# Manticore

CS769 Natural Language Processing

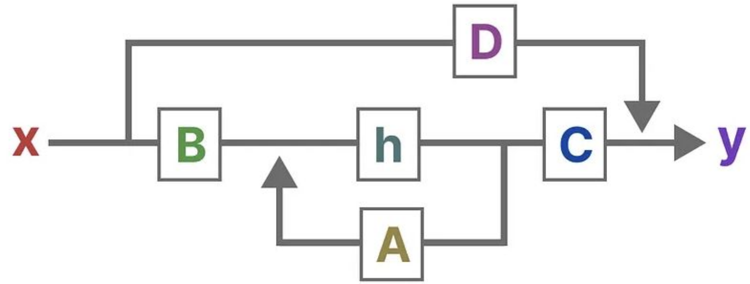
# Topics

- Motivation
- Manticore framework
- Implementation
  - Results on IMDB
  - Results on MAD tasks

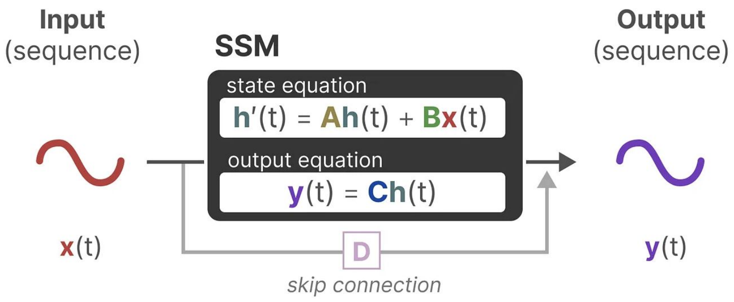
# Motivation

- Transformer models have dominated the NLP arena
  - But computational cost is quadratic in sequence length (attention layer)
  - **Transformer dominance:** Challenged by new subquadratic architectures (e.g. State Space Models, Mamba)
- State Space models
  - Mamba
- Hybrid architectures: Combine strengths of diverse models but suffer from:
  - **Manual design:** Inefficient and heuristic-driven
  - **Pre-training challenges:** Limited ability to reuse pretrained components (Neural a search)

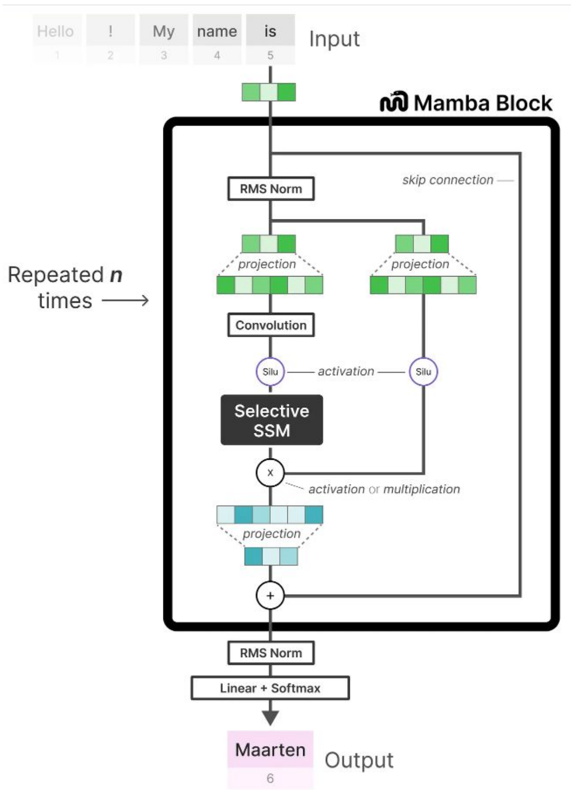
# Mamba



D
x
B
h
C
y
A

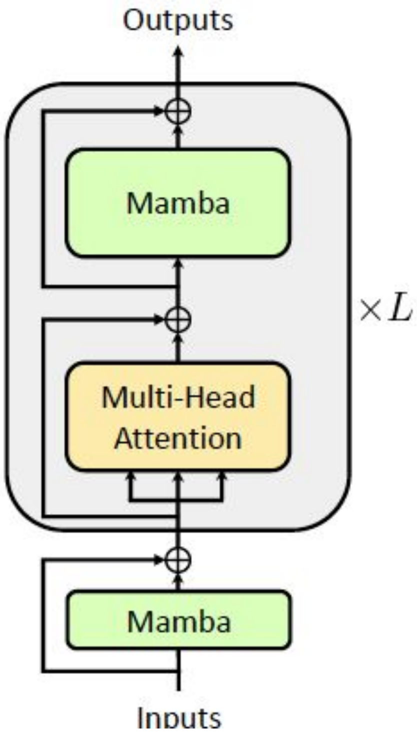



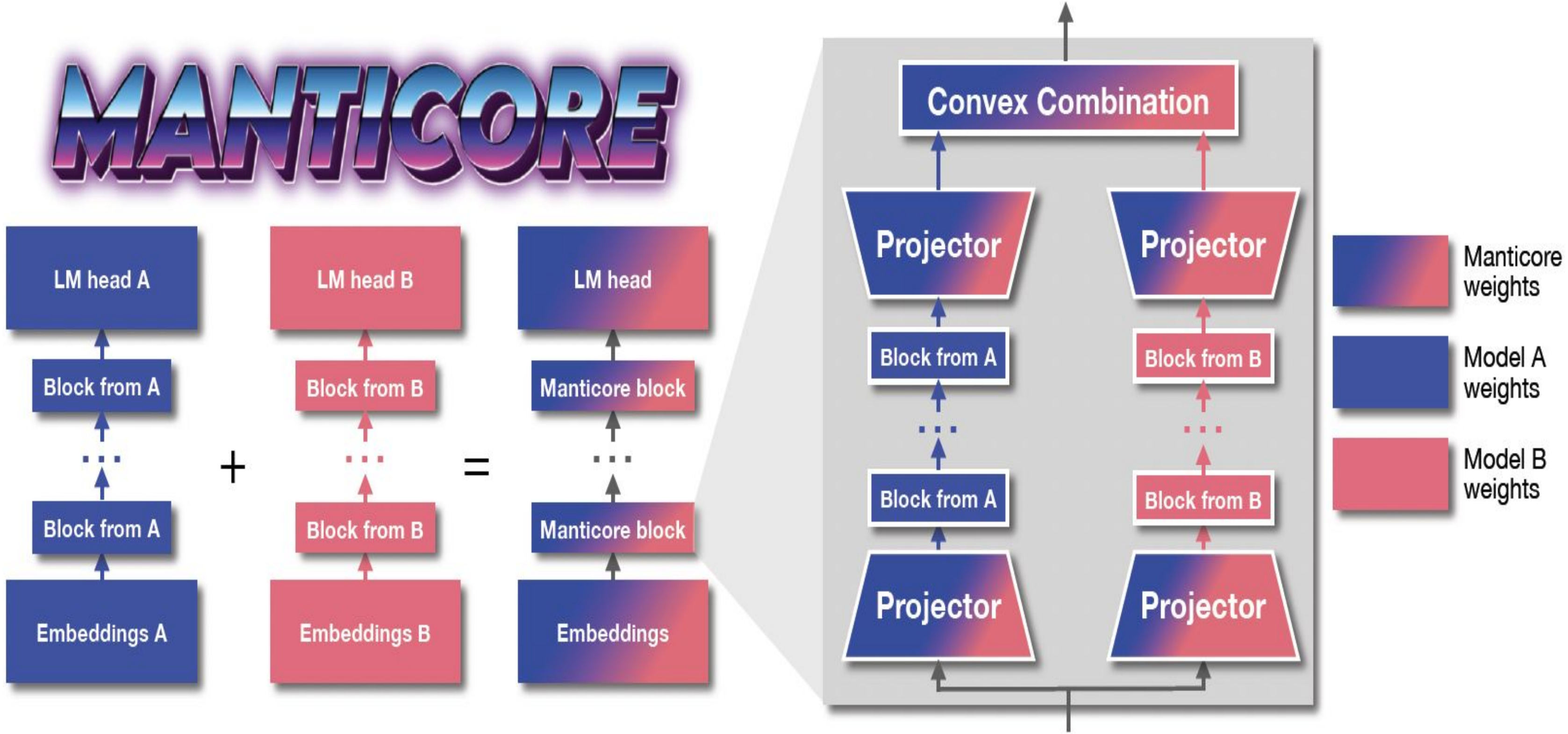
Input
(sequence)
SSM
Output
(sequence)
state equation
h′(t) = Ah(t) + Bx(t)
output equation
y(t) = Ch(t)
x(t)
y(t)
D
skip connection

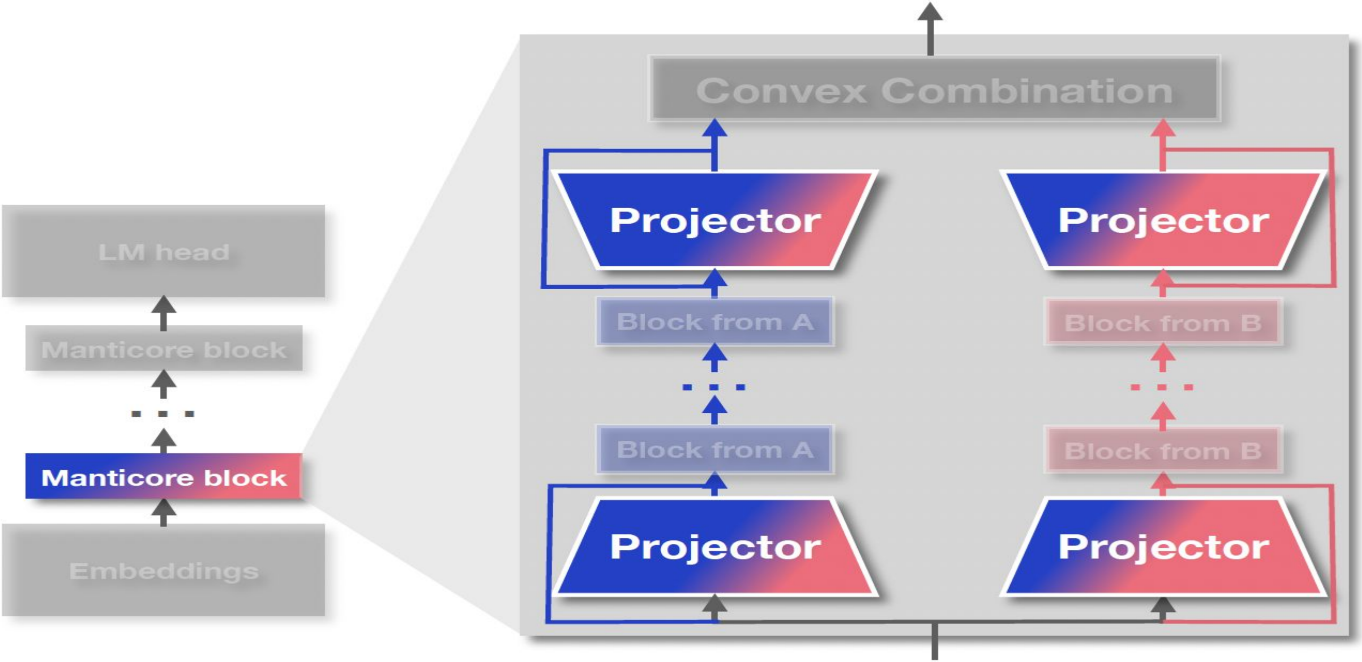
Hello
!
My
name
is
1
2
3
4
5
Input
Mamba Block
skip connection
RMS Norm
Repeated n times
projection
projection
Convolution
Silu
activation
Silu
Selective SSM
x
activation or multiplication
projection
+
RMS Norm
Linear + Softmax
Maarten
6
Output

# MambaFormer

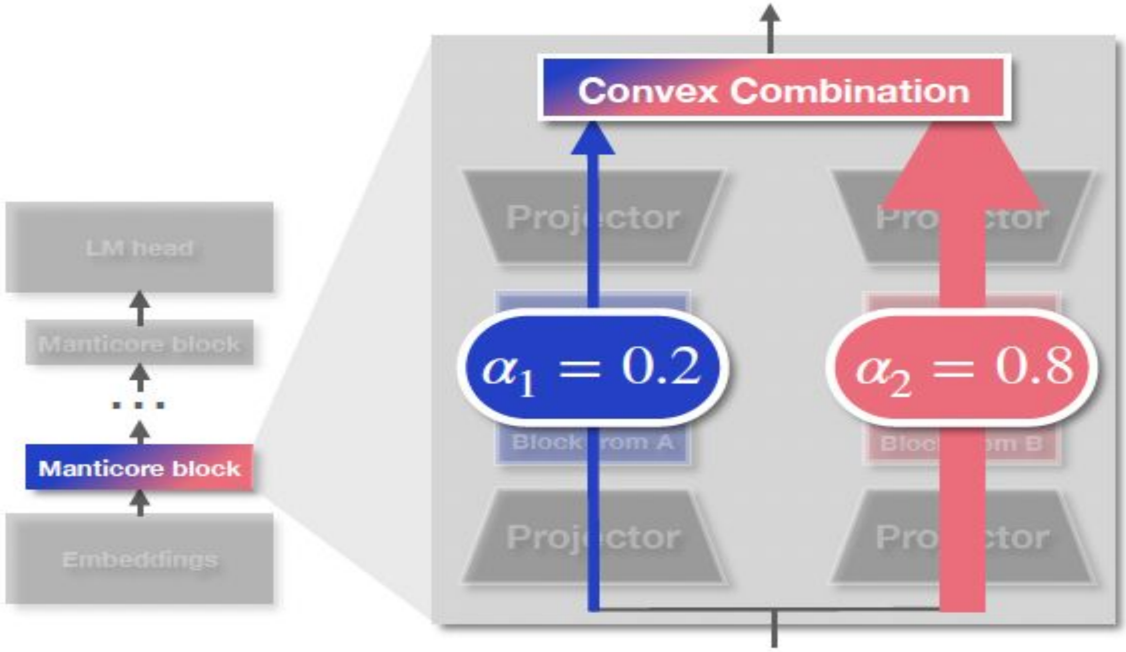



MANTICORE
LM head A
Block from A
Block from A
Embeddings A
+
LM head B
Block from B
Block from B
Embeddings B
=
LM head
Manticore block
Manticore block
Embeddings
Convex Combination
Projector
Block from A
Block from A
Projector
Projector
Block from B
Block from B
Projector
Manticore weights
Model A weights
Model B weights

# Manticore - Projectors



# Manticore - Convex combination



# Manticore - Equations

$$\text{Proj-in}^{(\ell)}_{(k)}(x;\alpha) := (1-\alpha)\cdot \text{Linear}_{d_{\max}\to d_{M_{(k)}}}(x) + \alpha \cdot \text{Trunc}(x; d_{M_{(k)}})$$

$$\text{Proj-out}^{(\ell)}_{(k)}(x;\alpha) := (1-\alpha)\cdot \text{Linear}_{d_{M_{(k)}}\to d_{\max}}(x) + \alpha \cdot \text{Pad}(x; d_{\max}).$$

$$h_k(x;\alpha_k, I_k) = \left(\text{Proj-out}^{(I_{k,n_k})}_{(k)} \circ M^{(I_{k,n_k})}_{(k)\text{Block}} \circ \ldots \circ M^{(I_{k,1})}_{(k)\text{Block}} \circ \text{Proj-in}^{(I_{k,1})}_{(k)}\right)(x;\alpha_k).$$

$$\mathbf{Mix}_{\alpha}(x; I_1, \ldots, I_K) = \sum_{k\in[K]} \alpha_k h_k(x;\alpha_k, I_k).$$

# Implementation Details

Created our own codebase from scratch with the following modular components:

- Gated Combiners (with convex combination)
- Gated Splitters
- Hybrid Model class (with task-specific output heads)
- Trainer

Next, trying to utilize DeepSpeed's ZeRO optimization or other model parallelism approaches to work with larger model backbones

# IMDB Results

With 3 epochs, 2500 iterations/ epoch, we got the following dev accuracies:

| Transformer | Mamba | Hybrid |
|---|---|---|
| 58.7 | 89.5 | 84.8 (incomplete training) |

# MAD Tasks

- Selective Copying

  input example: a c [b] t [b] [i] [i] [b] [i] | a c [b] t [b] a c [b] t

- In-context Recall

  input example: a b d e f g | a b f g

- Noisy In-context Recall

  input example: a b h d e f g | i a b f g

- Fuzzy In-context Recall

  input example: (a d) (b) (d a f) (e g) | (d a f) (e g)

- Memorization

  key-value dictionary example: {a:b, c:d, e:f}     input example: a [i] c [i] e [i] a [i] | a b c d e f a b

# MAD Task Config

- **MAD config**
  - Number of layers: 2
  - Vocab size: 16
  - Seq len: 32
  - Train examples: 4096
  - Test examples: 256
  - Hidden size: 64
  - Number of Transformer Heads: 4
  - Noise fraction: 0.2
- **Training**
  - Epochs: 20
  - Batch size: 32
  - Learning rate: 5e-4

# MAD Tasks Results

Table: Best Test Accuracy on MAD Tasks

| Task | Transformer | Mamba | Hybrid 1-blk | Hybrid 2-blk | MambaFormer |
|---|---|---|---|---|---|
| Selective Copying | 0.91 | 1.0 | 0.99 | 0.99 | 0.98 |
| In-context Recall | 0.35 | 1.0 | 1.0 | 1.0 | 0.99 |
| Noisy In-context Recall | 0.42 | 1.0 | 1.0 | 0.99 | 1.0 |
| Fuzzy In-context Recall | 0.35 | 0.36 | 0.42 | 0.43 | 0.15 |
| Memorization | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

# References


- Albert Gu, Karan Goel, and Christopher Ré. 2021. Efficiently modeling long sequences with structured state spaces. CoRR, abs/2111.00396.
- Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces.
- Jongho Park, et al. 2024. Can mamba learn how to learn? a comparative study on in-context learning tasks.
- Michael Poli,et. al. 2024. Mechanistic design and scaling of hybrid architectures.
- MohammadReza Davari and Eugene Belilovsky. 2024. Model breadcrumbs: Scaling multi-task model merging with sparse masks.
- Nicholas Roberts, et. al. 2024. Pretrained hybrids with mad skills.
- Takuya Akiba, et. al. Evolutionary optimization of model merging recipes.