# Mamba: Linear-Time Sequence Modeling with Selective State Spaces

**Albert Gu, Tri Dao**

**COLM 2024**

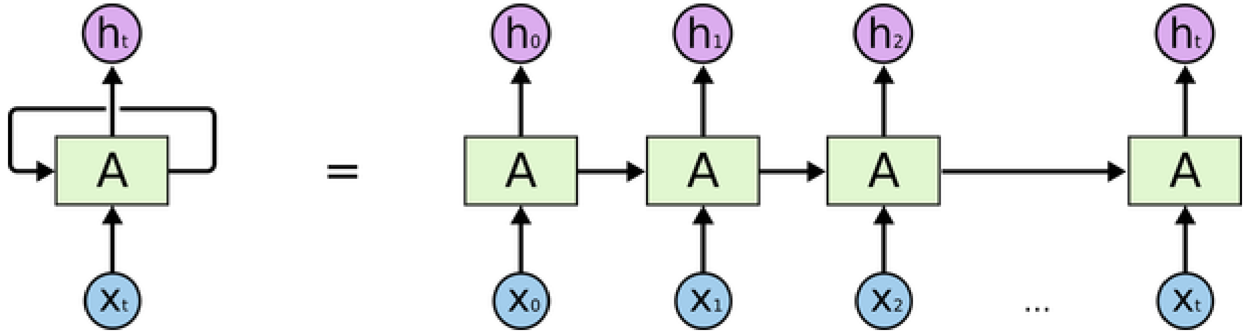
HUMANE Lab 석사과정 최종현

랩 세미나

2025.03.21

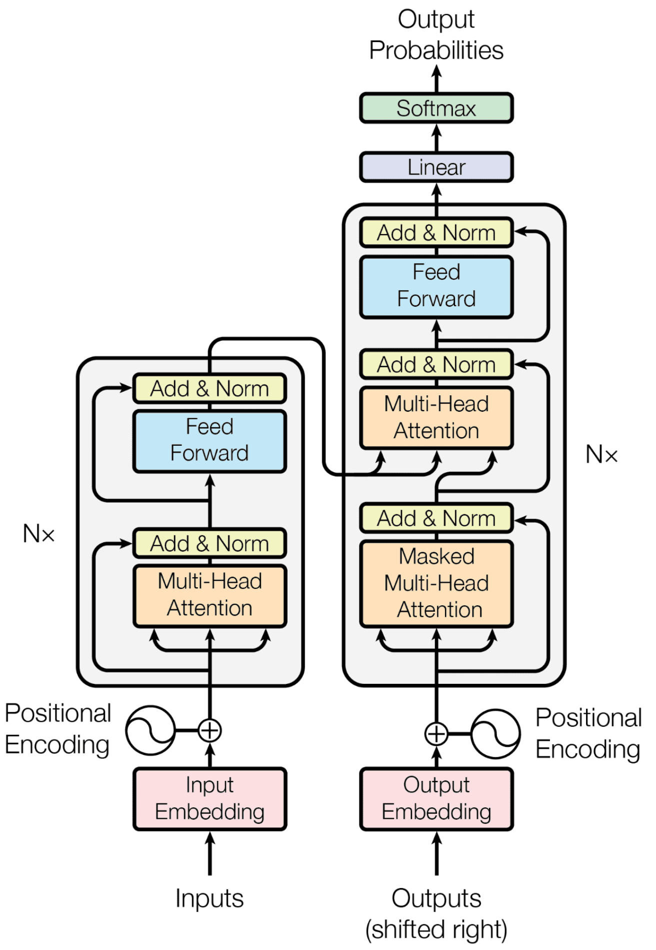
# Background

- RNNs are used for sequential modeling
- Ideal for language modeling, time series prediction, and speech recognition
- RNNs have limitations
  - fixed finite state
  - restricts ability to process long-context (i.e., vanishing gradient)
  - cannot compute in parallel

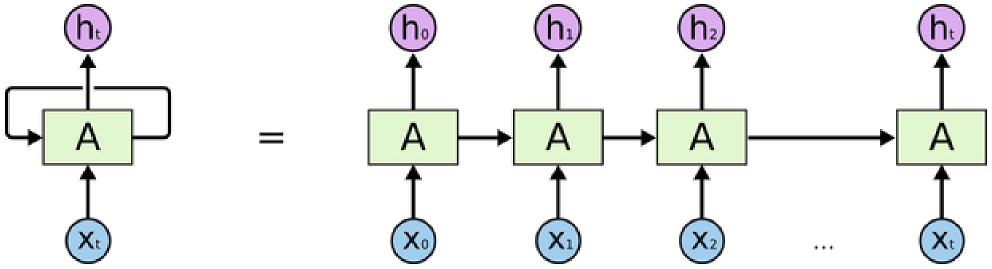


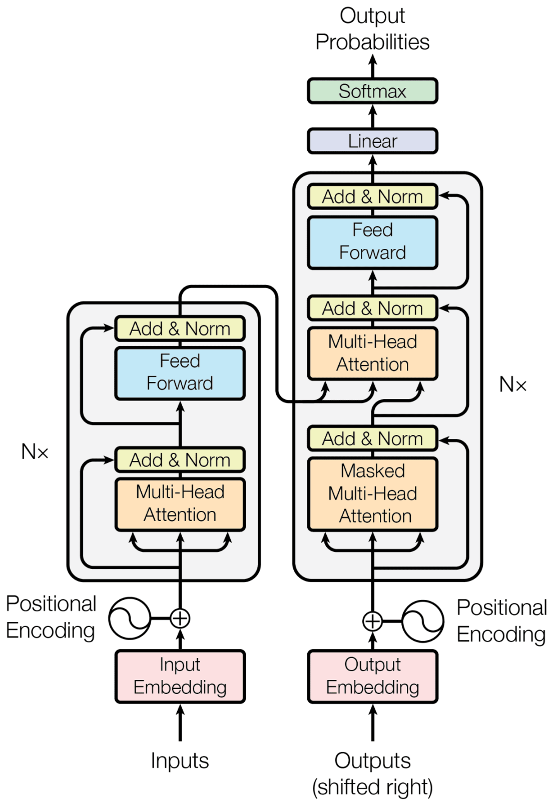
# Background

- Transformer uses attention
- Models all connections
- Long-range dependencies
- Can compute in parallel
- Transformers have limitations
  - self-attention scales quadratically with sequence length
  - require large caches during inference
  - limits context window size and efficiency
  - high computational costs 🔥



# Background



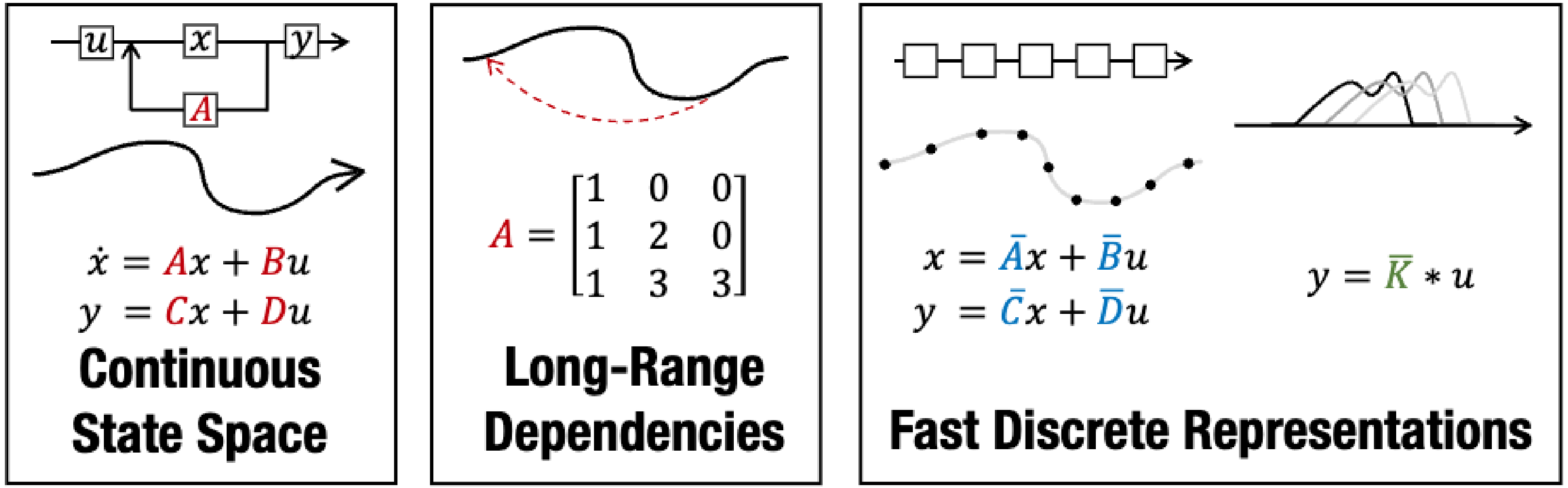




Efficiency 👍
Performance 👎

Efficiency 👎
Performance 👍

# Background

- Used in control theory to model a dynamic system via state variables
- Describes how a system evolves over time
- Handles sequential data



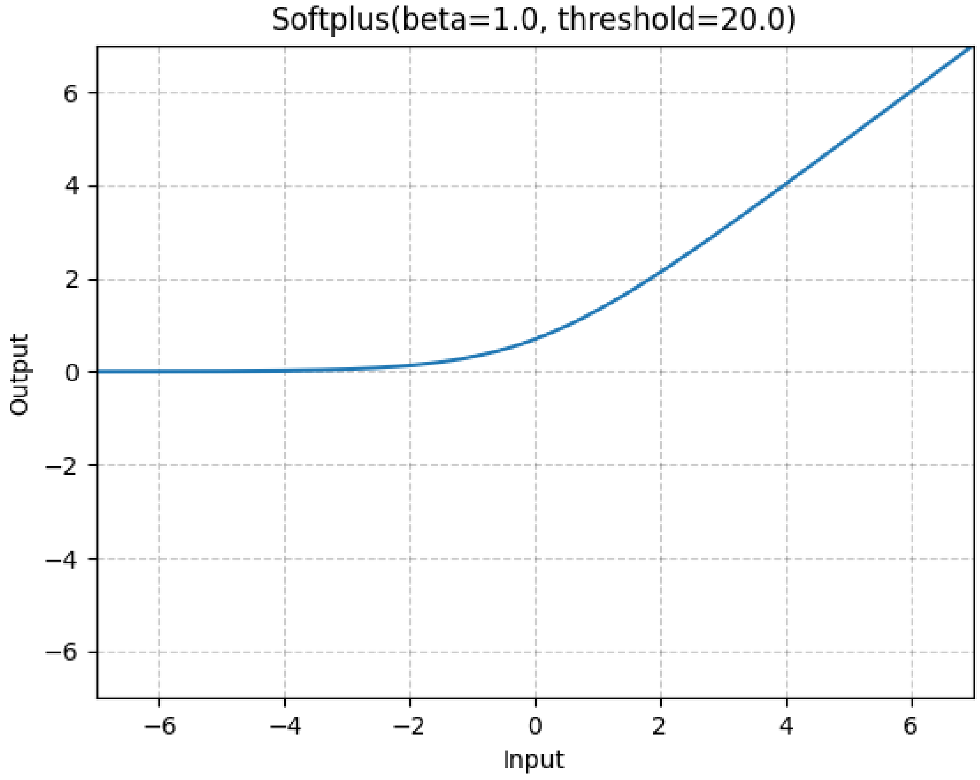
# Background

- Continuous SSM
  - $h(t)$: hidden state
  - $x(t)$: input
  - $y(t)$: output
  - $A \in \mathbb{R}^{N \times N}$: state transition matrix
  - $B \in \mathbb{R}^{N \times 1}$: input-to-state matrix
  - $\mathrm{C} \in \mathbb{R}^{1 \times N}$: state-to-output matrix
- State update equation: $h'(t) = \boldsymbol{A}h(t) + \boldsymbol{B}x(t)$
- Observation equation: $y(t) = \boldsymbol{C}h(t)$

# Background

- Zero-Order Hold (ZOH) is used for discretization
- ZOH converts continuous signal into a discrete one by holding each sampled value constant until the next sample is taken
- $\overline{A} = \exp(\Delta A), \overline{B} = (\Delta A)^{-1}(\exp(\Delta A) - \boldsymbol{I}) \cdot \Delta \mathrm{B}$
- $$\begin{aligned}\Delta_t &= \tau_\Delta(\text{Parameter} + s_\Delta(x_t)) \\ &= \text{softplus}(\text{Parameter} + \text{Linear}(x_t)) \\ &= \text{softplus}(\text{Linear}(x_t))\end{aligned}$$

# Background




$$\text{Softplus}(x) = \frac{1}{\beta} * \log(1 + \exp(\beta * x))$$

# Background

- Zero-Order Hold (ZOH) is used for discretization
- ZOH converts continuous signal into a discrete one by holding each sampled value constant until the next sample is taken
- $\overline{A} = \exp(\Delta A), \overline{B} = (\Delta A)^{-1}(\exp(\Delta A) - \boldsymbol{I}) \cdot \Delta \mathrm{B}$

$$\begin{aligned}
\overline{\boldsymbol{A}}_t &= \exp(\Delta \boldsymbol{A}) = \frac{1}{1+\exp(\mathrm{Linear}(x_t))} = \sigma(-\mathrm{Linear}(x_t)) \\
&= 1 - \sigma(\mathrm{Linear}(x_t)) \\
\overline{\boldsymbol{B}}_t &= (\Delta \boldsymbol{A})^{-1}(\exp(\Delta \boldsymbol{A}) - \boldsymbol{I}) \cdot \Delta \boldsymbol{B} = -(\exp(\Delta \boldsymbol{A}) - \boldsymbol{I}) = 1 - \overline{\boldsymbol{A}} \\
&= \sigma(\mathrm{Linear}(x_t)).
\end{aligned}$$

# Background

- Discrete SSM
  - $h_t$: hidden state
  - $x_t$: input
  - $y_t$: output
  - $\overline{A} \in \mathbb{R}^{N \times N}$: state transition matrix
  - $\overline{B} \in \mathbb{R}^{N \times 1}$: input-to-state matrix
  - $\overline{C} \in \mathbb{R}^{1 \times N}$: state-to-output matrix
- State update equation: $h_{t+1} = \overline{\boldsymbol{A}} h_t + \overline{\boldsymbol{B}} x_t$
- Observation equation: $y_t = \overline{\boldsymbol{C}} h_t$

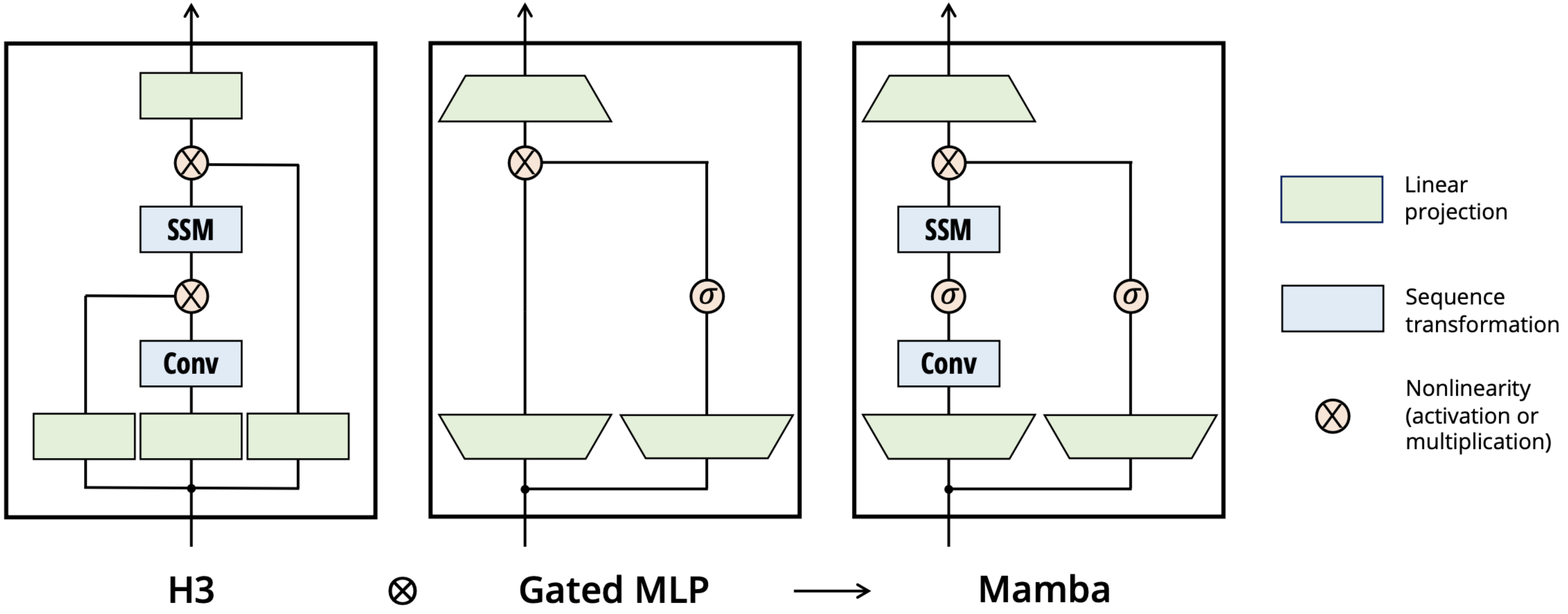
# Motivation

- Limitations of time-invariant (LTI) SSMs
  - In standard SSMs, parameter A, B, and C are fixed
  - LTI property restricts model’s ability to adapt dynamically to the content of the input
- Need for selection mechanism
- Dynamic gating via input-dependent parameters

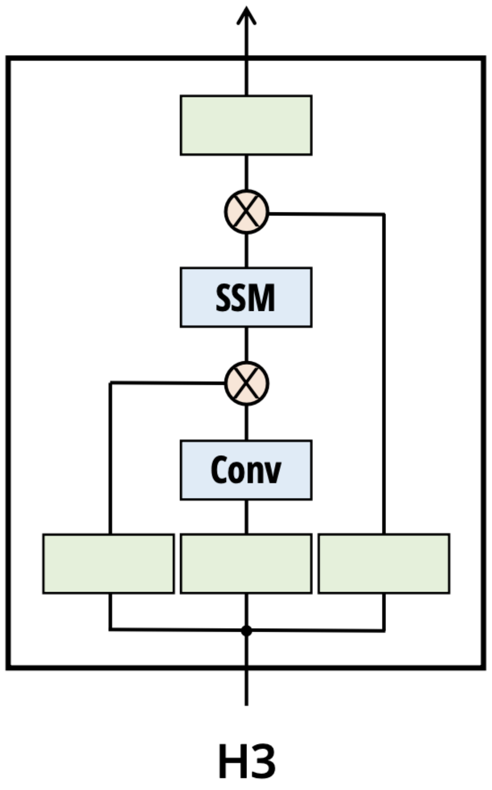
# Mamba

- Selective State Space Model (S6) – LTI to time-varying
- SSMs struggle with content-based reasoning (e.g., word remembering)
- Mamba improves SSMs with selectivity
- No attention mechanism → just Selective SSMs
- 5x faster inference than Transformers

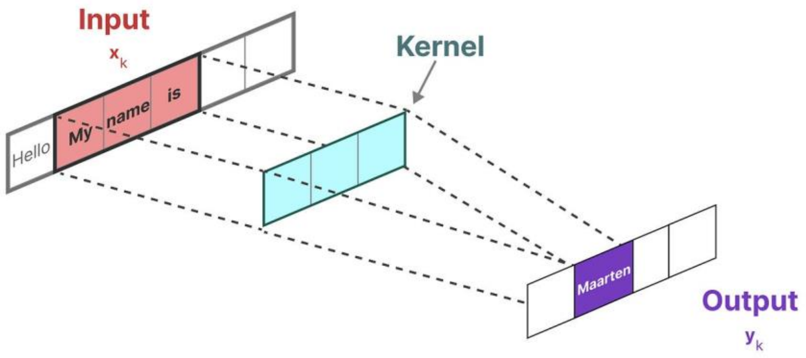
# Mamba


SSM
Conv
σ
Linear projection
Sequence transformation
Nonlinearity (activation or multiplication)
H3
⊗
Gated MLP
→
Mamba

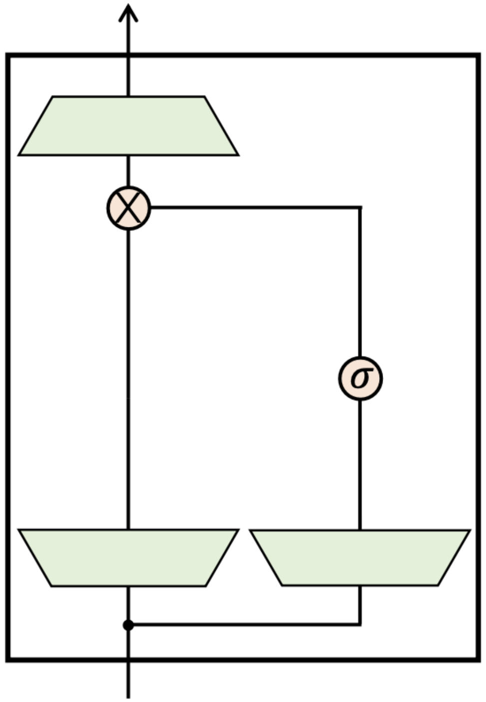
# Mamba




- Hybrid model that combines SSM and Convolution
- SSM branch: for long-range dependencies
- Conv branch: for local features
- Multiplicative interaction: outputs from both branches are combined via an element-wise operation



# Mamba

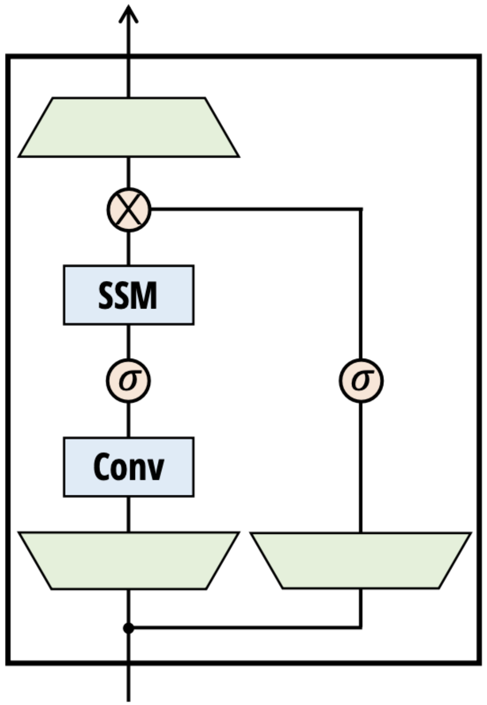



Gated MLP

-

- Gating mechanisms within an MLP structure
- Improve MLP by allowing selective information flow using multiplicative gating functions
- One of the paths is passed through a sigmoid function (creates gating signal)
- Element-wise multiplication (gating)
- Allows model to dynamically adjust which information is important for a given time step

# Mamba




→ **Mamba**

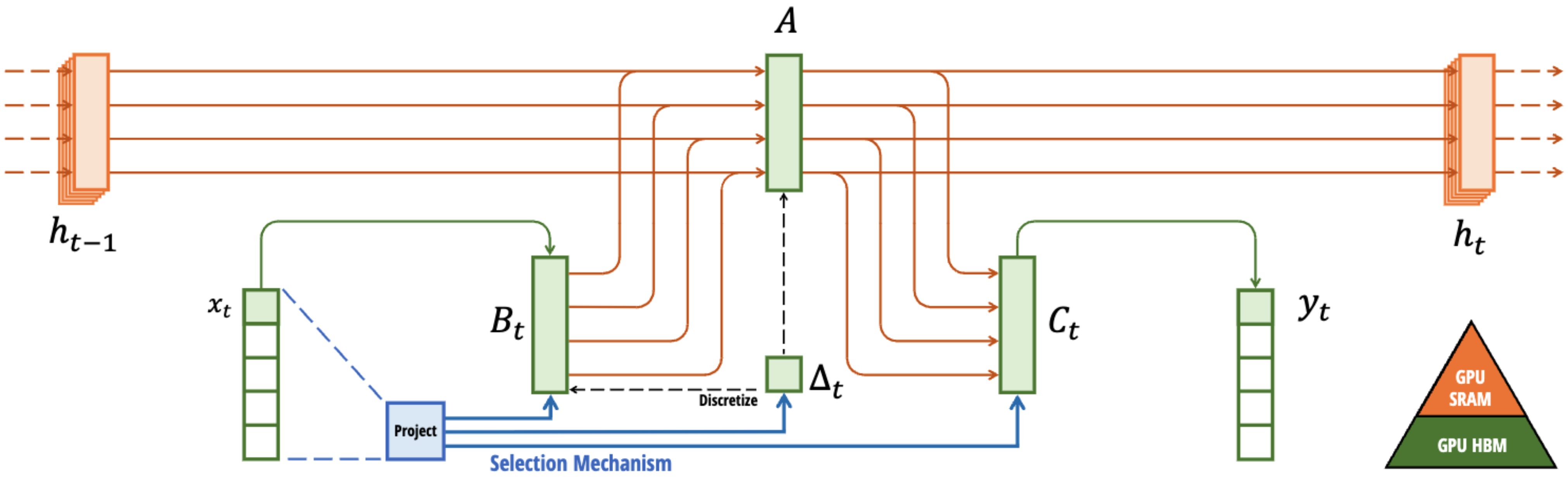
- Conv for local feature extraction (not for global)
- SSM for long-range sequence modeling
- Gating mechanism for adaptive feature selection

# Mamba

| **Algorithm 1** SSM (S4) | **Algorithm 2** SSM + Selection (S6) |
|---|---|
| **Input:** $x : (\mathtt{B,L,D})$ | **Input:** $x : (\mathtt{B,L,D})$ |
| **Output:** $y : (\mathtt{B,L,D})$ | **Output:** $y : (\mathtt{B,L,D})$ |
| 1: $\boldsymbol{A} : (\mathtt{D,N}) \leftarrow \mathsf{Parameter}$ | 1: $\boldsymbol{A} : (\mathtt{D,N}) \leftarrow \mathsf{Parameter}$ |
| ▷ Represents structured $N \times N$ matrix | ▷ Represents structured $N \times N$ matrix |
| 2: $\boldsymbol{B} : (\mathtt{D,N}) \leftarrow \mathsf{Parameter}$ | 2: $\boldsymbol{B} : (\mathtt{B,L,N}) \leftarrow s_B(x)$ |
| 3: $\boldsymbol{C} : (\mathtt{D,N}) \leftarrow \mathsf{Parameter}$ | 3: $\boldsymbol{C} : (\mathtt{B,L,N}) \leftarrow s_C(x)$ |
| 4: $\Delta : (\mathtt{D}) \leftarrow \tau_\Delta(\mathsf{Parameter})$ | 4: $\Delta : (\mathtt{B,L,D}) \leftarrow \tau_\Delta(\mathsf{Parameter} + s_\Delta(x))$ |
| 5: $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : (\mathtt{D,N}) \leftarrow \mathsf{discretize}(\Delta, \boldsymbol{A}, \boldsymbol{B})$ | 5: $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : (\mathtt{B,L,D,N}) \leftarrow \mathsf{discretize}(\Delta, \boldsymbol{A}, \boldsymbol{B})$ |
| 6: $y \leftarrow \mathsf{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C})(x)$ | 6: $y \leftarrow \mathsf{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C})(x)$ |
| ▷ Time-invariant: recurrence or convolution | ▷ Time-varying: recurrence (*scan*) only |
| 7: **return** $y$ | 7: **return** $y$ |

- Input dependent parameters
- $s_B(x_t), s_C(x_t), s_\Delta(x_t)$ are learned functions
- Varies with time since $\Delta_t, B_t, C_t$ change with $x_t$ - enables selective propagation and gating

# Mamba



$A$
$h_{t-1}$
$h_t$
$x_t$
$B_t$
$C_t$
$y_t$
$\Delta_t$
Discretize
Project
Selection Mechanism
GPU SRAM
GPU HBM

# Results

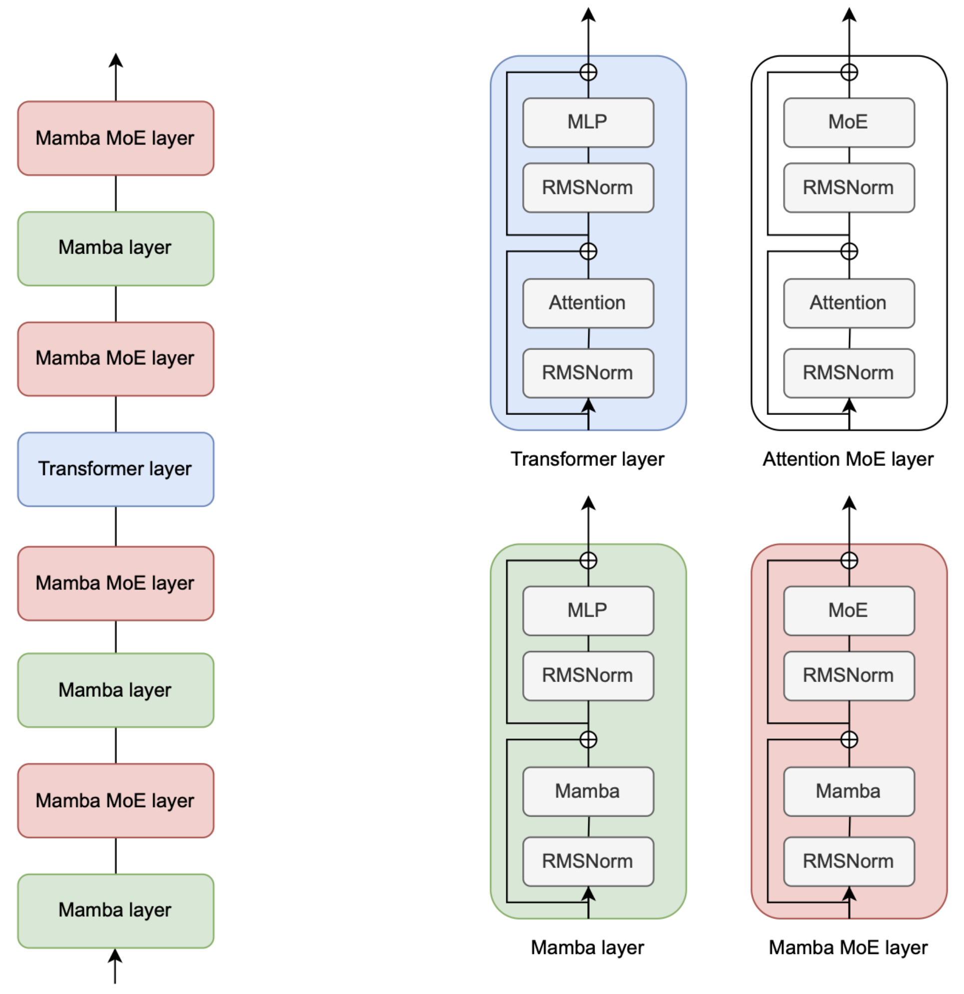
| Model | Token. | Pile PPL ↓ | LAMBADA PPL ↓ | LAMBADA ACC ↑ | HellaSwag ACC ↑ | PIQA ACC ↑ | Arc-E ACC ↑ | Arc-C ACC ↑ | WinoGrande ACC ↑ | Average ACC ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| Hybrid H3-360M | GPT2 | — | 12.58 | 48.0 | 41.5 | 68.1 | 51.4 | 24.7 | 54.1 | 48.0 |
| Pythia-410M | NeoX | 9.95 | 10.84 | 51.4 | 40.6 | 66.9 | 52.1 | 24.6 | 53.8 | 48.2 |
| **Mamba-370M** | NeoX | **8.28** | **8.14** | **55.6** | **46.5** | **69.5** | **55.1** | **28.0** | **55.3** | **50.0** |
| Pythia-1B | NeoX | 7.82 | 7.92 | 56.1 | 47.2 | 70.7 | 57.0 | 27.1 | 53.5 | 51.9 |
| **Mamba-790M** | NeoX | **7.33** | **6.02** | **62.7** | **55.1** | **72.1** | **61.2** | **29.5** | **56.1** | **57.1** |
| 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** |
| GPT-J-6B | GPT2 | – | 4.10 | 68.3 | 66.3 | 75.4 | 67.0 | 36.6 | 64.1 | 63.0 |
| OPT-6.7B | OPT | – | 4.25 | 67.7 | 67.2 | 76.3 | 65.6 | 34.9 | 65.5 | 62.9 |
| Pythia-6.9B | NeoX | 6.51 | 4.45 | 67.1 | 64.0 | 75.2 | 67.3 | 35.5 | 61.3 | 61.7 |
| RWKV-7.4B | NeoX | 6.31 | 4.38 | 67.2 | 65.5 | 76.1 | 67.8 | 37.5 | 61.0 | 62.5 |

# Conclusion

- Transition from LTI to input-dependent dynamics
- Improves ability to capture long-range and content-specific dependencies
- Mamba achieves competitive or superior performance compared to Transformers

# Research continues

- Jamba
  - Hybrid Transformer-Mamba architecture
  - Combines self-attention and SSMs
  - Reduces KV cache memory requirements up to 8x compared to Transformers




**(a)** Jamba block **(b)** Different types of layers

# Research continues

- Mamba-2
  - Structured State Space Duality (SSD)
  - SSD framework shows that attention and SSM approach can be mapped onto each other
  - Matrix transformation for GPU/TPU optimization
  - Improved memory efficiency

# Thoughts

- Efficient linear computational complexity → effective for long-sequence
- This method could help on-device models significantly
- Parameters change based on input → might be hard to train

# Open Questions

- Can this method solve “Lost in the Middle” problem?