State Space Models: An Efficient Alternative to Attention

MASTER'S THESIS

Supervised by Prof. Pranabendu Misra

Brief History* of Language Modelling

- Feed Forward Networks:
  - Use N^{th} Markov Assumption
    - You need last N tokens to predict $N+1^{th}$ token
  - FFNs have fixed length input
  - Computationally infeasible to do more

- Transformers:
  - Introduced in 2017 by Vaswani et al.
    - Attention is All You Need
  - Fully based on Attention Mechanism
    - Parallelizable and Scalable
    - Stable Training
  - Sets new standard for Language Modelling
    - All SoT models are transformer based
      - SuperGLUE Benchmark (Wang, 2019)

- RNNs:
  - Does not require Markov Assumption
    - Allows to condition on arbitrarily long sequences
  - Became popular after BPTT
  - Expressive
    - Proved to be Turing Complete (Siegelmann & Sontag, 1991)
  - Issues:
    - Optimization: Vanishing/Exploding Gradients
    - Scalability: Sequential (training can't be parallelized)
  - In 2014, RNNs shown to work at scale
    - Via LSTM (Graves)
    - Issues partially alleviated by techniques like Gradient Clipping, Gating etc.
  - LSTM becomes de facto for Seq2Seq

Is Attention All You Need?

Current Status: Yes

Time Remaining: 952d 14h 39m 46s

Proposition:

On January 1, 2027, a Transformer-like model will continue to hold the state-of-the-art position in most benchmarked tasks in natural language processing.

For the Motion

Jonathan Frankle
@jefrankle
Harvard Professor
Chief Scientist Mosaic ML

Against the Motion

Sasha Rush
@srush_nlp
Cornell Professor
Research Scientist Hugging Face 🤗

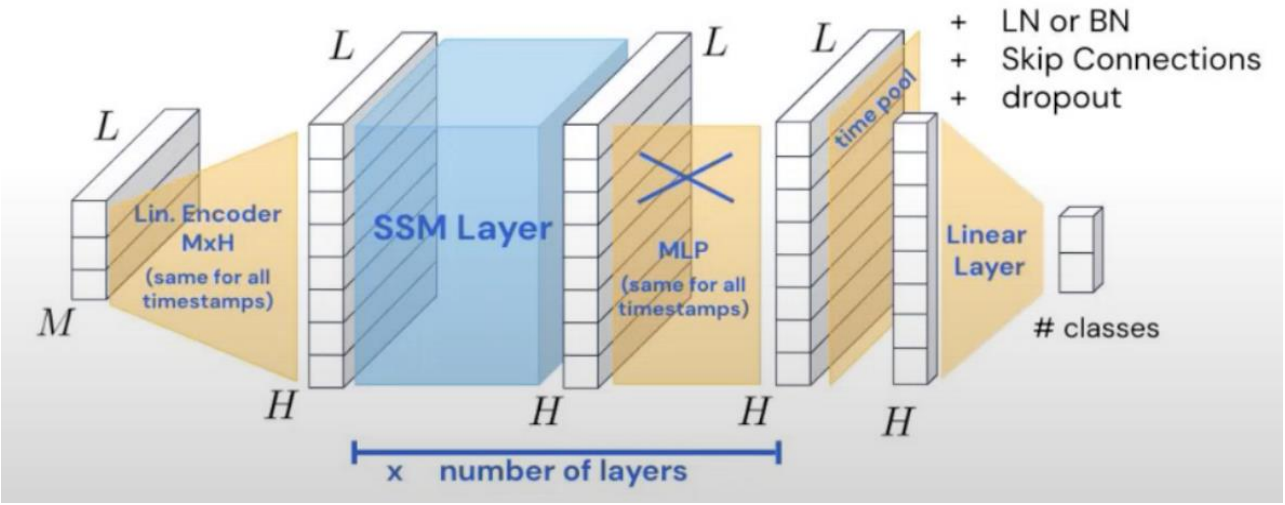
Wager

The wager is for donation of equity in Mosaic ML or Hugging Face to a charity of the winner's choice. Details to come.

www.isattentionallyouneed.com

Maybe Attention is NOT All You Need!

- State Space Models (SSM):
  - They are essentially RNNs trying to fix 2 problems:
    - Stable Training
    - Scalable Training
- Start with a core architecture for Seq2Seq Modelling
  - Work within transformer backbone
    - Replace MHA with SSM layer
    - Sometimes modify the Blocks
  - Pure SSM Models:
    - S4
    - Mamba
    - LRU
  - Hybrid:
    - Hawk and Griffin

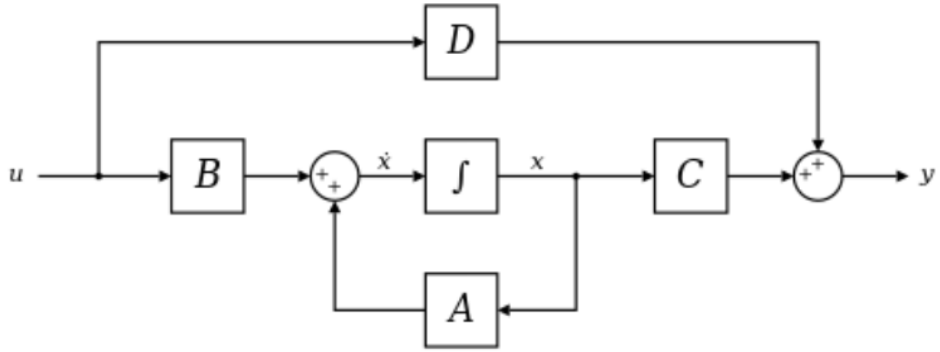


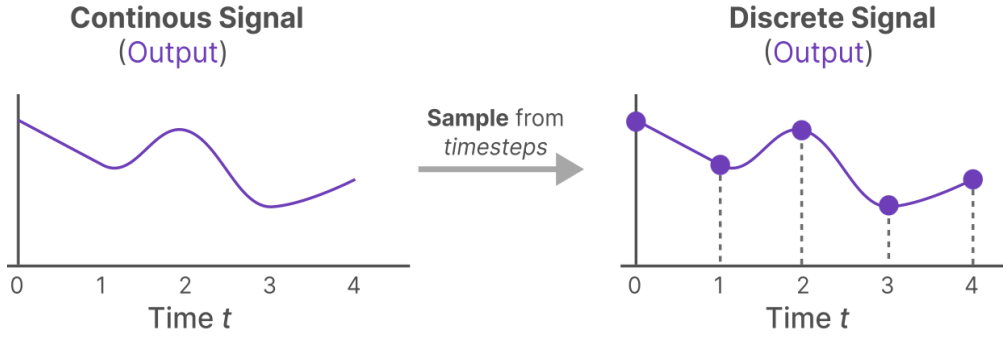
State Space Models (SSM)

- SSM Equations (LTI): $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$

  $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$

  - Introduced in Control Theory
  - Define in terms of ODE:
    - They assume continuous time
  - 'A' matrix is the most important:
    - Also called the **State Matrix**
    - We mostly use **deterministic** initializations for A
      - Different initializations for different models: HiPPO, S4D-Real/Complex etc.
    - Used to control the dynamics of the state vector.
  - Need to discretize the system to train ML Models
    - Discretization converts this into a difference/**recurrent** equation
    - Different discretization techniques used by different architectures
      - Euler's Method, Bilinear Transform, Zero-Order Hold etc.

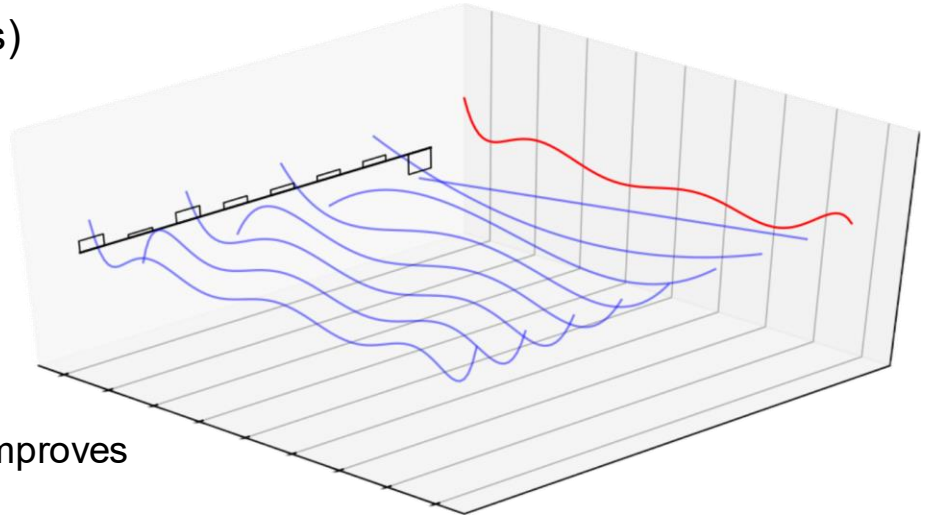




https://www.maartengrootendorst.com/blog/mamba/

Structured SSM (S4)

- By Gu et al., 2022
  - Promises best of both Transformers and RNNs
    - Fast parallelizable training via Convolutions (like Transformers)
    - High throughput during inference (like RNNs)
- Initialization of A:
  - HiPPO Matrix: $$\mathbf{A}_{nk} = -\begin{cases} (2n+1)^{\frac{1}{2}}(2k+1)^{\frac{1}{2}} & \text{if } n > k \\ n+1 & \text{if } n = k \\ 0 & \text{if } n < k \end{cases}$$
    - Works like magic!
      - Prior work: Simply modifying `A` from random matrix to HiPPO improves seq. MNIST classification accuracy from 60% to 98%



- Discretization:
  - Bilinear Transform: $$\mathbf{x}(t+\Delta) \simeq \left(\mathbf{I} - \frac{\Delta}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} + \frac{\Delta}{2}\mathbf{A}\right)\mathbf{x}(t) + \left(\mathbf{I} - \frac{\Delta}{2}\mathbf{A}\right)^{-1}\Delta\mathbf{B}\mathbf{u}(t)$$
    - Converts SSM ODE to a recurrence via:

$$x_k = \overline{A}x_{k-1} + \overline{B}u_k \qquad \overline{A} = (I - \Delta/2 \cdot A)^{-1}(I + \Delta/2 \cdot A)$$
$$y_k = \overline{C}x_k \qquad \overline{B} = (I - \Delta/2 \cdot A)^{-1}\Delta B \qquad \overline{C} = C$$

S4: Convolutions

- LTI SSMs allow Convolutional Representation
  - Start with SSM:

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

$$x_0 = \overline{B}u_0 \qquad x_1 = \overline{AB}u_0 + \overline{B}u_1 \qquad x_2 = \overline{A}^2\overline{B}u_0 + \overline{AB}u_1 + \overline{B}u_2$$
$$y_0 = \overline{CB}u_0 \qquad y_1 = \overline{CAB}u_0 + \overline{CB}u_1 \qquad y_2 = \overline{CA}^2\overline{B}u_0 + \overline{CAB}u_1 + \overline{CB}u_2$$

$$\ldots \quad y_k = \overline{CA}^k\overline{B}u_0 + \overline{CA}^{k-1}\overline{B}u_1 + \cdots + \overline{CAB}u_{k-1} + \overline{CB}u_k$$

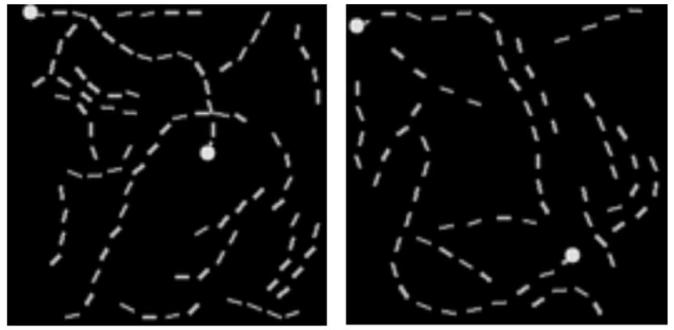
$$\Rightarrow \quad y = \overline{K} * u$$

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

  - Parallelize by pre-computing kernel K
    - Naïve Computation: $O(N^2L)$ computations, $O(NL)$ space
      - Prohibitive for large N
    - Need more efficient ways to compute K
  - Make use of structure on A (HiPPO Matrix):
    - It has NPLR form
    - Allows to compute states in $\tilde{O}(N+L)$ computations

S4: Performance

- Long Range Arena (LRA)
  - Benchmark by Tay et al., 2020
    - Tasks to test models' ability to reason over very long sequences
  - Pathfinder and PathX:
    - Test model's ability to learn long range spatial dependencies
    - PathX: 128 x 128 images
      - Requires ability to reason over a context length of 16384
  - S4: First model to crack PathX
- Still bad at Language Modelling

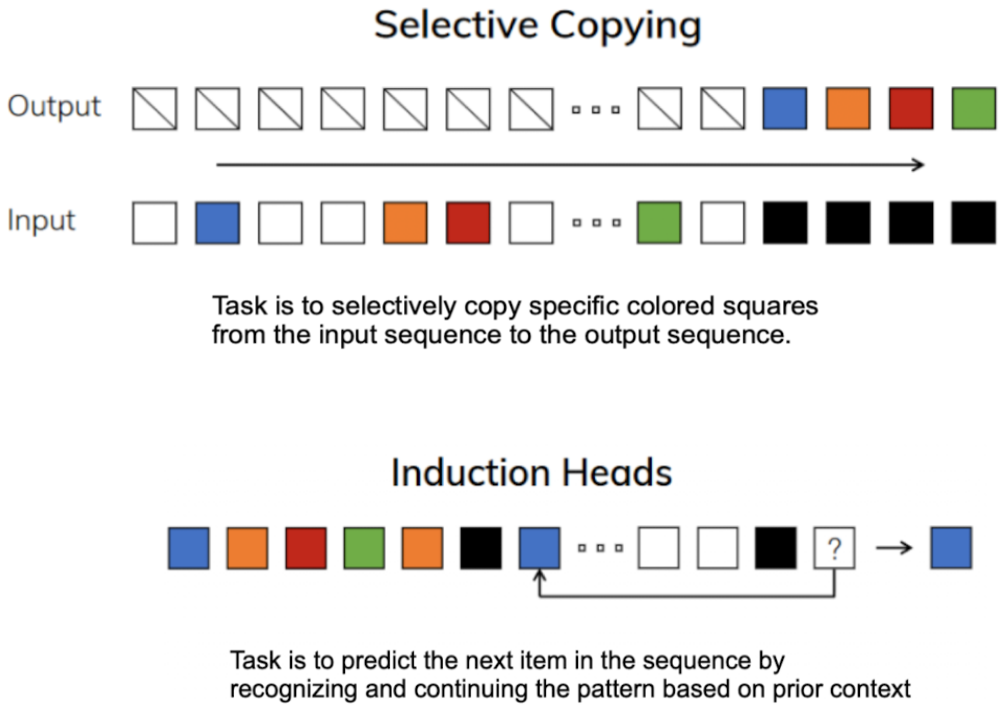


(a) A positive example. (b) A negative example.

Limitations of LTI

- Parameters of SSMs are independent of inputs
  - Makes it bad at tasks like Language Modelling
    - They require input dependent selection

- Examples:
  - Selective Copying:
    - Useful for retrieval tasks
  - Induction Heads:
    - Useful for in-context learning



Mamba: S6

- By Gu et al., 2023
  - Adds input dependent selectivity to S4
    - Make Δ, B, C input dependent
- Initialization of A:
  - S4D Initialization by Gu et al., 2022
    - Complex: $A_k = -\frac{1}{2} + ik$
      - Suitable in low data regimes
    - Real: $A_k = -(k+1)$
- Discretization:
  - Zero-Order Hold (ZoH):

$$\mathbf{x}(t+\Delta) \simeq e^{\Delta \mathbf{A}}\mathbf{x}(t) + (\Delta\mathbf{A})^{-1}(e^{\Delta\mathbf{A}} - \mathbf{I})(\Delta\mathbf{B})\mathbf{u}(t)$$

    - Converts SSM ODE to recurrence via:

$$\overline{\mathbf{A}} = e^{\Delta\mathbf{A}}$$

$$\overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(e^{\Delta\mathbf{A}} - \mathbf{I})(\Delta\mathbf{B})$$

- Making Δ, B, C input dependent:
  - Pass input **(u)** through linear projections:

$$\mathbf{B} = Linear_N(\mathbf{u})$$

$$\mathbf{C} = Linear_N(\mathbf{u})$$

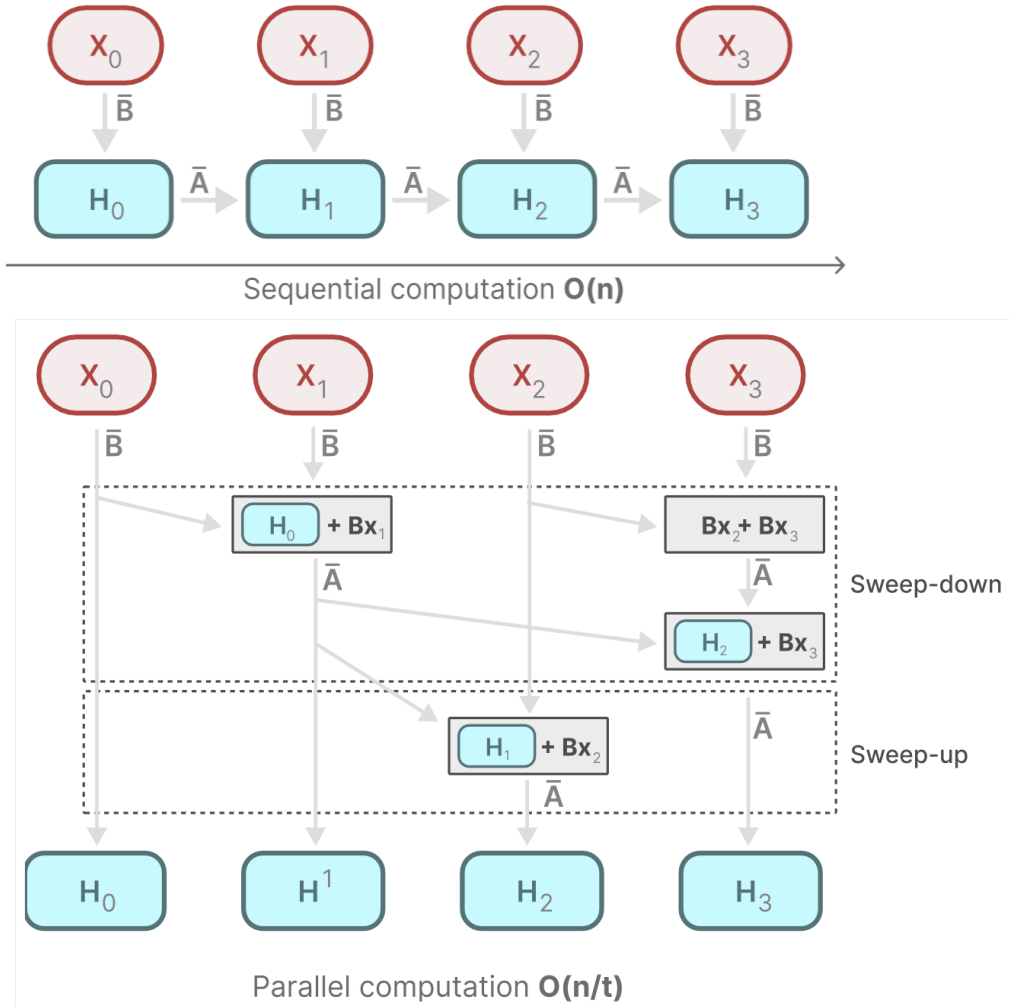
$$\Delta = Broadcast_D(Linear_1(\mathbf{u}))$$

Here, $Linear_d$ represents a parameterized (and learnable) linear projection to dimension d, and $Broadcast_D$ expands the output of a linear layer to dimension D by replicating each element.

    - Even though **A** is constant, $\bar{\mathbf{A}}$ is input dependent
      - Because of it's dependence on **Δ**

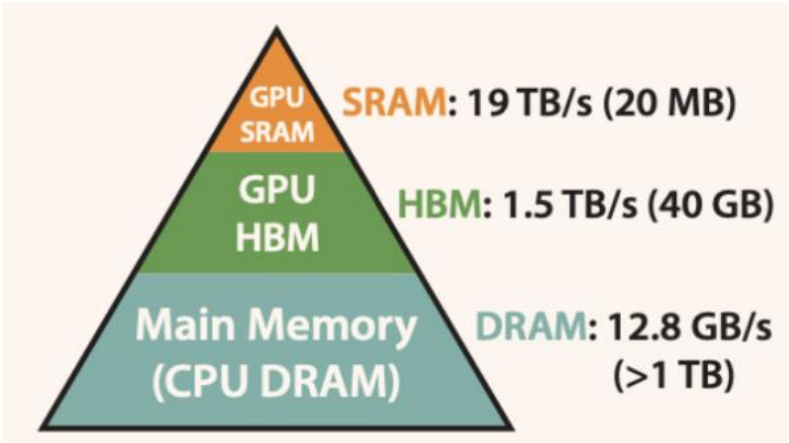
Mamba: Parallel Scan

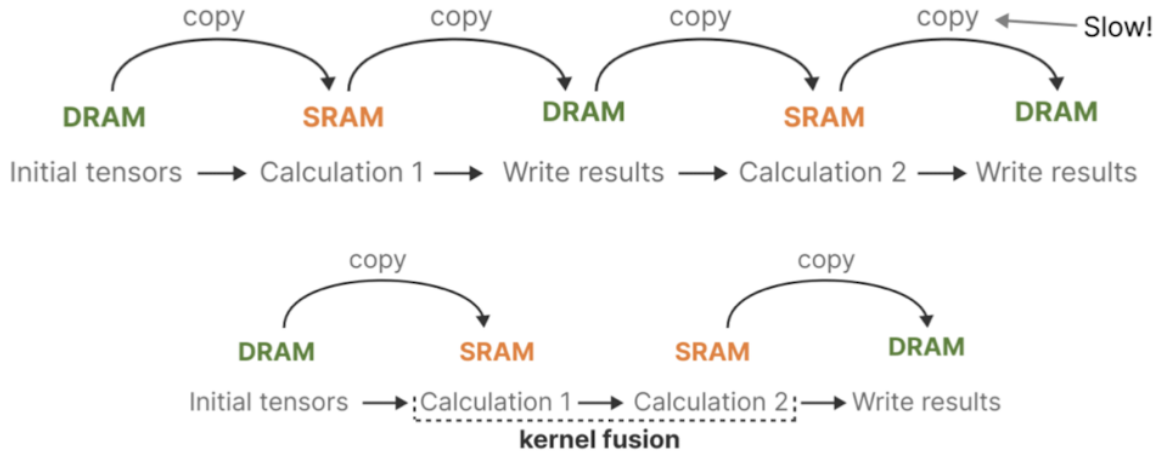
- $\bar{A}$ $\bar{B}$ $\bar{C}$ are input dependent
  - This breaks LTI of the system
    - Can't use Convolutional Form anymore
- Associative Parallel Scan
  - Inspired from the Prefix Sum Problem
    - Recurrence is still Linear and Associative
    - Improves computation from O(n) to O(n/t) steps
      - `t` is the # of threads

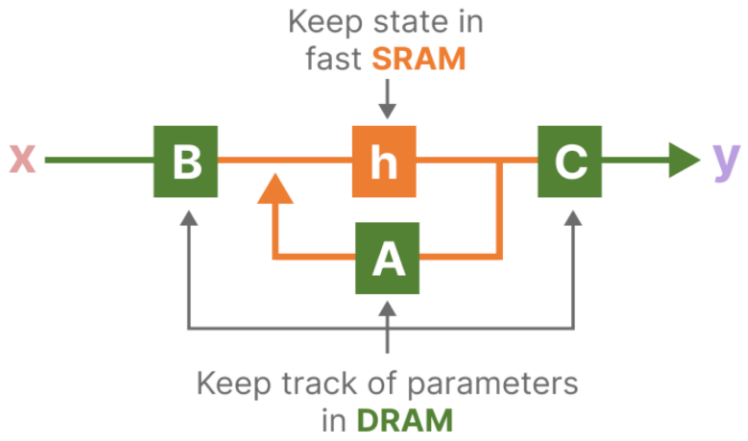


Mamba: Hardware Optimizations

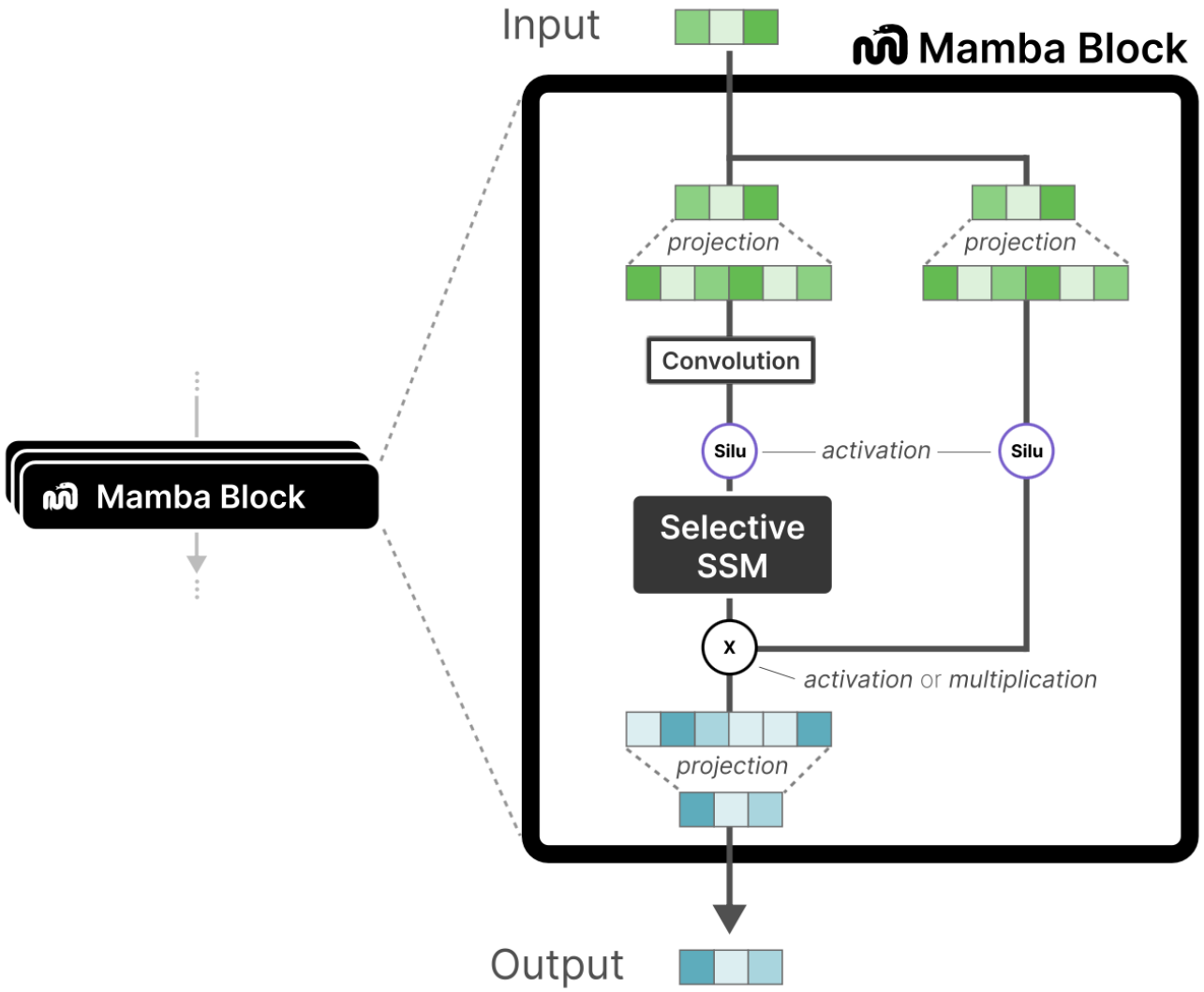
- GPUs limited by I/O speeds
  - Copying between SRAM to DRAM takes more time than the computations
  - Use Kernel Fusion:
    - Custom Kernel prevents models from writing intermediate results and continuously performs computations until it's done
    - Like Flash Attention
      - By Dao et al., 2022





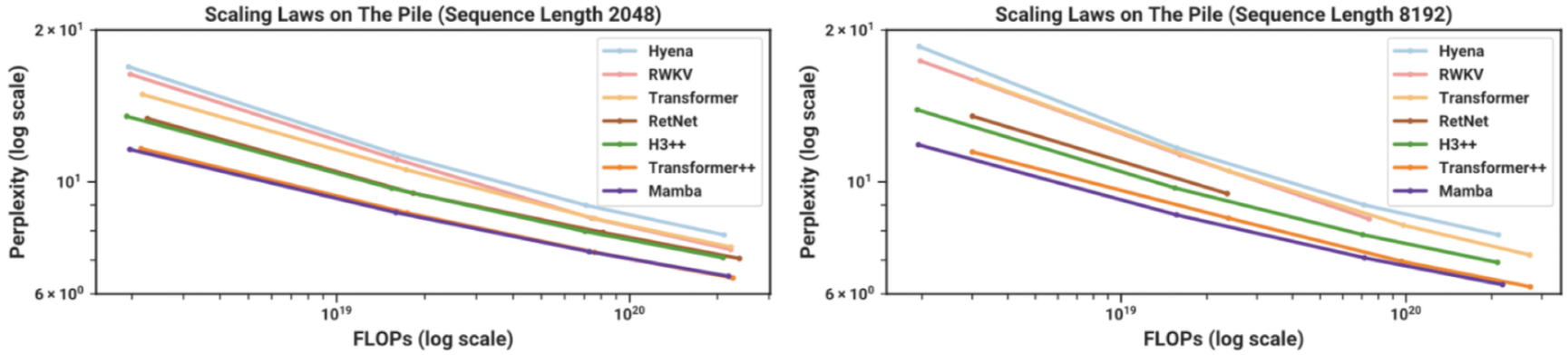


Mamba: Block


Input
Mamba Block
projection
projection
Convolution
Silu
activation
Silu
Selective SSM
x
activation or multiplication
projection
Output
Mamba Block

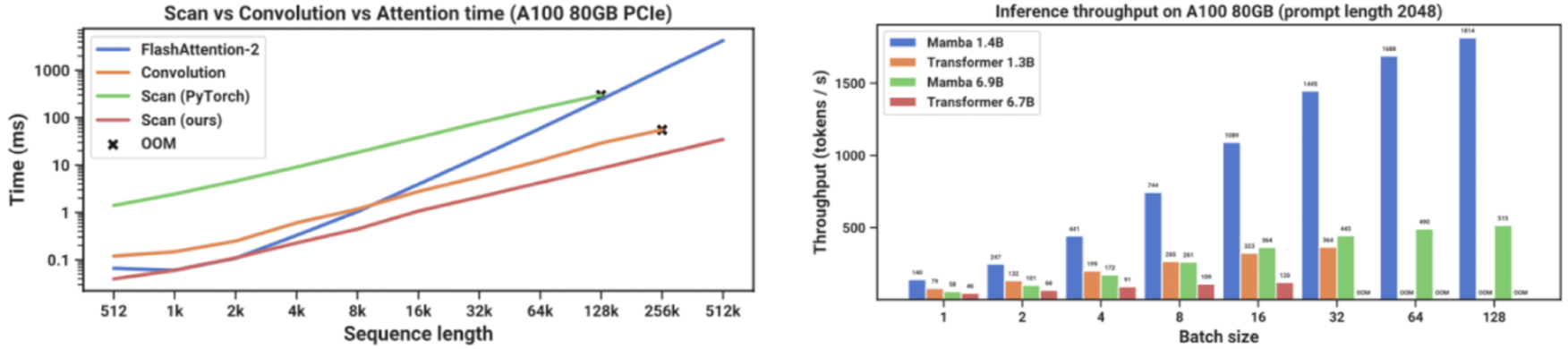
Mamba: Performance

- Scaling Laws



Scaling Laws of models with sizes from 125M − 1.3B parameters, trained on Pile Dataset

- Efficiency Benchmarks



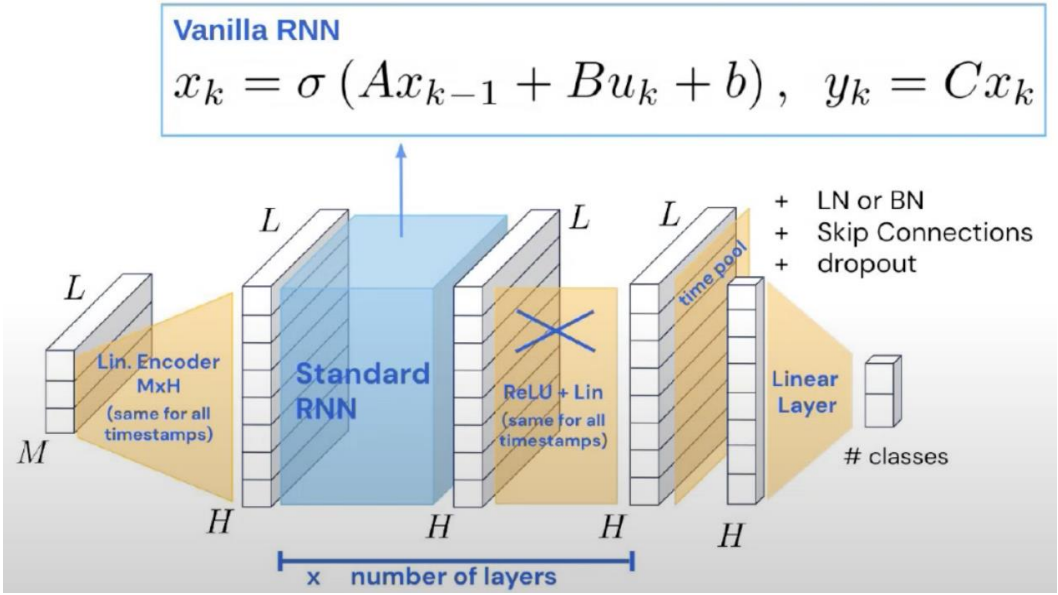
(Left) Training Times (Right) Inference Times of Mamba vs Transformers

Disentangling What is Important

- Many variants of SSMs proposed:
  - Multiple choices for initializations of **A**:
    - HiPPO, S4D-Real/Complex, H3 etc.
  - Multiple choices for discretization:
    - Euler, Bilinear, ZoH etc.
  - Unclear what to use when
    - Unclear how changes to block interact with SSM parametrizations
  - Very complicated mathematical framework
- Can we get away with something simpler?
  - Can we start with Vanilla RNN and make careful modifications?
    - To achieve comparable performances to SSMs in LRA tasks

Linear Recurrent Unit (LRU)

- By Orvieto et al., 2023
  - Go back to transformer core
    - Replace MHA with Vanilla RNN
- Highlights of LRU:
  - Instead of SSMs, Linear RNNs are sufficient
    - Due to the block structure (including MLPs)
  - Linearity allows diagonal recurrence
    - Computationally efficient
  - Linear + Diagonal helps with Stability
    - Direct control of linear system with eigenvalues of State Matrix (**A**)
  - Deterministic Initialization of A is not necessary
  - Continuous ODE form is not necessary
    - Hence, no need for discretization



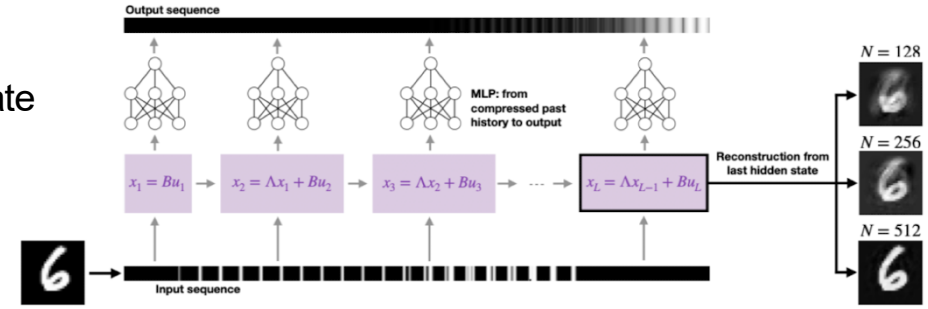
LRU: Removing Non-Linearity from Recurrence

- Expressivity?
  - Guaranteed by Universality Theorem for Linear RNNs:
    - By Orvieto et al., 2024
    - Theorem (Informal):
      - Linear RNNs followed by non-linear projections can approximate arbitrarily well any finite Seq2Seq map
        - RNN act as compressor
        - MLP for reconstruction
    - Seems better for generalization:
      - LRNNs act as weak learners
      - Significant gains in expressivity by stacking blocks (unlike LSTMs)
  - Empirical observations on LRA:

$$x_k = \cancel{\sigma}(Ax_{k-1} + Bu_k + b), \quad y_k = Cx_k$$

Recurrence	sCIFAR	ListOps	Text	Retrieval
RNN-ReLU	69.7 (0.2)	37.6 (8.0)	88.0 (0.1)	88.5 (0.1)
RNN-Tanh	69.9 (0.3)	43.9 (0.1)	87.2 (0.1)	88.9 (0.2)
RNN-Lin	**72.2** (0.2)	**50.4** (0.2)	**89.1** (0.1)	**89.1** (0.1)

Effect of removing the non-linearity from the recurrence on the feasible LRA tasks



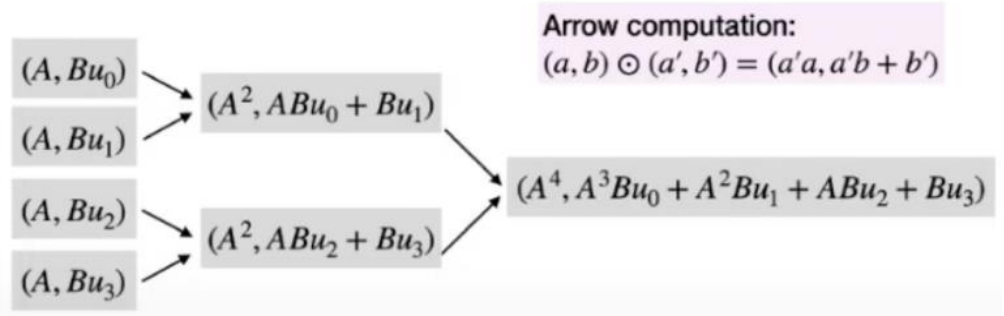
Linear RNN + position-wise MLP for flattened MNIST digits reconstruction. The role of linear RNN is to compress and store the input sequence into the hidden states, from which we can recover past tokens using a MLP. As hidden size N increases, the reconstruction becomes more and more faithful.

LRU: Benefits of Linear RNNs

- They can be **parallelized**:
  - Like Mamba, associative parallel scan still applicable
- They can be **diagonalized**:
  - Almost all* matrices can be diagonalized in $\mathbb{C}$

$\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$, where $\mathbf{P} \in \mathbb{C}^{N\times N}$ is an invertible matrix

$\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_N) \in \mathbb{C}^{N\times N}$ is a diagonal matrix



- Start with LRNN recurrence: $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k$

Assuming $\mathbf{x}_{-1} = 0$: $\mathbf{x}_k = \sum_{j=0}^{k-1} \mathbf{A}^j \mathbf{B}\mathbf{u}_{k-j}$

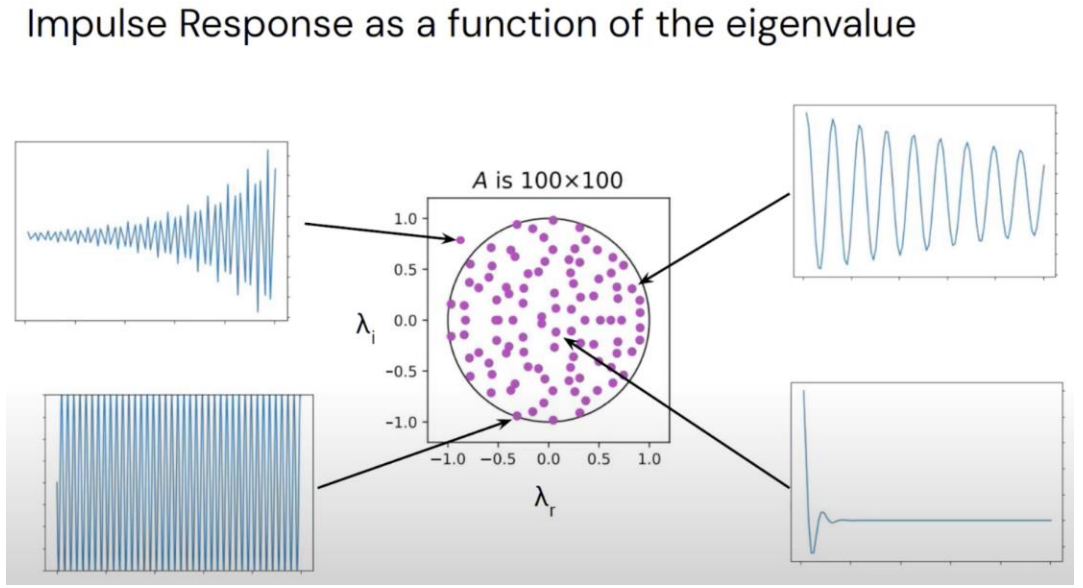
Renaming $\overline{\mathbf{x}_k} \leftarrow \mathbf{P}^{-1}\mathbf{x}_k$ and $\overline{\mathbf{B}} \leftarrow \mathbf{P}^{-1}\mathbf{B}$:

$$\overline{\mathbf{x}_k} = \sum_{j=0}^{k-1} \mathbf{\Lambda}^j \overline{\mathbf{B}}\mathbf{u}_{k-j}$$

- Rewrite the recurrence as: $\overline{\mathbf{x}_k} = \mathbf{\Lambda}\overline{\mathbf{x}}_{k-1} + \overline{\mathbf{B}}\mathbf{u}_k$

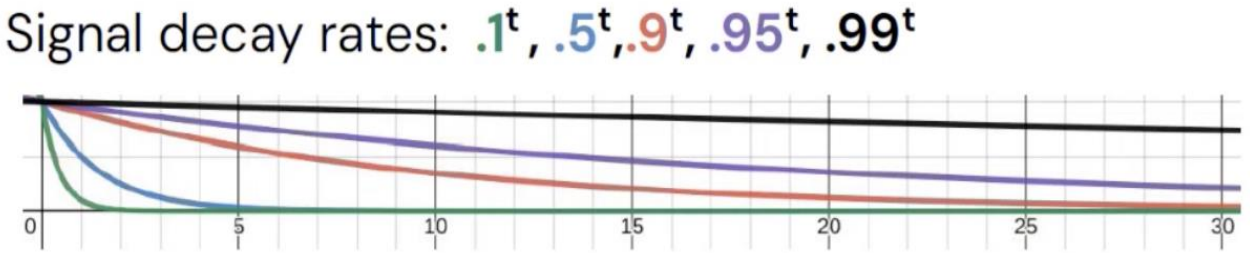
LRU: Controlling System

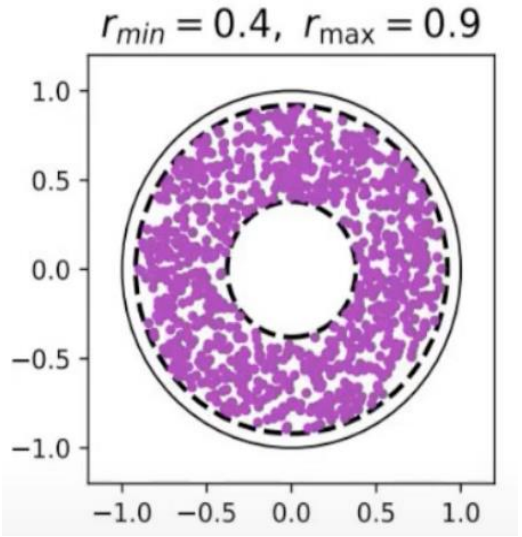
- Directly learn complex diagonal **A**
  - Exponentiation is trivial
    - Fewer recurrent computations
  - Eigenvalues are diagonal entries themselves!
    - Magnitude of eigenvalues controls the amplitude of impulse response
    - Phase of eigenvalues controls the oscillation of the response
  - Directly initialize the diagonals:
    - We want to sample from the unit disc
      - This is also backed by Xavier-Glorot initialization of Dense RNNs
        - Strong Circular Law says that such initializations have eigenvalues uniformly sampled from unit disc.



LRU: Stability (1)

- Preventing Exploding Gradients
  - Initialize magnitude and phase of diagonals separately: $\Lambda_i = e^{\alpha+i\beta}$
    - Ensure α<0:
      - This ensures magnitude is <1
- Preventing Vanishing Gradients
  - Eigenvalues near origin decay quickly
    - Initialize on a ring close to unit circle
      - Set r_{min} and r_{max}
        - Typical values (0.9, 0.999)
- High resolution parametrization
  - Double exponential for magnitude:
    - Small changes in decay rate near 1:
      - big differences
      - Ex: 0.95 vs 0.99
    - Large changes in decay rate when close to 0:
      - minimal differences
      - Ex: 0.1 vs 0.5
    - Replace α < 0 with –exp(α)





LRU: Stability (2)

- Initializing small phase:
  - Large phase (β) means more oscillation
    - Instability during training, slower convergence
    - Replace β with exp(β), where β<0
- Normalization
  - Issues with large eigenvalues (|Λ| close to 1)
    - They accumulate information over many tokens
    - Recurrent features become too large
    - Result (Informal):
      - Assume inputs are random and uncorrelated
      - At the limit: $\mathbb{E}|x_k|^2 \xrightarrow{\infty} \sum_{i=0}^{\infty} |\lambda|^{2i} = \frac{1}{1-|\lambda|^2}$
  - Normalize the recurrence:

$$x_{k+1} = \lambda x_k + \gamma \cdot B u_k$$

$$\gamma = \sqrt{(1 - |\lambda|^2)}$$

- Are Complex no.s necessary?
  - Diagonalization of arbitrary linear systems
    - Complex eigenvalues
  - Real eigenvalues:
    - Symmetric linear system
    - Result (Informal):
      - We can create a symmetric system doing similar computation by blowing up the state size (at least twice as large)
  - Complex eigenvalues give compact state representations!

$$A\mathbf{x} = \frac{1}{2}\begin{pmatrix} 0 & A \\ A & 0 \end{pmatrix}\begin{pmatrix} \mathbf{x} \\ \mathbf{x} \end{pmatrix}$$

LRU: Final Form

$$\mathbf{x}_k = \mathbf{diag}(\lambda)\mathbf{x}_{k-1} + \gamma \odot \mathbf{B}\mathbf{u}_k$$

where:

- $\lambda_j = \exp(-\exp(v_j^{log})) + i\exp(\theta_j^{log}))$ are the diagonal elements of the recurrent matrix, parameterized using the stable exponential form with enforced stability.
- $\gamma_j = \sqrt{1 - |\lambda_j|^2}$ are the normalization factors, ensuring stability during the forward pass.
- $\mathbf{B}$ is the input projection matrix, mapping the input to the hidden state.

LRU: Performance

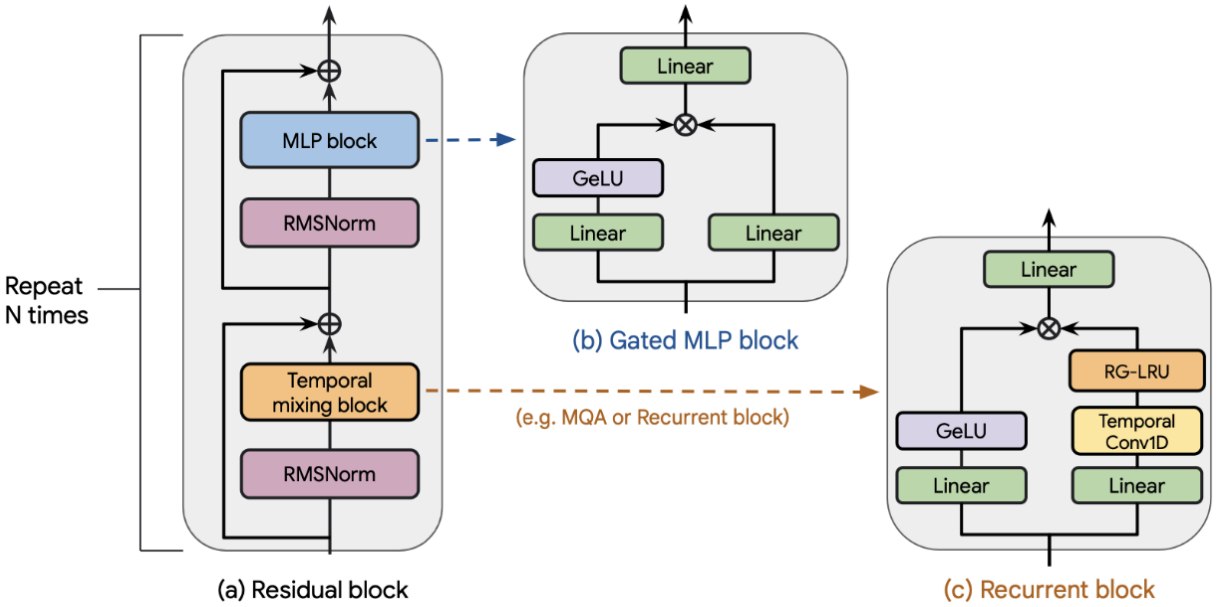
- Long Range Arena (LRA)

	sCIFAR	ListOps	Text	Retrieval	Pathfinder	PathX
Linear Dense RNN	72.2 (0.2)	50.4 (0.2)	89.1 (0.1)	89.1 (0.1)	✗	✗
Diagonal Complex RNN	86.5 (0.1)	58.8 (0.3)	87.4 (0.3)	87.8 (0.5)	✗	✗
Stable Exp Param w/ Ring Init	88.1 (0.0)	59.4 (0.3)	89.4 (0.1)	90.1 (0.1)	94.4 (0.3)	✗
$[r_{\min}, r_{\max}]$	[0.9, 0.99]	[0.0, 1.0]	[0.0, 0.9]	[0.5, 0.9]	[0.9, 0.999]	
$+\gamma$ Normalization **(LRU)**	89.0 (0.1)	60.2 (0.8)	89.4 (0.1)	89.9 (0.1)	95.1 (0.1)	94.2 (0.4)
$[r_{\min}, r_{\max}]$	[0.9, 0.999]	[0.0, 0.99]	[0.5, 0.9]	[0.5, 0.9]	[0.9, 0.999]	[0.999, 0.9999]
S4D (our reproduction)	91.5 (0.2)	60.2 (0.3)	86.4 (0.0)	89.5 (0.0)	94.2 (0.3)	97.5 (0.0)
S5 (our reproduction)	88.8 (0.1)	58.5 (0.3)	86.2 (0.1)	88.9 (0.0)	95.7 (0.1)	96.0 (0.1)
S4 (paper results)	*91.1*	*59.6*	*86.8*	*90.9*	*94.2*	*96.4*
S4D-LegS (paper results)	89.9	60.5	86.2	89.5	93.1	91.9
S5 (paper results)	*90.1*	*62.2*	*89.3*	*91.4*	95.3	*98.6*

LRU for Language Modelling: Hawk and Griffin

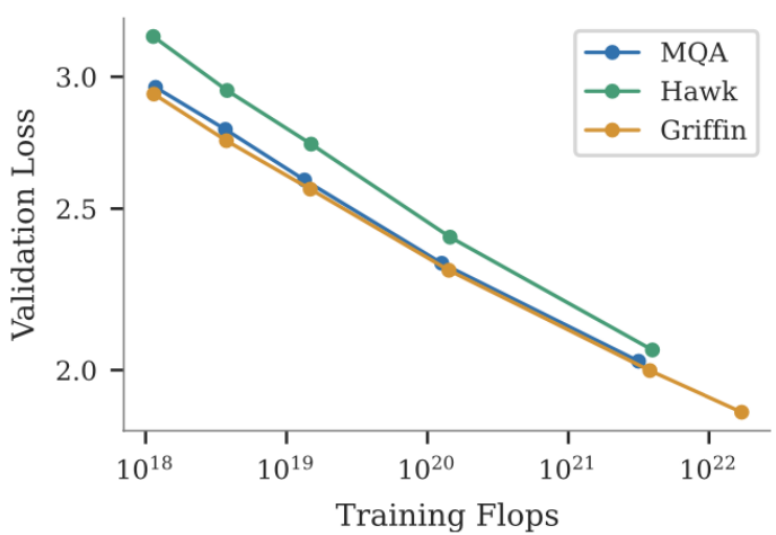
- Modifies LRU to RG-LRU:
  - Real Gated LRU
    - Makes model scalable for multi-GPU systems
      - Megatron Sharding
  - Modifies LRU by adding:
    - Recurrent and Input Gates
      - Helps bring input dependent selectivity to LRUs
        - Like Mamba for S4
    - Elementwise multiplication instead of matrix multiplication
      - Enable better parallelization
  - Conv 1D Layer:
    - Helps modelling very short-term dependencies
      - Filter size usually 4
  - Final Form:

$$\mathbf{r}_t = \sigma(\mathbf{W}_a \mathbf{x}_t + \mathbf{b}_a), \text{ recurrence gate}$$
$$\mathbf{i}_t = \sigma(\mathbf{W}_x \mathbf{x}_t + \mathbf{b}_x), \text{ input gate}$$
$$\mathbf{a}_t = \mathbf{a}^{c\mathbf{r}_t},$$
$$\mathbf{h}_t = \mathbf{a}_t \odot \mathbf{h}_{t-1} + \sqrt{1 - \mathbf{a}_t^2} \odot (\mathbf{i}_t \odot \mathbf{x}_t)$$

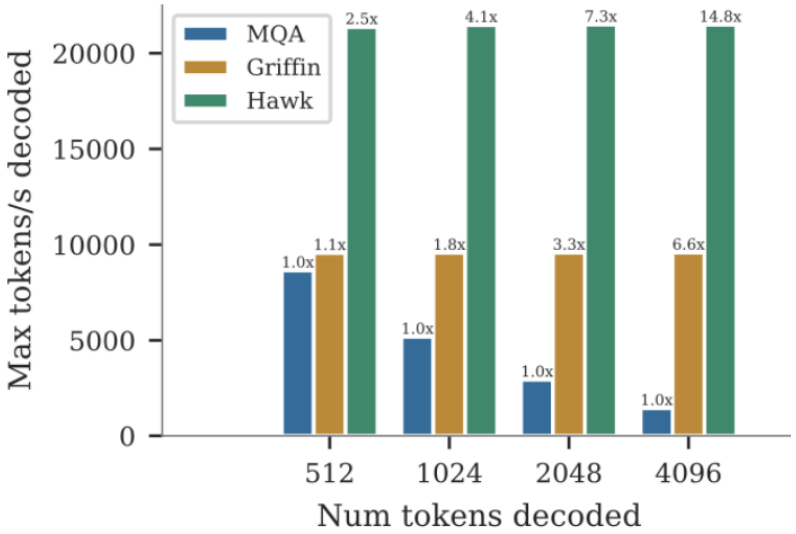


Hawk and Griffin

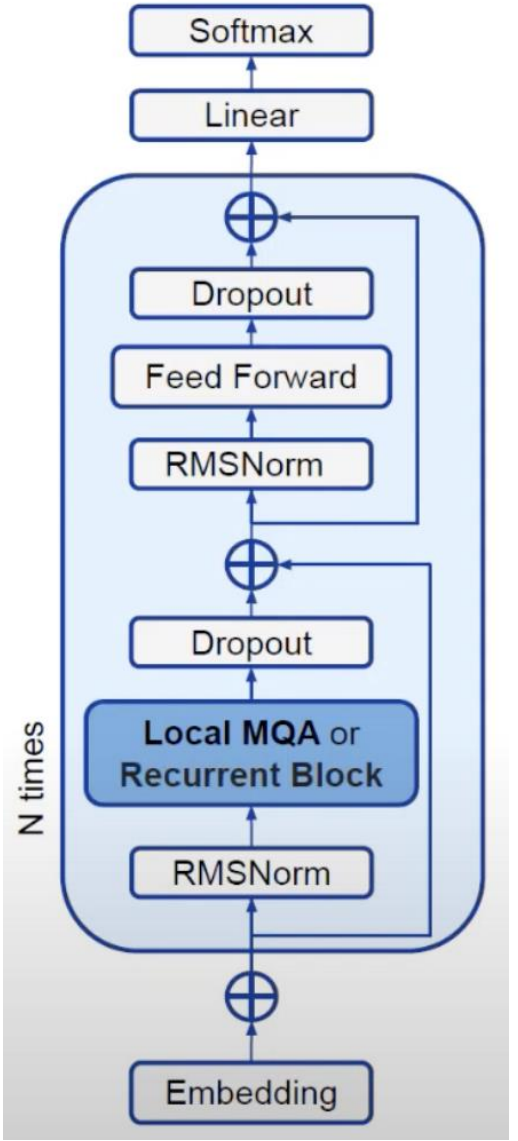
- Hawk:
  - Uses only RG-LRU
- Griffin:
  - Uses 2 RG-LRU + 1 Local Attention Layer alternatively
- Scaling and Throughput:



(a) Scaling curve during training



(b) Maximum throughput at 1B parameter scale.



Hawk and Griffin: Performance

- Character Normalized Accuracy:
  - On various downstream tasks

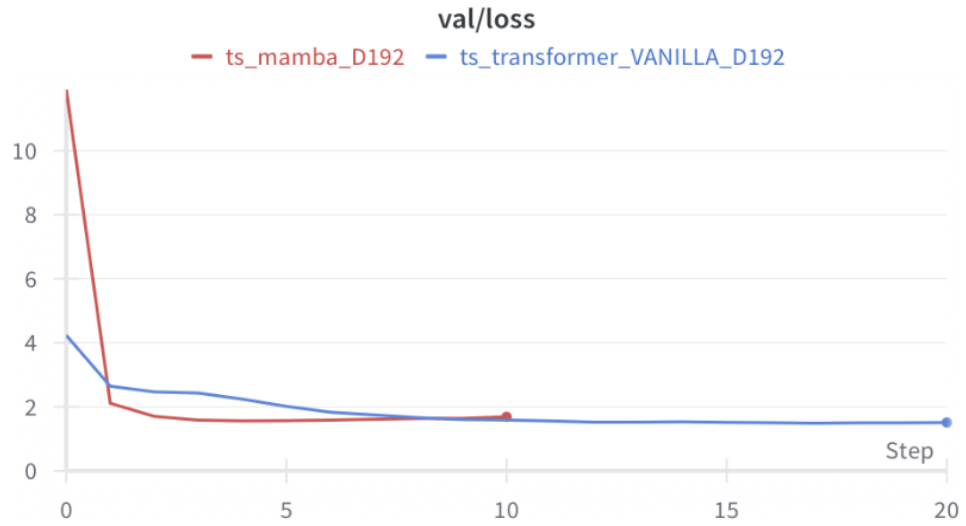
Model Type	Model Size	Training Tokens	MMLU	HellaSwag	PIQA	WinoGrande	ARC-E	ARC-C	Average
Mamba	3B	600B	26.2	71.0	78.1	65.9	68.2	41.7	58.5
Llama-2	7B	2T	45.3	77.2	78.8	69.2	75.2	45.9	65.3
	13B	2T	**54.8**	80.7	80.5	72.8	77.3	49.4	69.3
MQA Transformer (Baseline)	1B	300B	28.9	64.8	75.0	62.0	60.2	35.4	54.4
	3B	300B	31.7	71.0	77.6	66.1	68.1	39.2	59.0
	6B	300B	38.9	77.0	79.5	70.4	74.1	45.2	64.2
Hawk	1B	300B	29.7	63.3	76.1	57.2	60.6	34.6	53.6
	3B	300B	31.3	71.7	78.8	66.5	68.4	40.2	59.5
	7B	300B	35.0	77.6	80.0	69.9	74.4	45.9	63.8
Griffin	1B	300B	29.5	67.2	77.4	65.2	67.0	36.9	57.2
	3B	300B	32.6	73.5	78.1	67.2	71.5	41.4	60.7
	7B	300B	39.3	78.6	81.0	72.6	75.4	47.9	65.8
	14B	300B	49.5	**81.4**	**81.8**	**74.1**	**79.1**	**50.8**	**69.5**

Experiments

- Datasets:
  - Tiny Shakespeare:
    - Text
  - Maestro V2
    - Audio
  - Seq MNIST
    - Image
- Models:
  - Transformer (GPT 2)
    - Context Window: 256
    - 2.4 M Params
    - Cosine LR: [10^{-3}, 10^{-4}]
  - Mamba
    - 2.2 M Params
    - Cosine LR: [10^{-4}, 10^{-5}]

Experiment: Tiny Shakespeare

- Dataset
  - 40000 lines of text from plays by Willian Shakespeare
  - Task is to train a character-level LM
    - Model should generate Shakespeare-like text
- Results



Generated Text #0

And it say the punish be the liberty.
And I were not thee to little with the sacring
The seat of itself: you shall not person.

ISABELLA:
'Tis this word of the country's cause to the
Musician: for it, not still us, and so.

DUKE VINCENTIO:
Is but known not I have proved no love for
Than bring serves me have no such a soul,
As a man to die, I can be with him.

ESCALUS:
There is none reason!
O, the people and a form death.

DUKE VINCENTIO:
This is true, to leave your honour, and with warr

Generated Text #1

Not know you be confession, profess, and love.

ROMEO:
Come; at her a fearful, a pil-night, and the world.

BENVOLIO:
O, that sovereign that shall be heary be appeal as
subjects, and true remove the time: therefore there man.

ROMEO:
How dost too exile of the fair of call
Have with a slain and beauty.

(a) Vanilla GPT Transformer

Generated Text #0

And let me gain of me; for his brother of my good sense
That which amble lord, as gladly servants of the coronation
That thou art a shame, wife and medicy
Their men made me stilpsh is wellest from the city.

First Xest thou liest.

KING RICHARD II:
Ah, knock a beaver play'd witness but thy soul,
And what we life and gives, and Paris in the town
And longer are not so fail, this here and men
To understand that fills and unybattle and
To three of my good lord;
My lord, and he hath a heaven blood

Generated Text #1

PARIS:
Oncy to your house, marry, whence my son, the poor eyes death
To better than on my power coldly.
O our dishonour'd and vain despair.

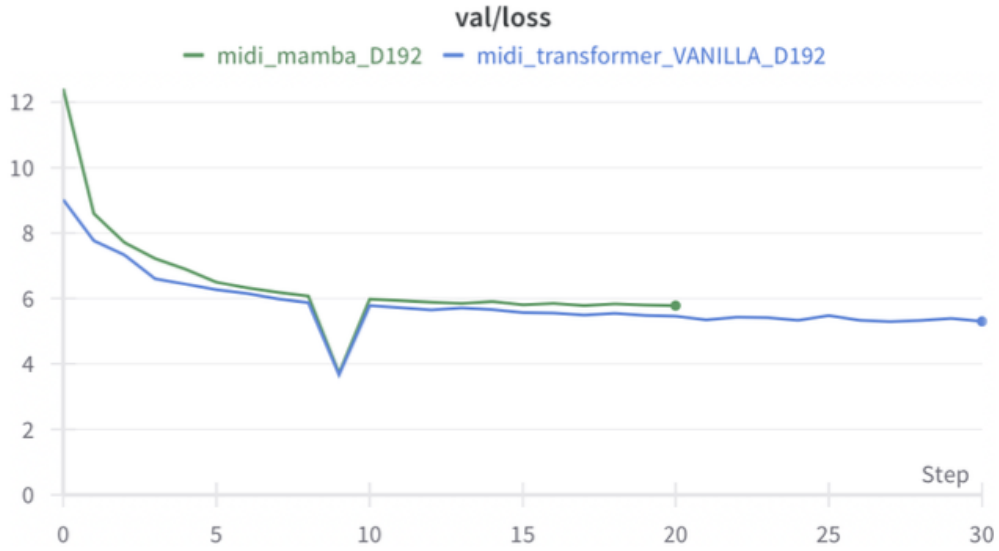
DUKE VINCENTIO:
Return your treason: bond thus.

ESCALUS:
Look, what I have seen a milk'd in heaven,
That thus and his loyal drinks, do London can makes the curst word
Than a duke of the Faenting leaves and every one,
The coronation of his unto the travellect,
A holy eyes him well, my lord,
If gracious lord, a plenteous counsel:

(b) Mamba

Experiment: Maestro V2

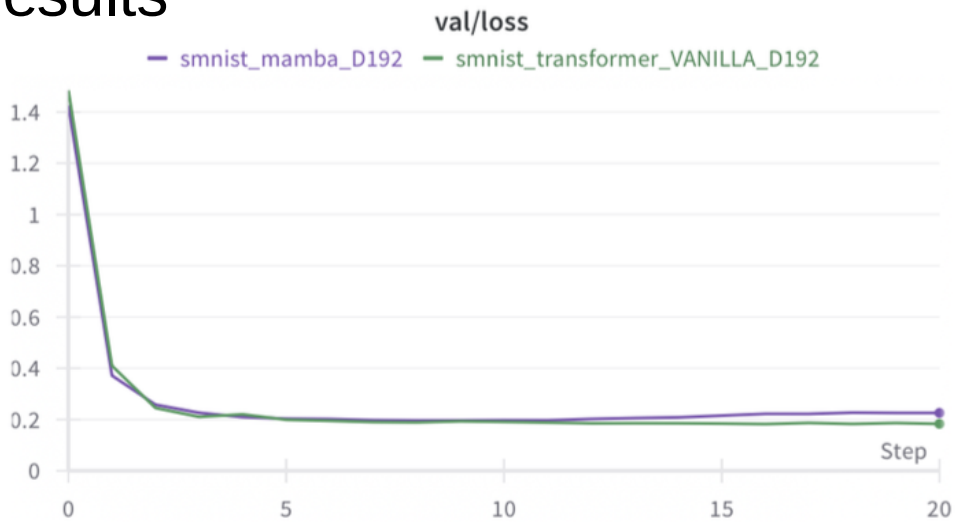
- Dataset:
  - 200 hours of MIDI recordings
    - From 10 years of International Piano-e-Competition
  - Task is to train MIDI generating LM
    - Model should generate smooth sounding piano songs
- Results

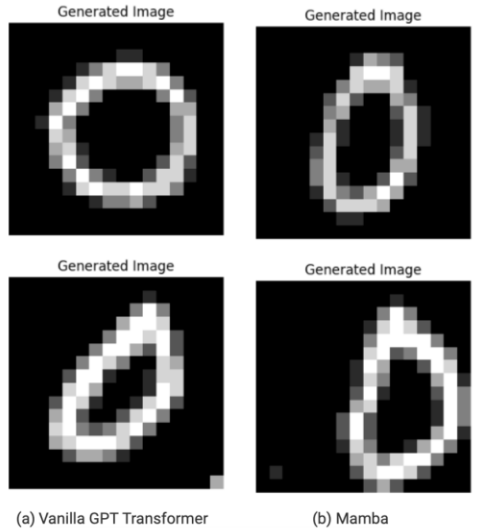


Experiment: Seq MNIST

- Dataset
  - 60000 handwritten digits
    - Pixels quantized to 3-bits: [0,7]
    - Images reshaped to (16,16)
    - Flattened to 256-dim 1D sequence.
      - Flattening technique:
        - Left to Right, row after row
    - Model trained only on '0' digit
  - Task is to train '0' generating LM
    - Model should generate MNIST '0' looking images:
      - sequentially from scratch

- Results





(a) Vanilla GPT Transformer (b) Mamba

THANK YOU