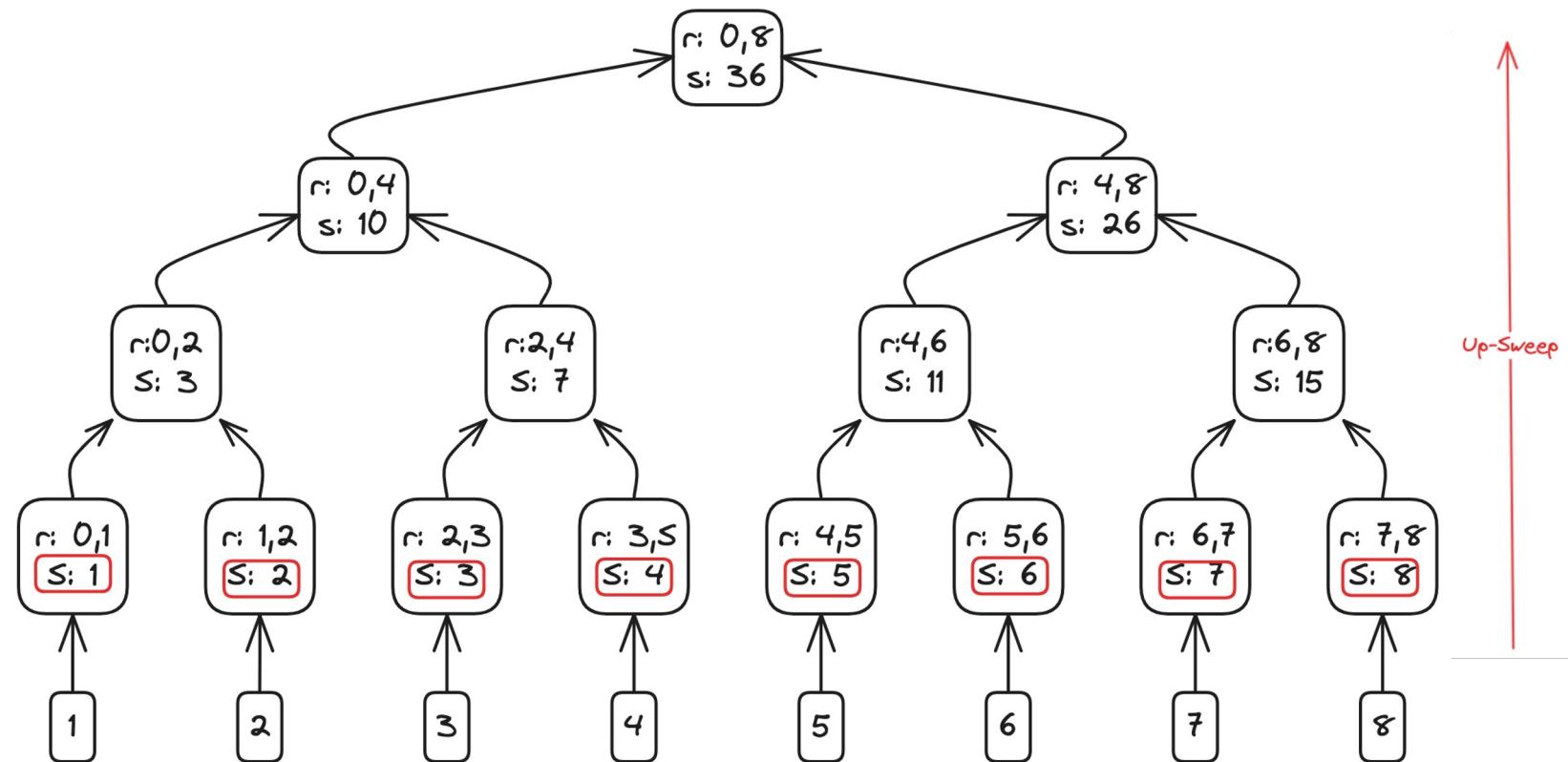
$$y_t = Cx_t$$

Current State: Sum of previous state + input

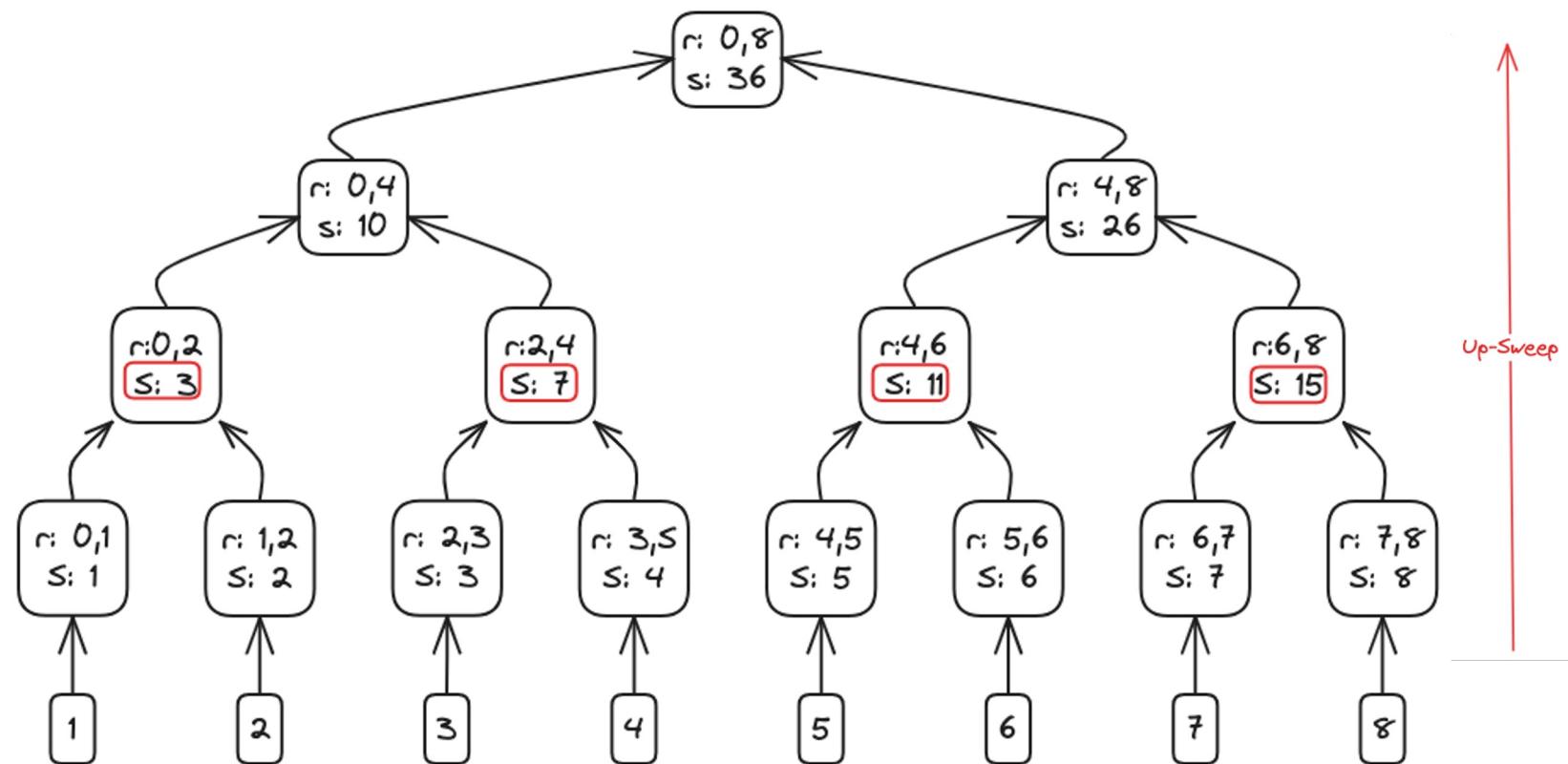
Parallelize a recurrence relation?



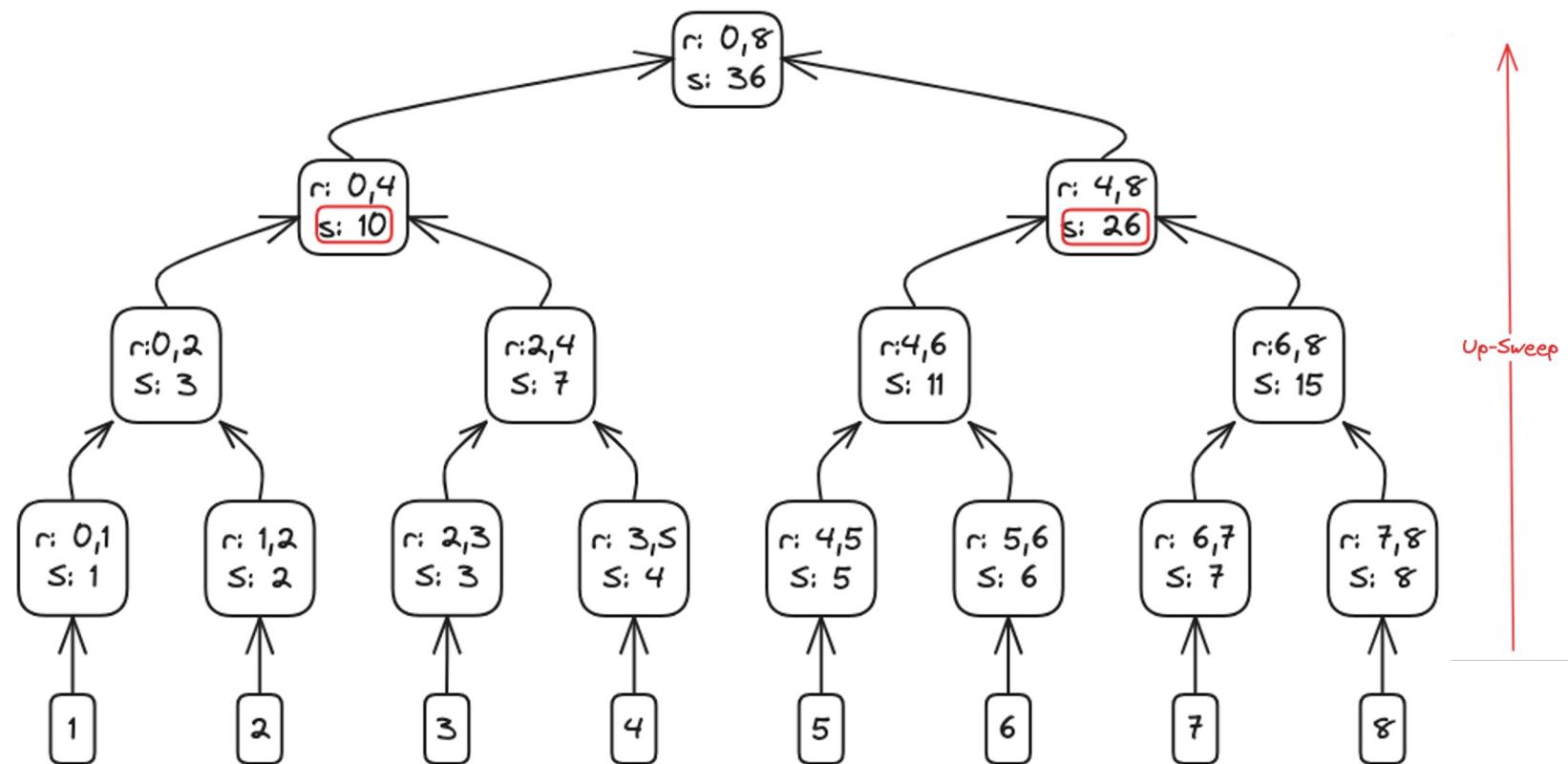
Parallel Scan: Up-Sweep



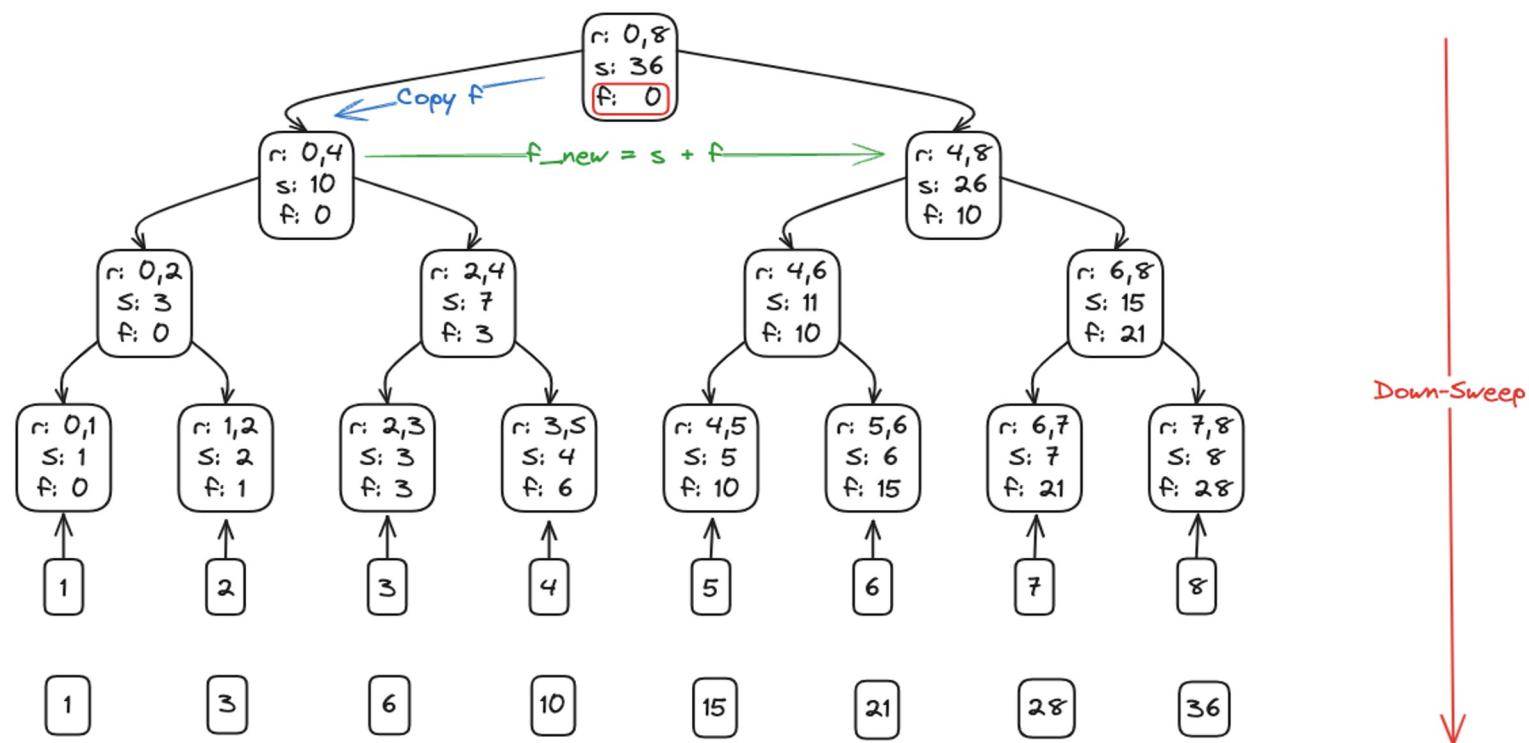
Parallel Scan: Up-Sweep



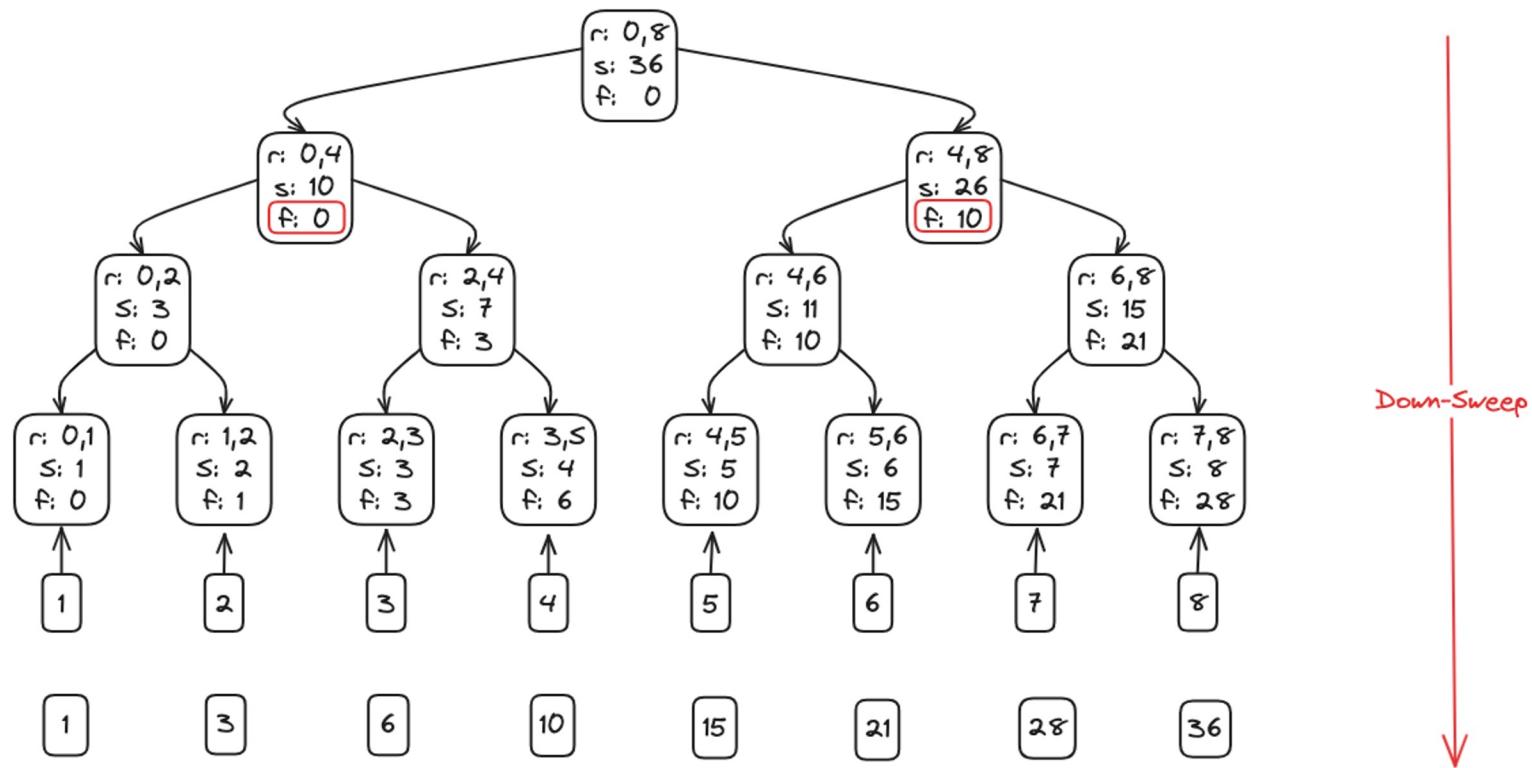
Parallel Scan: Up-Sweep



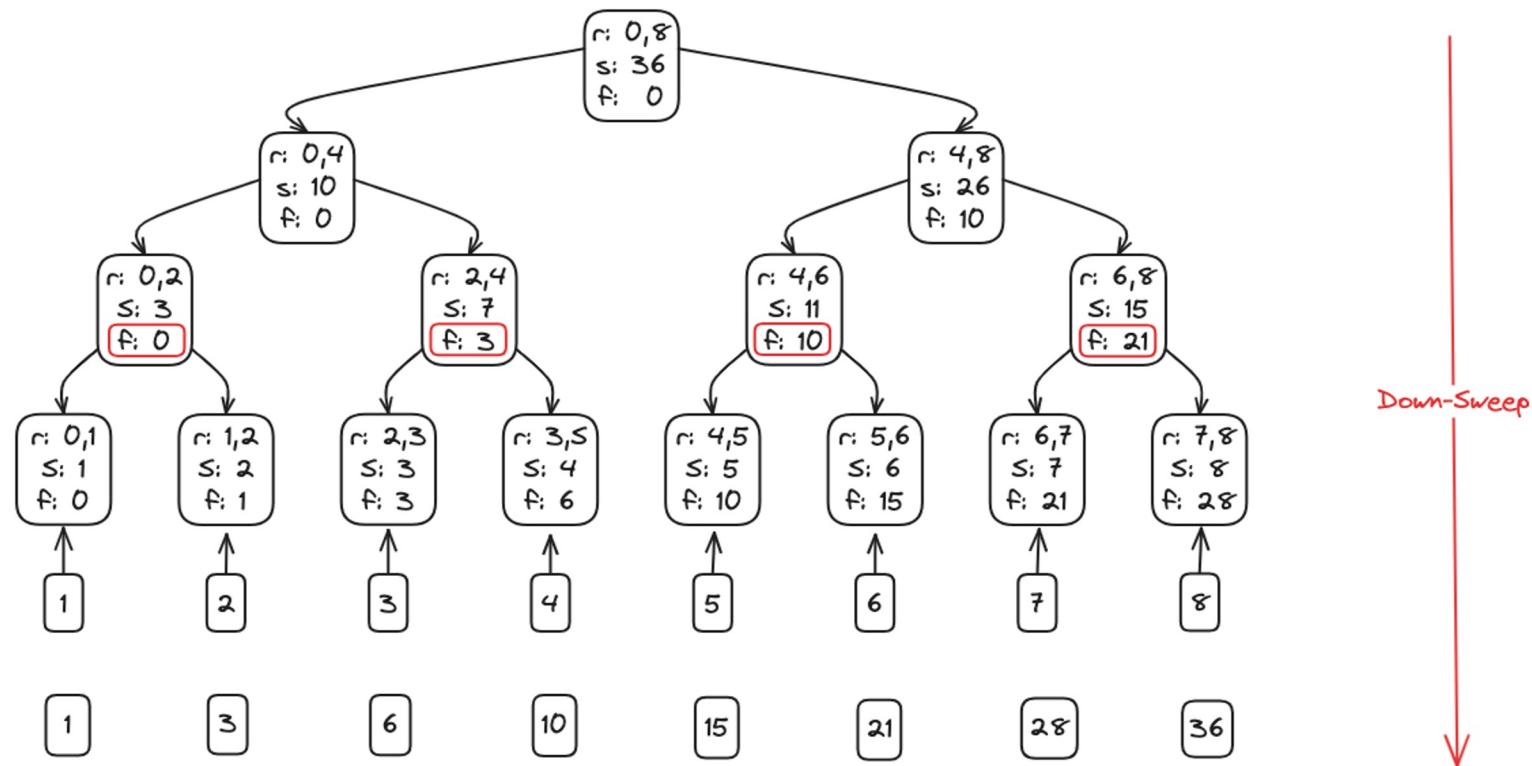
Parallel Scan: Down-Sweep



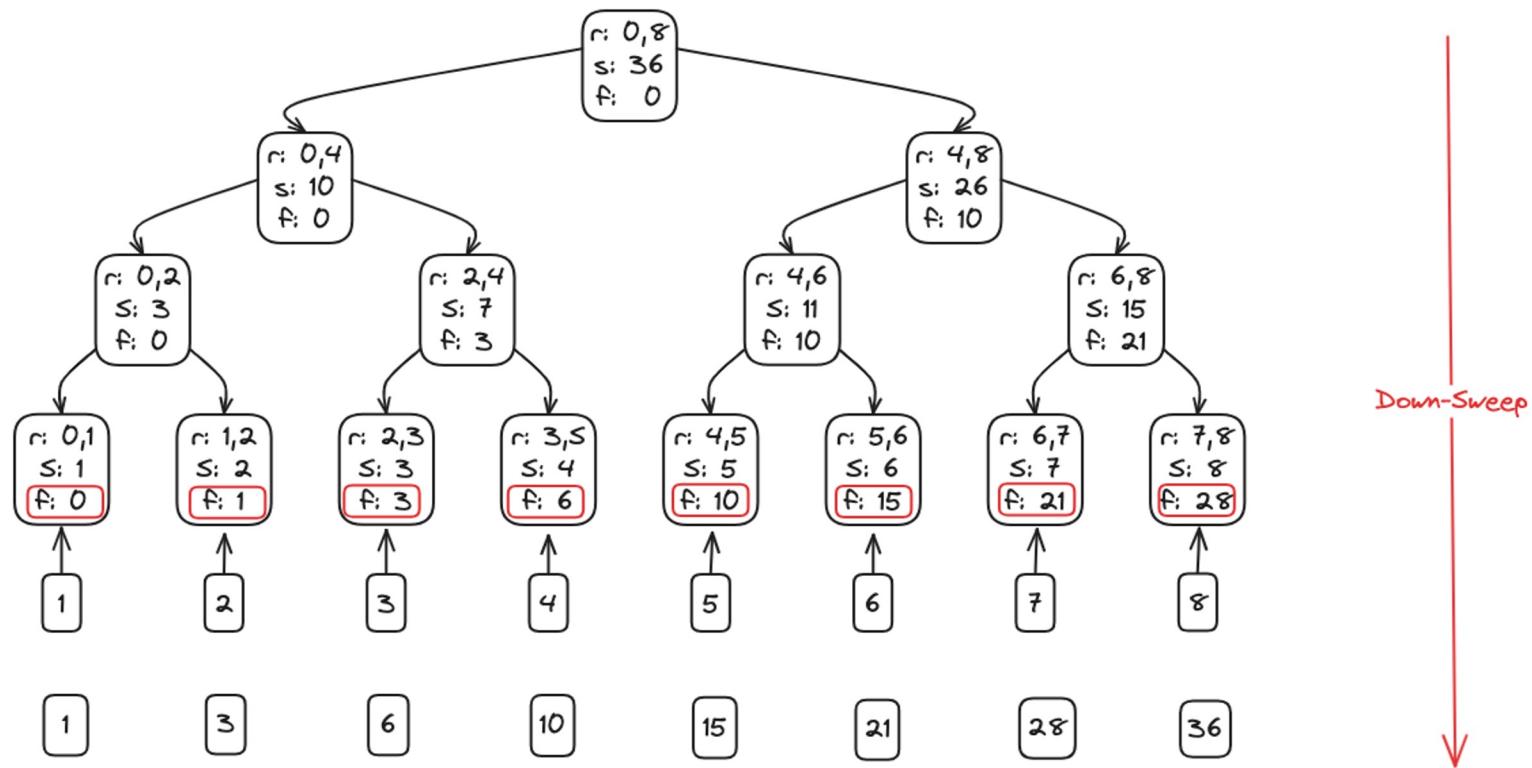
Parallel Scan: Down-Sweep



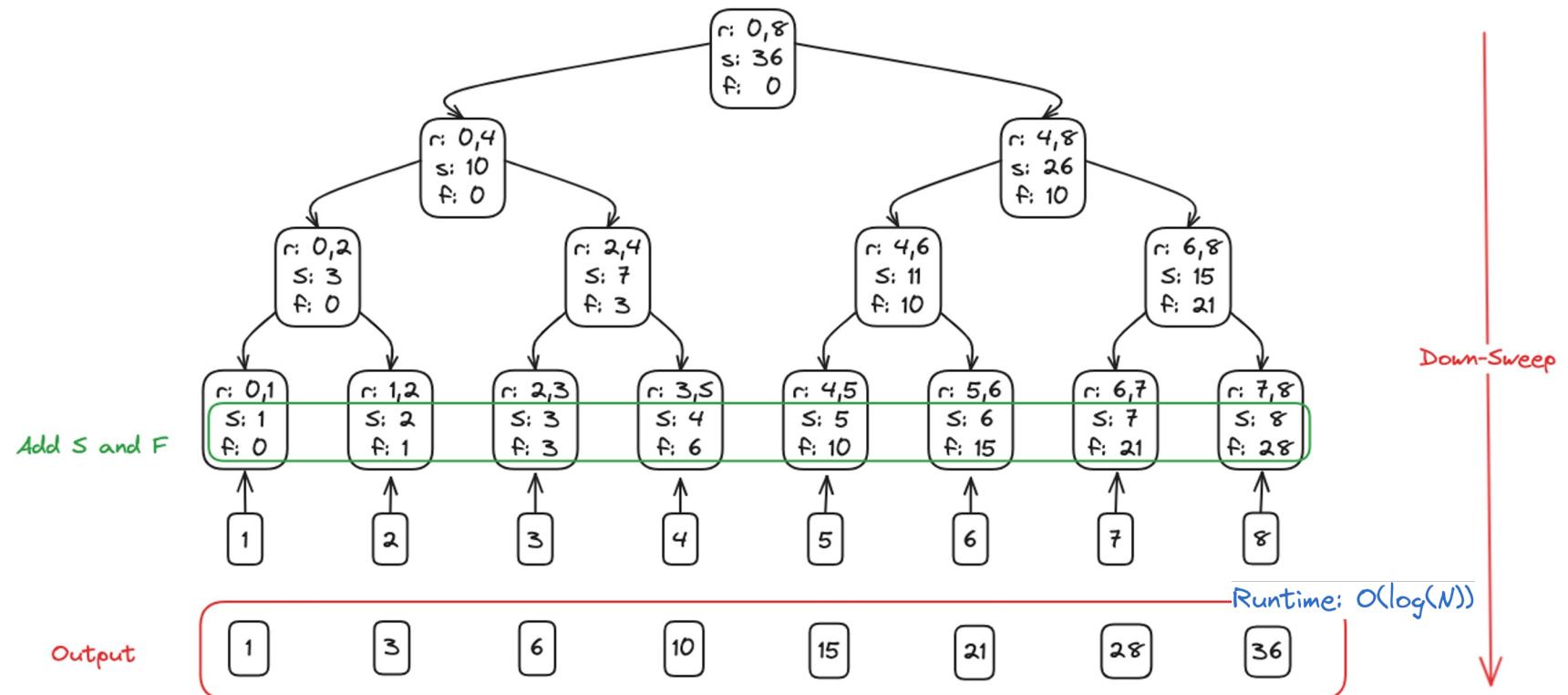
Parallel Scan: Down-Sweep



Parallel Scan: Down-Sweep



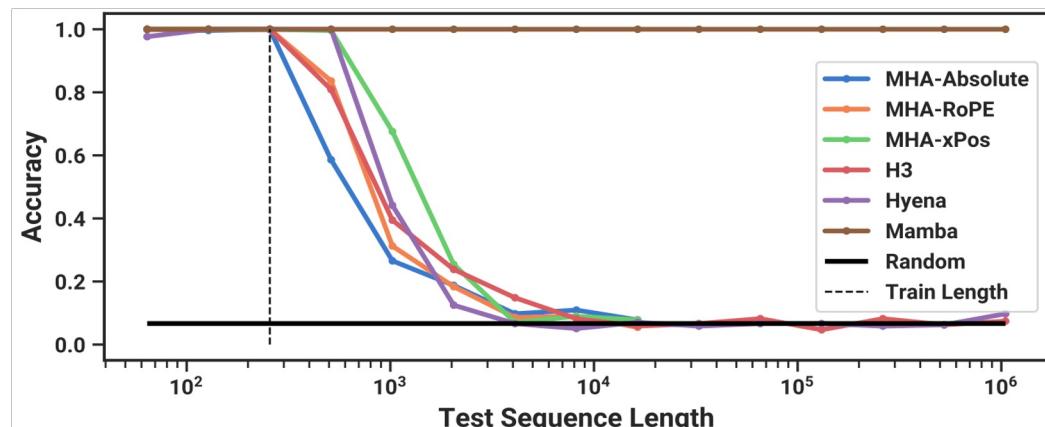
Parallel Scan: Final Result



Mamba: Synthetic Tasks Results

Model	Arch.	Layer	Acc.
S4	No gate	S4	18.3
-	No gate	S6	97.0
H3	H3	S4	57.0
Hyena	H3	Hyena	30.1
-	H3	S6	99.7
-	Mamba	S4	56.4
-	Mamba	Hyena	28.4
Mamba	Mamba	S6	99.8

Selective Copying



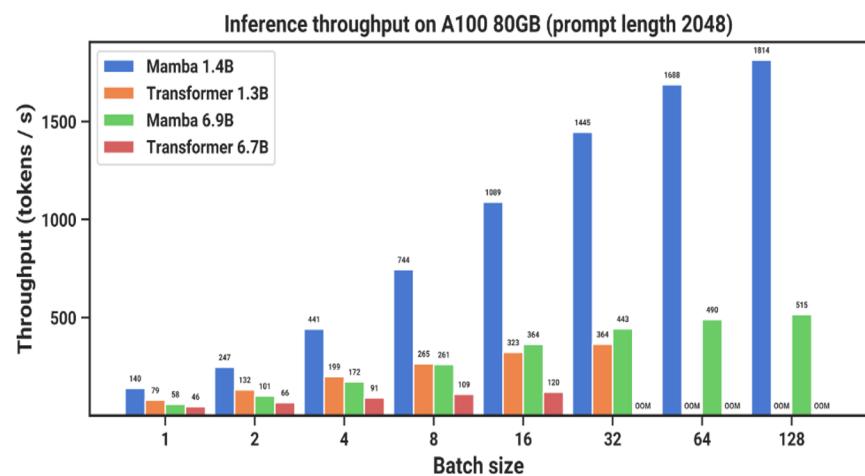
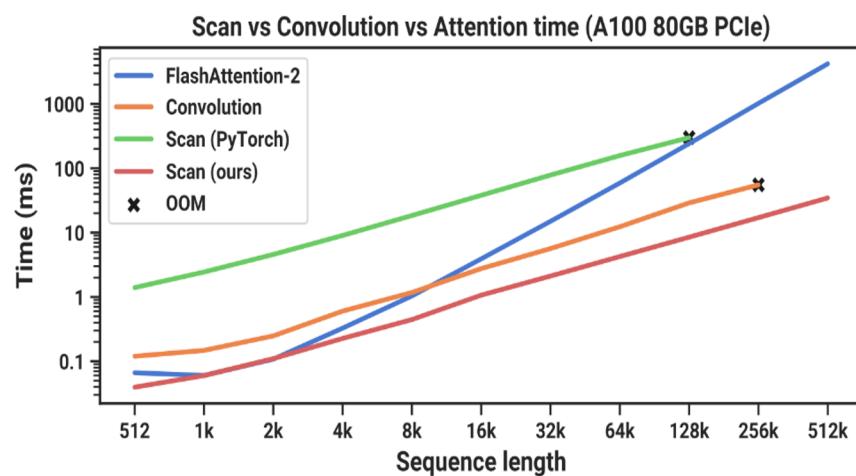
Induction Heads

Mamba: LM Results

Model	Token.	Pile ppl ↓	LAMBADA ppl ↓	LAMBADA acc ↑	HellaSwag acc ↑	PIQA acc ↑	Arc-E acc ↑	Arc-C acc ↑	WinoGrande acc ↑	Average acc ↑
GPT-Neo 1.3B	GPT2	—	7.50	57.2	48.9	71.1	56.2	25.9	54.9	52.4
Hybrid H3-1.3B	GPT2	—	11.25	49.6	52.6	71.3	59.2	28.1	56.9	53.0
OPT-1.3B	OPT	—	6.64	58.0	53.7	72.4	56.7	29.6	59.5	55.0
Pythia-1.4B	NeoX	7.51	6.08	61.7	52.1	71.0	60.5	28.5	57.2	55.2
RWKV-1.5B	NeoX	7.70	7.04	56.4	52.5	72.4	60.5	29.4	54.6	54.3
Mamba-1.4B	NeoX	6.80	5.04	64.9	59.1	74.2	65.5	32.8	61.5	59.7
GPT-Neo 2.7B	GPT2	—	5.63	62.2	55.8	72.1	61.1	30.2	57.6	56.5
Hybrid H3-2.7B	GPT2	—	7.92	55.7	59.7	73.3	65.6	32.3	61.4	58.0
OPT-2.7B	OPT	—	5.12	63.6	60.6	74.8	60.8	31.3	61.0	58.7
Pythia-2.8B	NeoX	6.73	5.04	64.7	59.3	74.0	64.1	32.9	59.7	59.1
RWKV-3B	NeoX	7.00	5.24	63.9	59.6	73.7	67.8	33.1	59.6	59.6
Mamba-2.8B	NeoX	6.22	4.23	69.2	66.1	75.2	69.7	36.3	63.5	63.3

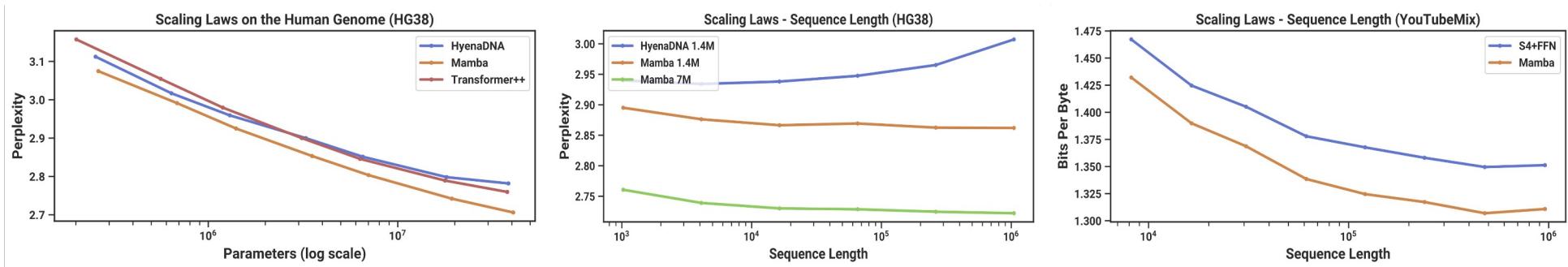
Zero-shot Eval on popular benchmarks

Mamba: Scaling Results



Source: Mamba [Albert Gu, Tri Dao 2023]

Mamba: Other Modalities



Zero-shot Eval on popular benchmarks

Mamba: Derivatives

- MambaByte
 - Vision Mamba
 - Mamba MoE
 - U-Mamba
 - Mamba-Morph
 - Swin-UMamba
 - Graph Mamba
- [J Wang et al., 2024]
[Zhu, L, et al. , 2024]
[Pióro, Maciej, 2024]
[Ma, Jun et al., 2024]
[Guo, Tao et al., 2024]
[Liu, J, et al., 2024]
[Behrouz et al., 2023]

Sasha Rush (@srush_nlp) posted on Jan 25, 2024 at 10:32 PM. The post reads:

Mamba apparently was rejected !? (openreview.net/forum?id=AL1fq...)
Honestly I don't even understand. If this gets rejected, what chance do us 😞's have.

The post includes the OpenReview.net logo and a snippet of the paper's abstract: "Mamba: Linear-Time Sequence Modeling with Selective Statistical Foundation models, now powering most of the exciting applications in deep learning, are almost universally base..."

10:32 PM · Jan 25, 2024 · 566.8K Views

Mamba rejected from ICLR 2024 😞

Conclusion

- SSMs are a promising alternative to attention.
- Inference according to input was realized by Mamba.
- Hardware-Aware algorithms are the key to scale.
- Promising other research:
 - RWKV [Peng, Bo et al., 2023]
 - Gemini 1.5 [Gemini Team, 2024]



Moar SSM and Mamba Goodness

Thank you!
