UC San Diego

Mamba State Space Model

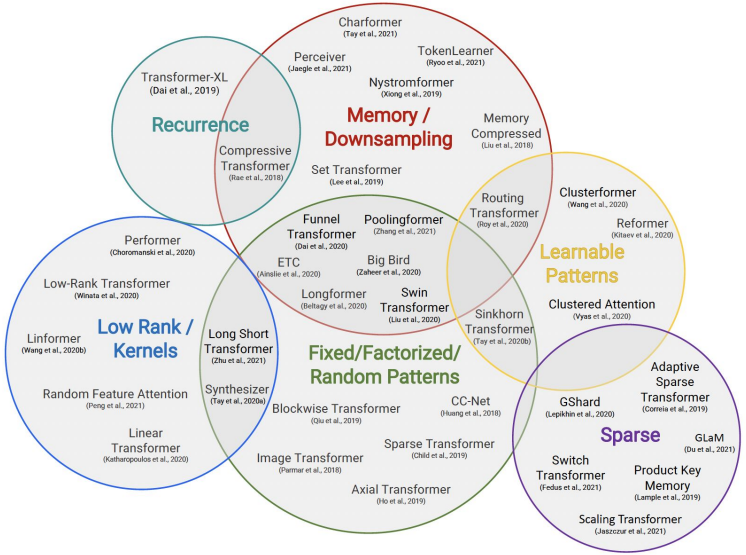
Sanidhya Singal and Aditya Gulati

CSE 291 (L00) Winter 2024: Theory of LLMs | February 20th, 2024

Contents

Motivation

- Transformers face challenges in handling long sequences.
- Subquadratic-time architectures have not matched attention-based models' performance on language processing.
- Focus on a new state space model called Mamba that tries to address this limitation.
    - Selection Mechanism and Parallel Scan based on Hardware-Aware State Expansion.
    - Fast inference and linear scaling in inference length.

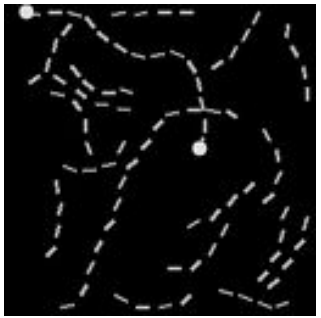


Taxonomy of Efficient Transformer Architectures (Image Source: [12])

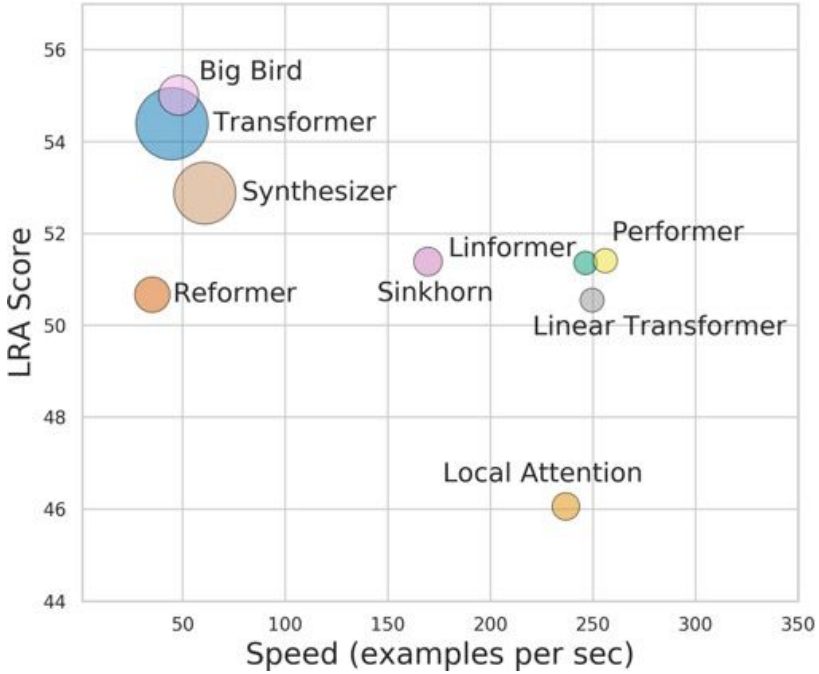
Long Range Dependencies in State Space Models

Long Range Arena (LRA) Benchmark

- Deep generative models struggle with long sequences of inputs.
- LRA Benchmark evaluates model quality under long-context scenarios.
    - Long List Ops
    - Pathfinder



Pathfinder Task



LRA Score for Various Models (Image Source: [11])

Paradigms for Long Range Dependencies

- Continuous Time Models (CTMs), Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs)
- Linear State Space Layers (LSSL)
    - Continuous-time state space representation

$$x'(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

    - Discrete-time state space representation

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

$$y_t = Cx_t + Du_t$$

Paradigms for Long Range Dependencies

$$x_0 = \overline{B}u_0 \qquad x_1 = \overline{AB}u_0 + \overline{B}u_1 \qquad x_2 = \overline{A}^2\overline{B}u_0 + \overline{AB}u_1 + \overline{B}u_2 \qquad \dots$$

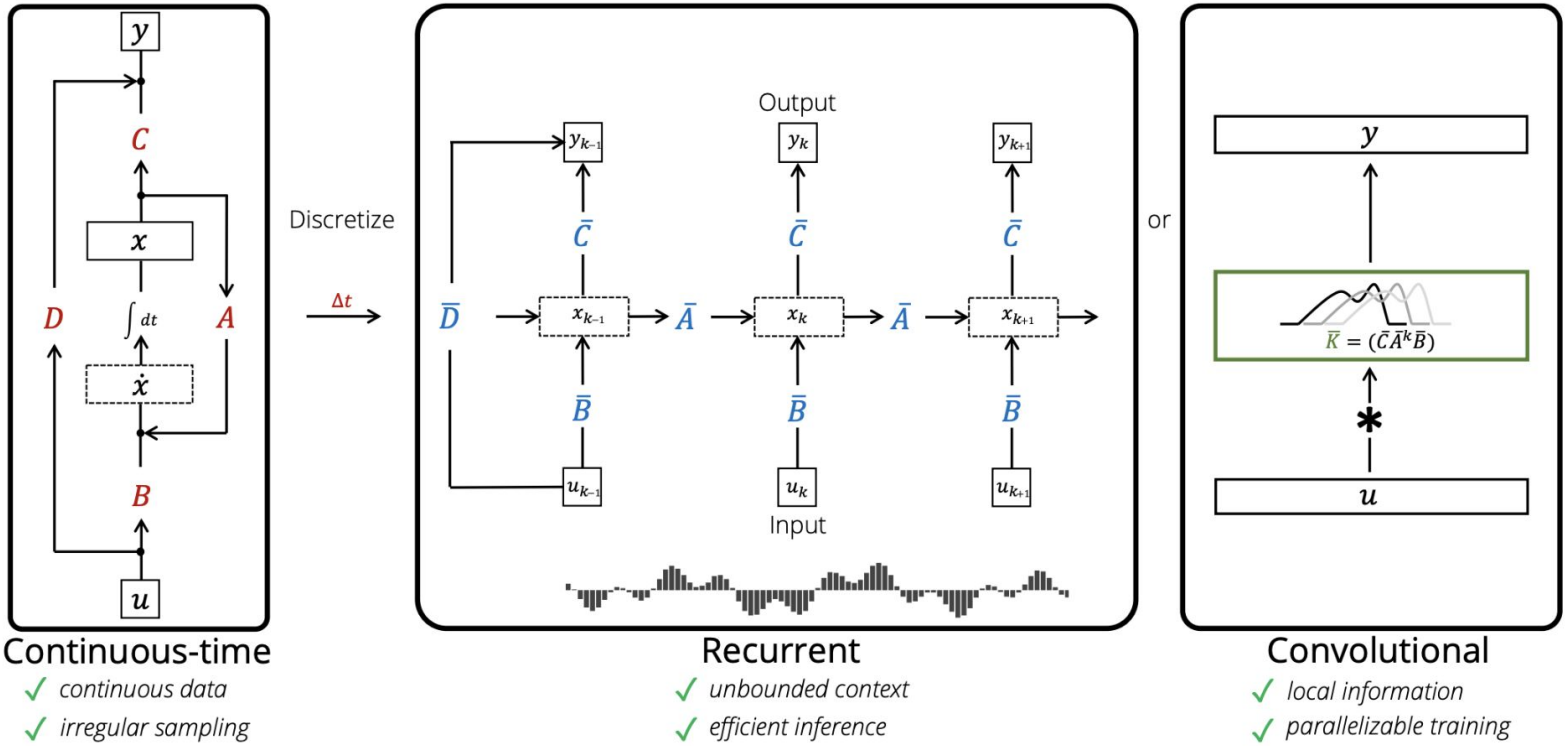
$$y_0 = \overline{CB}u_0 \qquad y_1 = \overline{CAB}u_0 + \overline{CB}u_1 \qquad y_2 = \overline{CA}^2\overline{B}u_0 + \overline{CAB}u_1 + \overline{CB}u_2 \qquad \dots$$

Unrolling the Discrete-time State Space Representation (Image Source: [5])

$$y_k = \overline{CA}^k\overline{B}u_0 + \overline{CA}^{k-1}\overline{B}u_1 + \cdots + \overline{CAB}u_{k-1} + \overline{CB}u_k$$
$$y = \overline{K} * u.$$

$$\overline{K} \in \mathbb{R}^L := \mathcal{K}_L(\overline{A}, \overline{B}, \overline{C}) := \left(\overline{CA}^i\overline{B}\right)_{i\in[L]} = (\overline{CB}, \overline{CAB}, \dots, \overline{CA}^{L-1}\overline{B}).$$

Vectorizing the Unrolled Discrete-time State Space Representation into a Convolution (Image Source: [5])



(Left) As an implicit continuous model, irregularly-spaced data can be handled by discretizing the same matrix A using a different timescale Δt. (Center) As a recurrent model, inference can be performed efficiently by computing the layer timewise, i.e., one vertical slice at a time by unrolling the linear recurrence. (Right) As a convolutional model, training can be performed efficiently by computing the layer depthwise in parallel by convolving with a particular filter (Image Source: [6]).

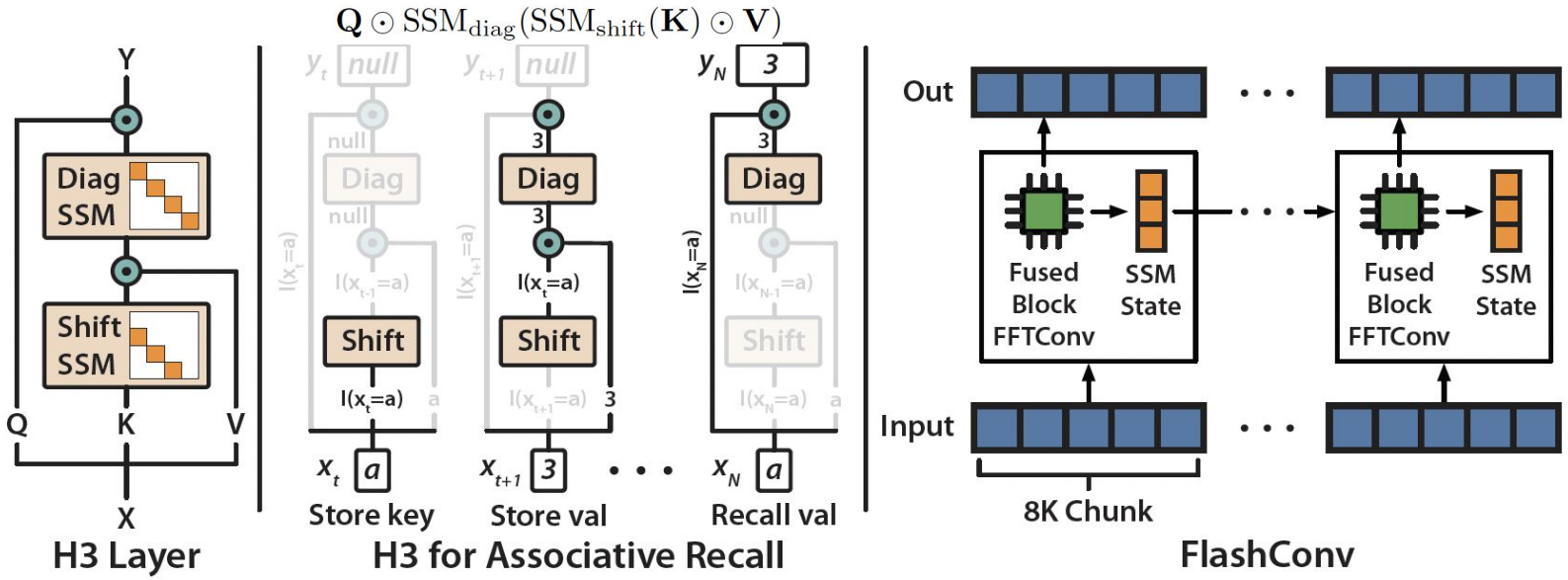
Structured State Spaces (S4)

- Challenges with high powers of A: Vanishing gradients; $O(N^2L)$ computations.
- Use HiPPO operators: Matrix A can be diagonalized stably; $O(N+L)$ computations for high powers of A

Model	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Reformer	37.27	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	65.90	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	77.80	✗	54.42
Nyströmformer	37.15	65.52	79.56	41.58	70.94	✗	57.46
Luna-256	37.25	64.57	79.29	47.38	77.72	✗	59.37
S4	**59.60**	**86.82**	**90.90**	**88.65**	**94.20**	**96.35**	**86.09**

Comparative results for the LRA benchmark (Image Source: [5])

Hungry Hungry Hippos (H3)



(Left) H3 stacks two discrete SSMs with shift and diagonal matrices and uses multiplicative interactions between input projections and their outputs to model comparisons between points in a sequence. (Middle) H3 can perform associative recall, which is easy for attention, but not existing SSMs. (Right) FlashConv uses a new state-passing algorithm over fused block FFTConv to increase hardware efficiency of SSMs, allowing H3 to scale to billion-parameter models (Image Source: [2]).

Hyena Hierarchy

- Key Contributions:
  - Local Convolution (Shift-SSM) similar to H3
  - MLP-parameterized Global Convolution
  - Gated Recurrence
- Captures Long-range Dependencies
- Scalable and Efficient
- Potential for Interpretability

Mamba State Space Model

Key Contributions

1. **Selection Mechanism**
   - Similar to attention
   - Parameterize model parameters based on the input sequence
2. **Hardware-aware Algorithm**
   - Unable to use convolutional trick from S4
   - Uses parallel scan in combination with efficient use of memory
3. **Simplified SSM Architecture**
   - Combines the design of SSM architectures with the MLP block of Transformers into a single block

Selection Mechanism

Algorithm 1 SSM (S4)

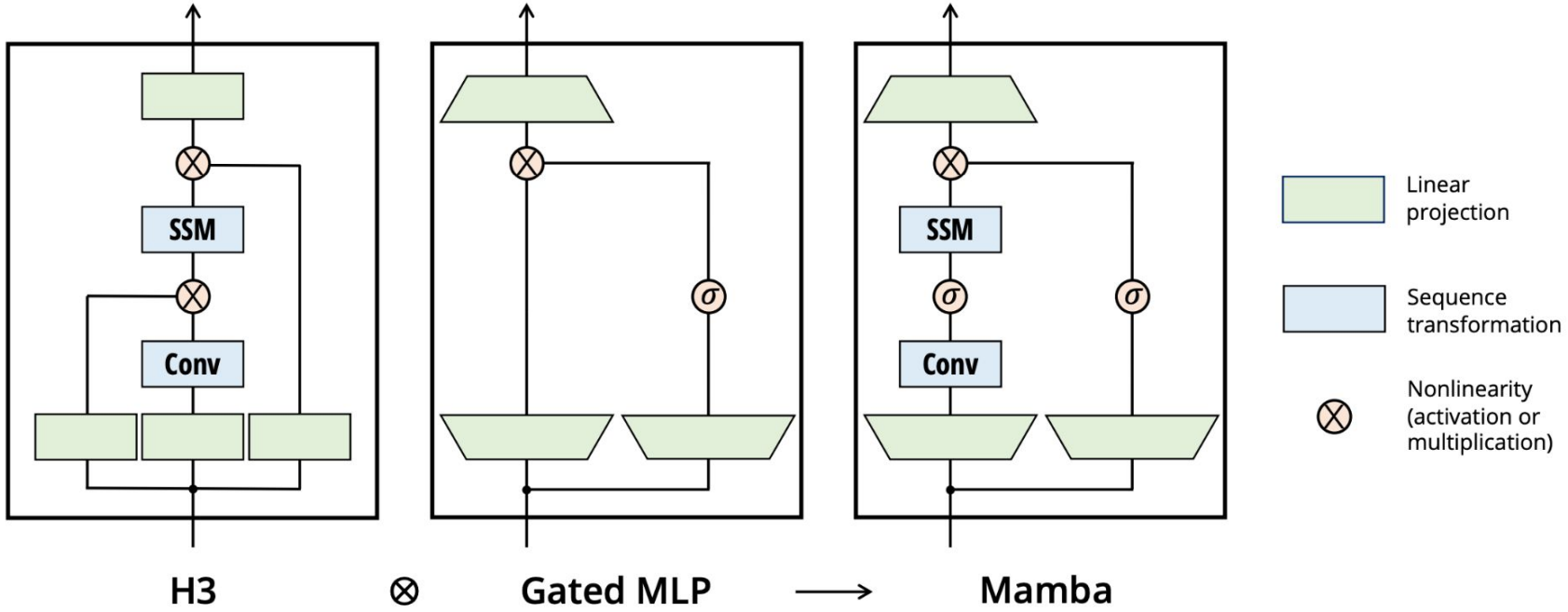
Input: $x : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
Output: $y : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
1: $\boldsymbol{A} : (\mathtt{D}, \mathtt{N}) \leftarrow$ Parameter
▷ Represents structured $N \times N$ matrix
2: $\boldsymbol{B} : (\mathtt{D}, \mathtt{N}) \leftarrow$ Parameter
3: $\boldsymbol{C} :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

Algorithm 2 SSM + Selection (S6)

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

Side-by-side comparison of SSM (S4) and SSM + Selection (S6) Algorithms (Image Source: [3])

Simplified SSM Architecture



Mamba Architecture based on the H3 Block and the Gated MLP Block (Image Source: [3])

Properties of SSMs

$$h'(t) = Ah(t) + Bx(t) \quad (1a)$$
$$y(t) = Ch(t) \quad (1b)$$

Recurrent Form

$$h_t = \overline{A}h_{t-1} + \overline{B}x_t \quad (2a)$$
$$y_t = Ch_t \quad (2b)$$

$$\overline{K} = (C\overline{B}, C\overline{AB}, \dots, C\overline{A}^k\overline{B}, \dots) \quad (3a)$$
$$y = x * \overline{K} \quad (3b)$$

Convolution Form

$$\overline{A} = \exp(\Delta A) \qquad \overline{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B$$

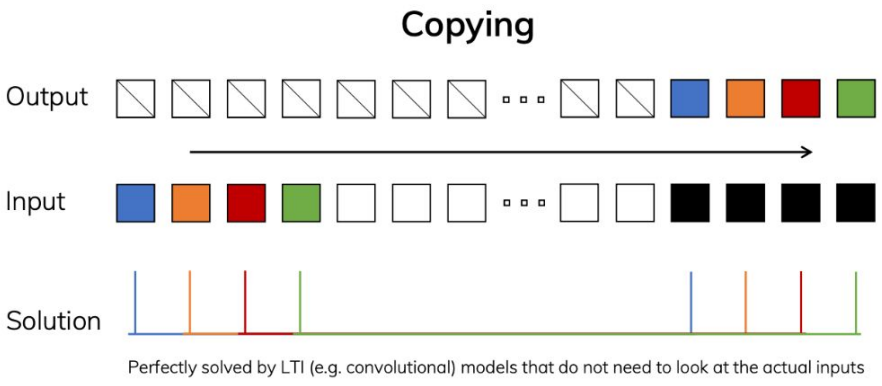
Discretized Form

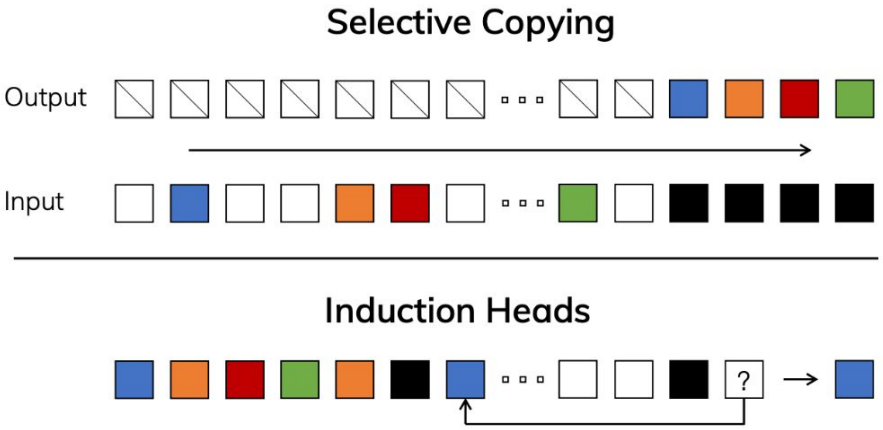
Under certain constraints, SSMs -> Gating Mechanism!

Theorem 1. *When* $N = 1, A = -1, B = 1, s_\Delta = \text{Linear}(x)$, *and* $\tau_\Delta = \text{softplus}$, *then the selective SSM recurrence (Algorithm 2) takes the form*

$$g_t = \sigma(\text{Linear}(x_t))$$
$$h_t = (1 - g_t)h_{t-1} + g_t x_t. \quad (5)$$

Toy Task Definition





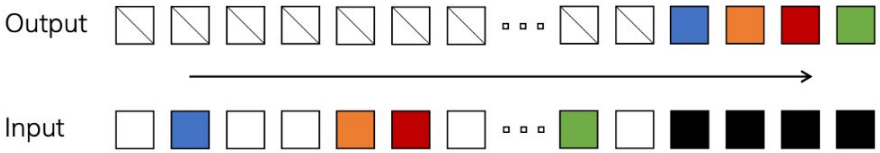
Selective Copying = Context Filtering = requires “data-dependence”

Induction Heads = Associative Recall = In-context Learning

Interpretation of SSMs

$$\overline{A} = \exp(\Delta A) \qquad \overline{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B$$

Selective Copying



1.Variable Spacing = Filtering irrelevant tokens = gt->0

Boundary Resetting

2. Independent Sequences = LTI Bleed information = SSMs Reset by gt->1 or Δ_t -> ∞

$$g_t = \sigma(\text{Linear}(x_t))$$
$$h_t = (1 - g_t)h_{t-1} + g_t x_t.$$

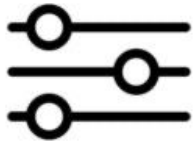
Interpretation of Δ

Focus on the past,
Ignore the current

Ignore the past,
Focus on current

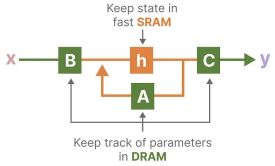
3. Δ balances between focusing or ignoring the current input!

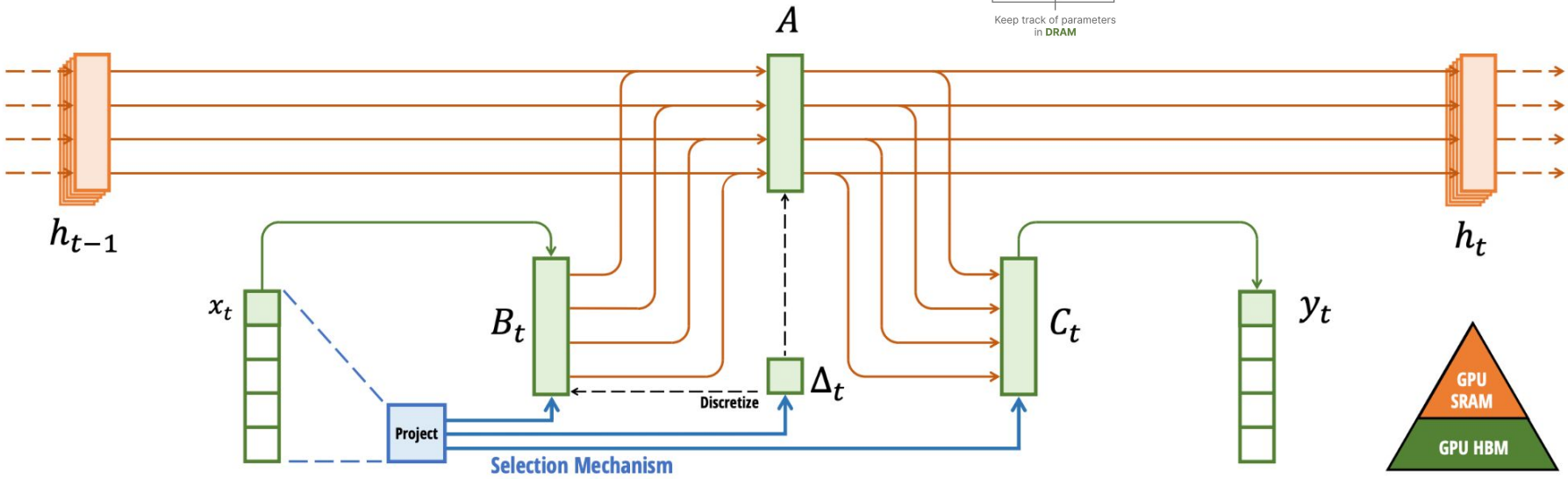
Filtering Context



4. LTI Models -> Can't effectively ignore irrelevant context
SSMs -> Reset state to remove extraneous history

Hardware-aware Algorithm





Selective State Space Model with Hardware-Aware State Expansion (Image Source: [3])

1. Read (Δ, A, B, C) from slow HBM to fast SRAM.
2. Discretize to produce A, B of size (B, L, D, N) in SRAM.
3. Yield intermediate states in SRAM.
4. Multiply and Sum with C, produce and write output to HBM.

Additionally use Kernel Fusion and Recompution.

1. Instead of preparing the scan input $(\bar{A}, \bar{B})$ in GPU HBM,
2. Load the SSM parameters A, B, C, Δ directly from the slow HBM to fast SRAM, and perform the discretization and recurrence.
3. Then, write the final results of size $B \times L \times D$ back to the HBM.

Results and Inferences

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**

Table 1: (**Selective Copying.**) Accuracy for combinations of architectures and inner sequence layers.

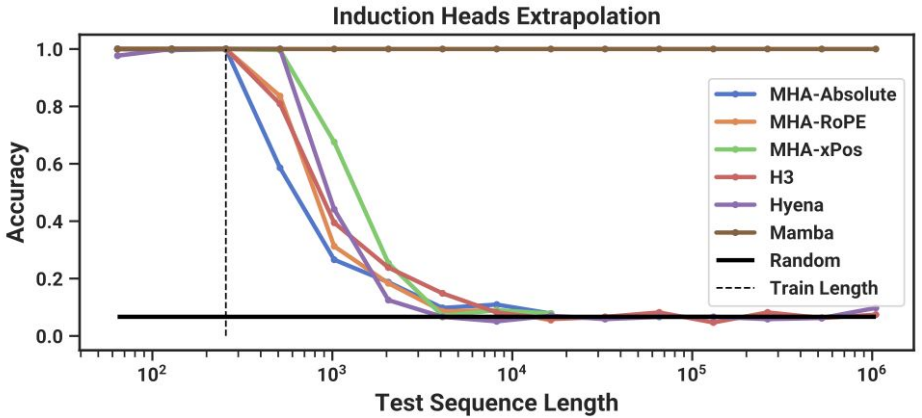
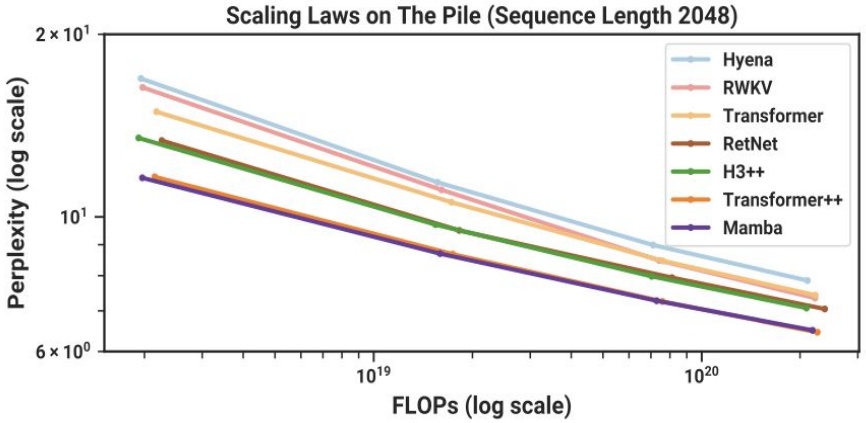
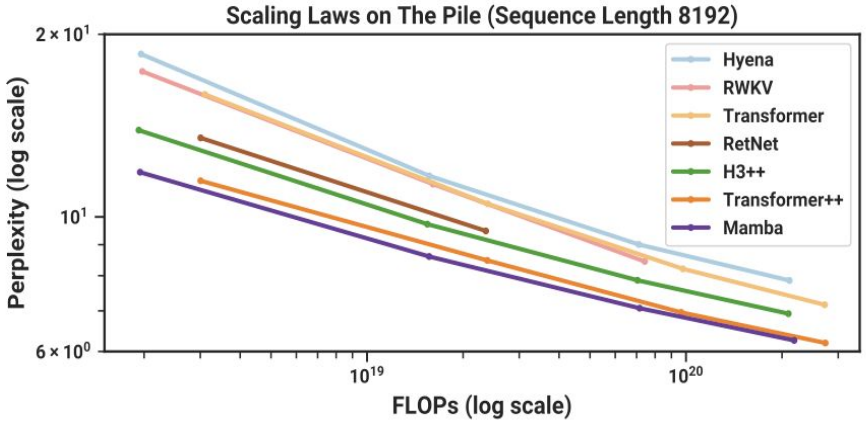


Table 2: (**Induction Heads.**) Models are trained on sequence length $2^8 = 256$, and tested on increasing sequence lengths of $2^6 = 64$ up to $2^{20} = 1048576$. Full numbers in Table 11.

Mamba Performance on Vanilla Tasks (Image Source: [3])

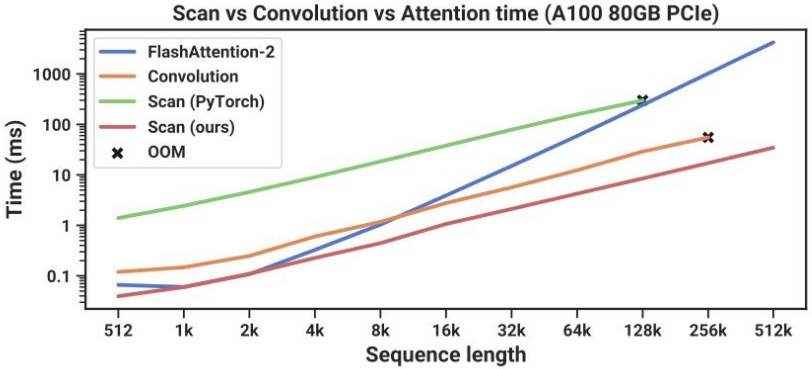
Results and Inferences



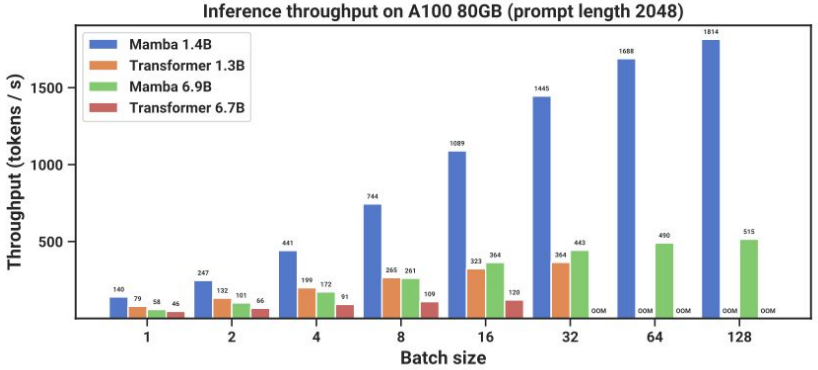


Model Size = 125M -> 1.3B; Mamba scales better than attention-free models and matches Transformer++

Results and Inferences



Efficient scan is 40x Faster!



Mamba achieves 5x higher Throughput!

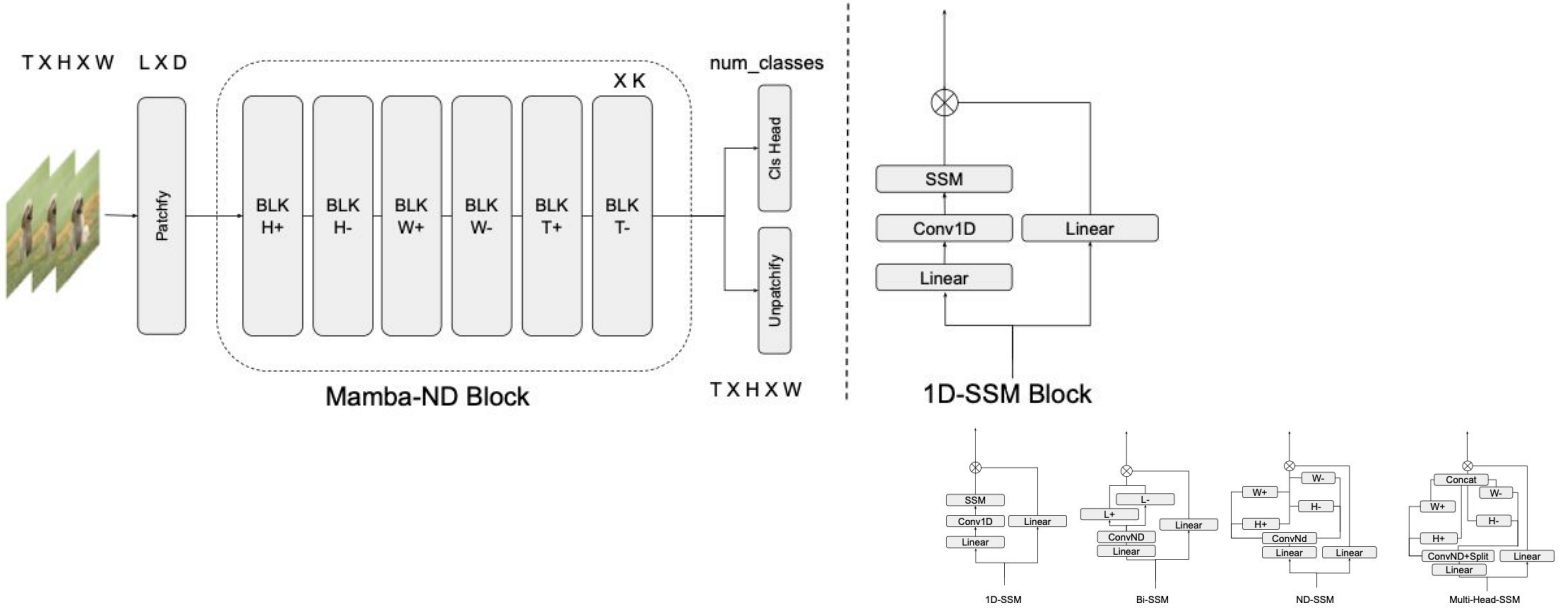
Results and Inferences

- DNA Modeling: Mamba's pretraining perplexity improves smoothly with model size, and that Mamba scales better than both HyenaDNA and Transformer++.
- Ablation Studies:
  - Architecture: The Mamba block performs similarly to H3 while being simpler.
  - Selective SSM: Δ is the most important parameter due to its connection to RNN gating.

Extensions of Mamba

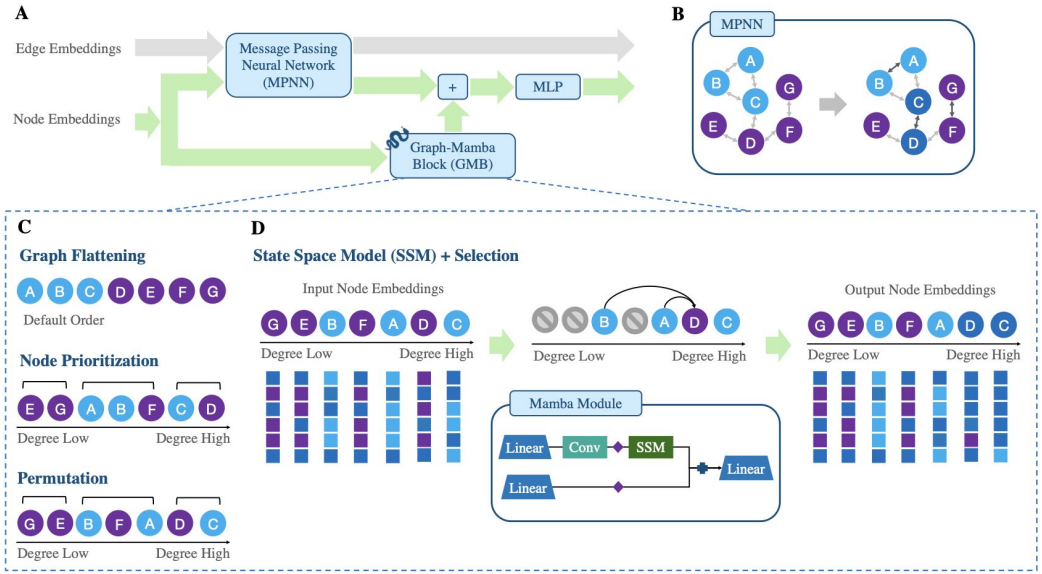
Mamba-ND (12 days old)

- Selective State Space Modeling for Multi-Dimensional Data (Image, Video)



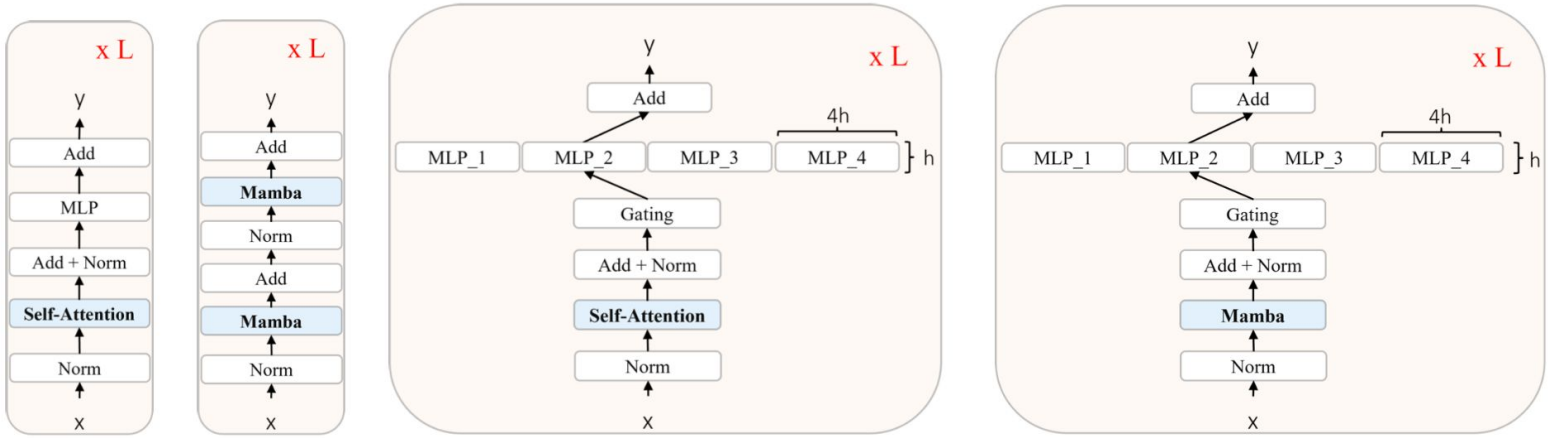
Graph Mamba (20 days old)

- Extending Mamba for long-range dependencies in Graph nodes



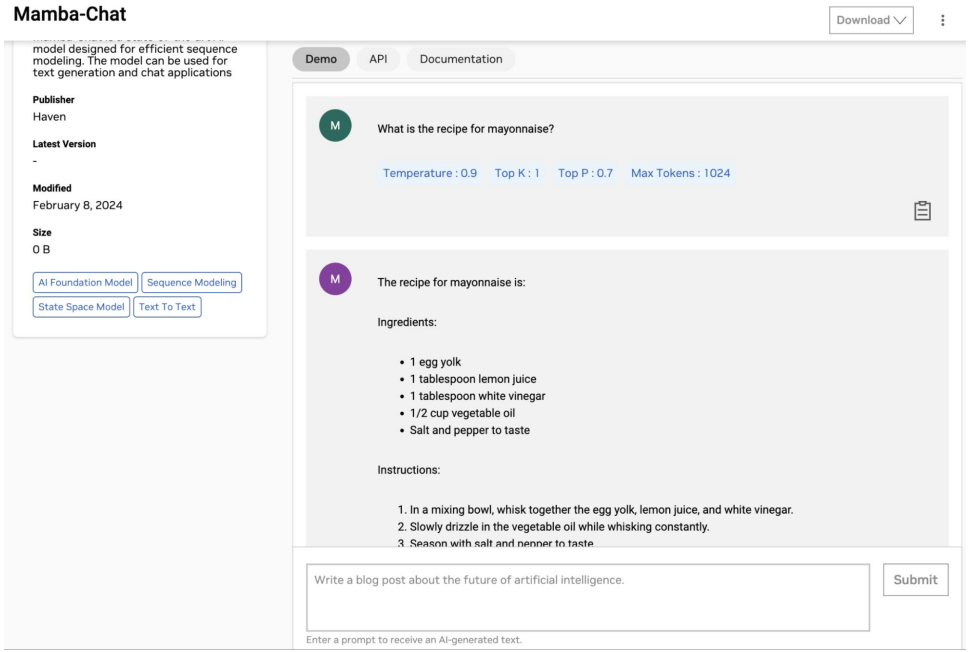
Black Mamba (20 days old)

- Modeling Mixture-of-Experts equivalent models for Mamba



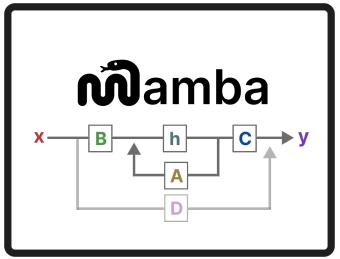
Mamba Chat (12 days old)

- Instruction Tuned Chat Model as NVIDIA AI Foundation Models and Endpoints



Future Work

- Scale Mamba models to an extent of the Transformer models
- Extend Mamba models to:
    - Multimodal (Images/Graphs)
    - Mixture-of-Experts
    - Instruction Fine Tuning



- Perform MLSys-centric research on Mamba SSMs:
    - Prompting
    - In-Context Learning
    - RLHF
    - Quantization
    - Distillation

References

[1] Anthony, Q., Tokpanov, Y., Glorioso, P., and Millidge, B. Blackmamba: Mixture of experts for state-space models, 2024.

[2] Fu, D. Y., Dao, T., Saab, K. K., Thomas, A.W., Rudra, A., and Ré, C. Hungry hungry hippos: Towards language modeling with state space models, 2023.

[3] Gu, A., and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

[4] Gu, A., Dao, T., Ermon, S., Rudra, A., and Re, C. Hippo: Recurrent memory with optimal polynomial projections, 2020.

[5] Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces, 2022.

[6] Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., and Ré, C. Combining recurrent, convolutional, and continuous-time models with linear state-space layers, 2021.

[7] Li, S., Singh, H., and Grover, A. Mamba-nd: Selective state space modeling for multi-dimensional data, 2024.

[8] Patel, C. Performance-Efficient Mamba-Chat from NVIDIA AI Foundation Models. [Online; Feb 12, 2024].

[9] Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models, 2023.

[10] Smith, J. T. H., Warrington, A., and Linderman, S. W. Simplified state space layers for sequence modeling, 2023.

[11] Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. Long range arena: A benchmark for efficient transformers, 2020.

[12] Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. Efficient transformers: A survey, 2022.

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023.

[14] Voelker, A., Kajić, I., and Eliasmith, C. Legendre memory units: Continuous-time representation in recurrent neural networks. In Advances in Neural Information Processing Systems (2019), H.Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc.

[15] Wang, C., Tsepa, O., Ma, J., and Wang, B. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces, 2024.


UC San Diego
Thank You!
Questions?