Thinking Slow, Fast: Scaling Inference Compute with Distilled Reasoners

Daniele Paliotta, Junxiong Wang, Metteo Pagliardini, Kevin Y. Li, Aviv Bick, J. Zico Kolter, Albert Gu, Francois Fleuret, Tri Dao

Chanhee Lee1, Yongjun Kim2, Jiwoo Kim1
1 Department of Electrical Engineering
2 Department of Computer Science

Contents

I. Introduction

II. Background

III. Method

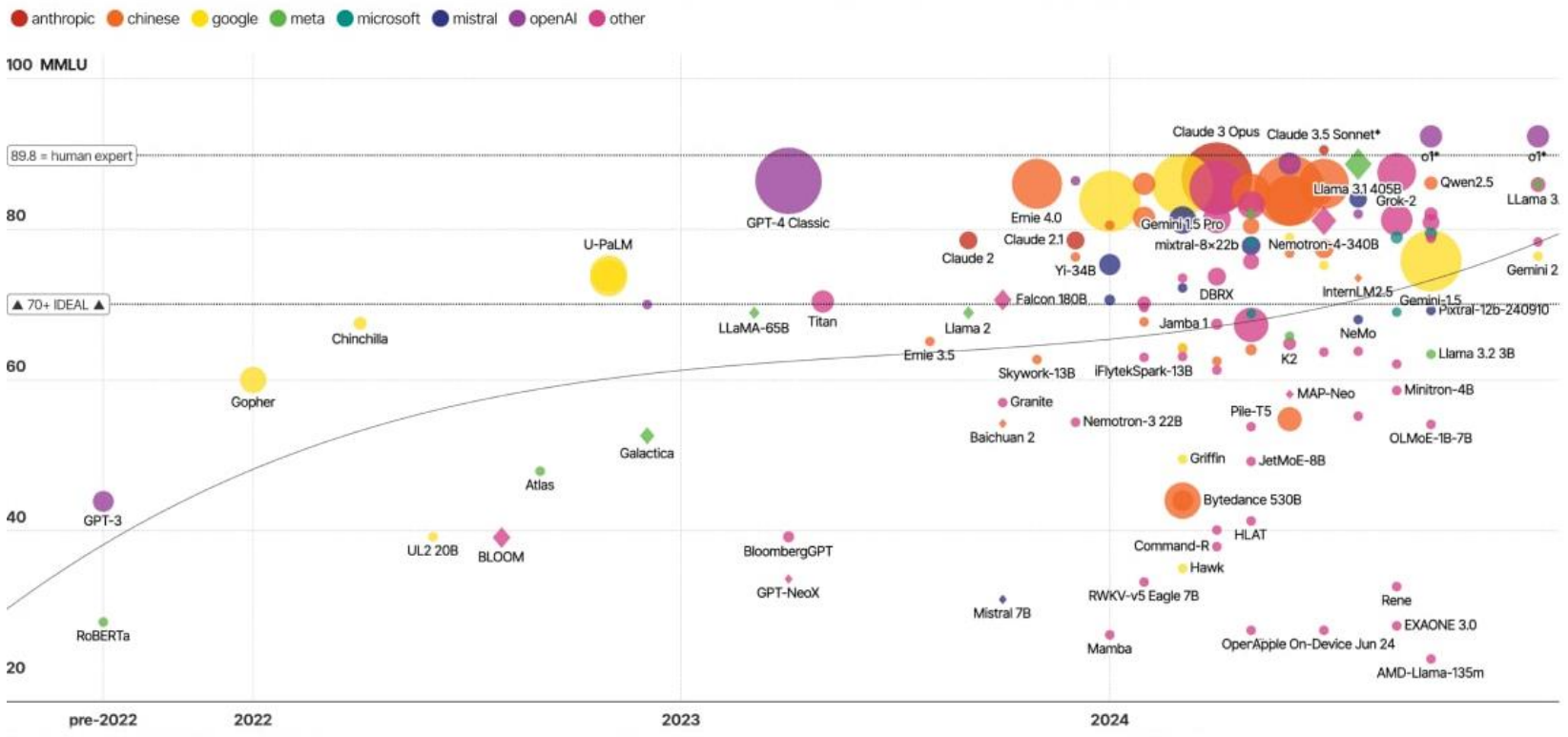
IV. Experimental Result

V. Conclusion

Introduction

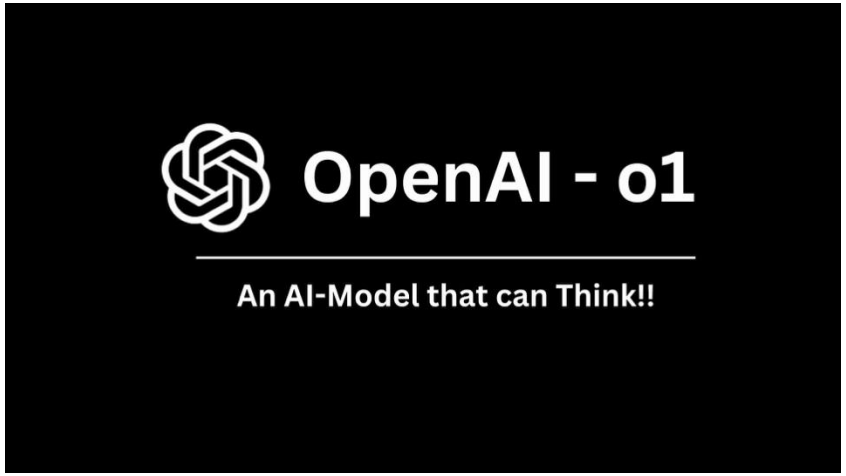
Improvement of recent LLMs inference performance

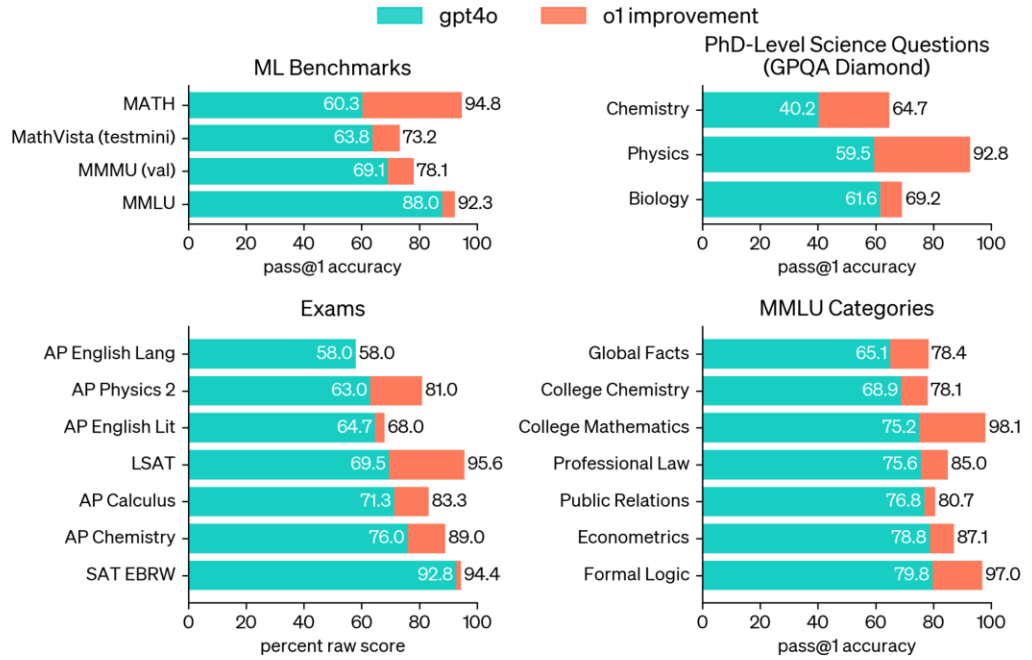
- **Problem of LLM**
  - Large Language model (LLM) gains outstanding performance in recent studies, but it has clear limitation on complex inference.
  - **Simply scaling up the model size is inefficient** from latency, memory, computational complexity perspective.



Improvement of recent LLM inference performance

- **Overcome the limitation**
  - Test-time compute is one of the solution of these problems.
  - o1 model of OpenAI has proven usefulness of **test-time compute**
  - It gives outstanding performance than original models.





[R1] GPT-o1, OpenAI, 2024

Optimal architecture for test-time compute

- **Disadvantage of test-time compute**
  - In transformer architecture, **High memory usage and slow inference,** Due to their long sequence length.
  - This is especially noticeable in transformer structures where a lot of cost is required in the inference step.

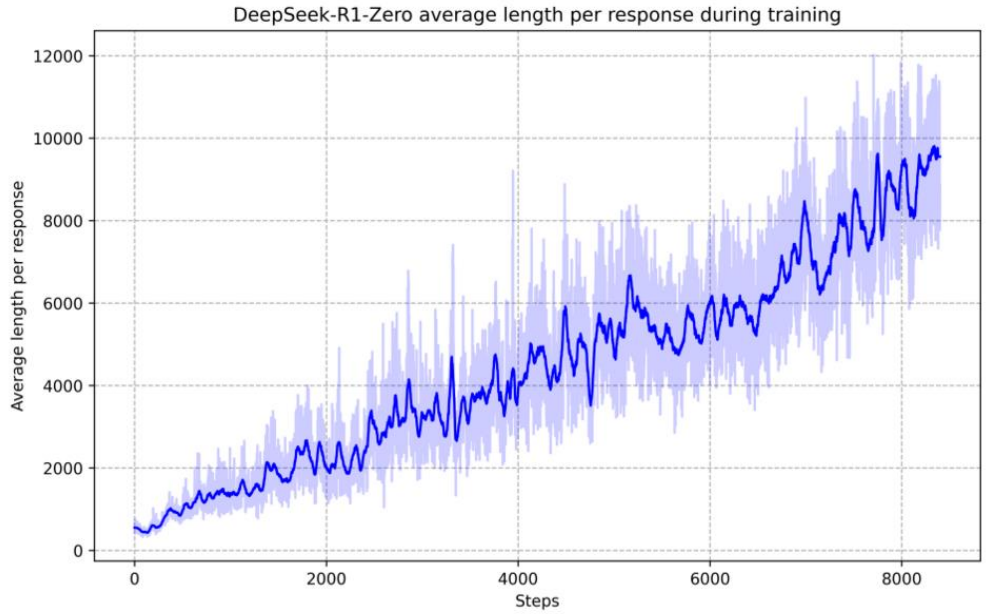
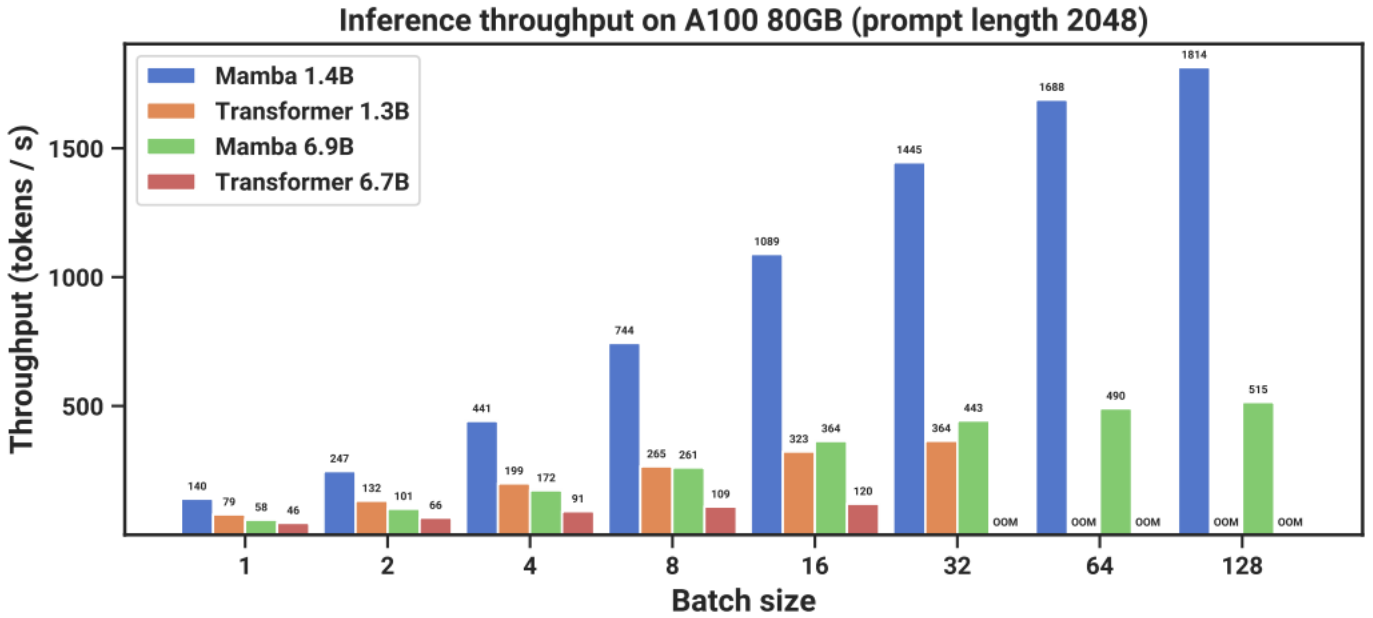


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

Optimal architecture for test-time compute

- **Subquadratic models**
  - **Faster, less memory usage** than transformer, which is foundation model of LLMs
  - **Inefficient well-trained subquadratic models**. → **distillation**
  - Mamba is well-known subquadratic model.



What the author do?

- **Problems and solutions summary**
  - Test time compute needs long context → Use subquadratic model
  - Lack of well-trained subquadratic model → Distill transformer to subquadratic model

Background

Chain of Thought (CoT)

- **Step-by-step inference**
  - Not compute the result directly, but implement the flow of thought
  - Show good performance some specific tasks
    - Mathematical problem
    - Logical determination
    - Analysis of complex sentence

Chain of Thought (CoT)

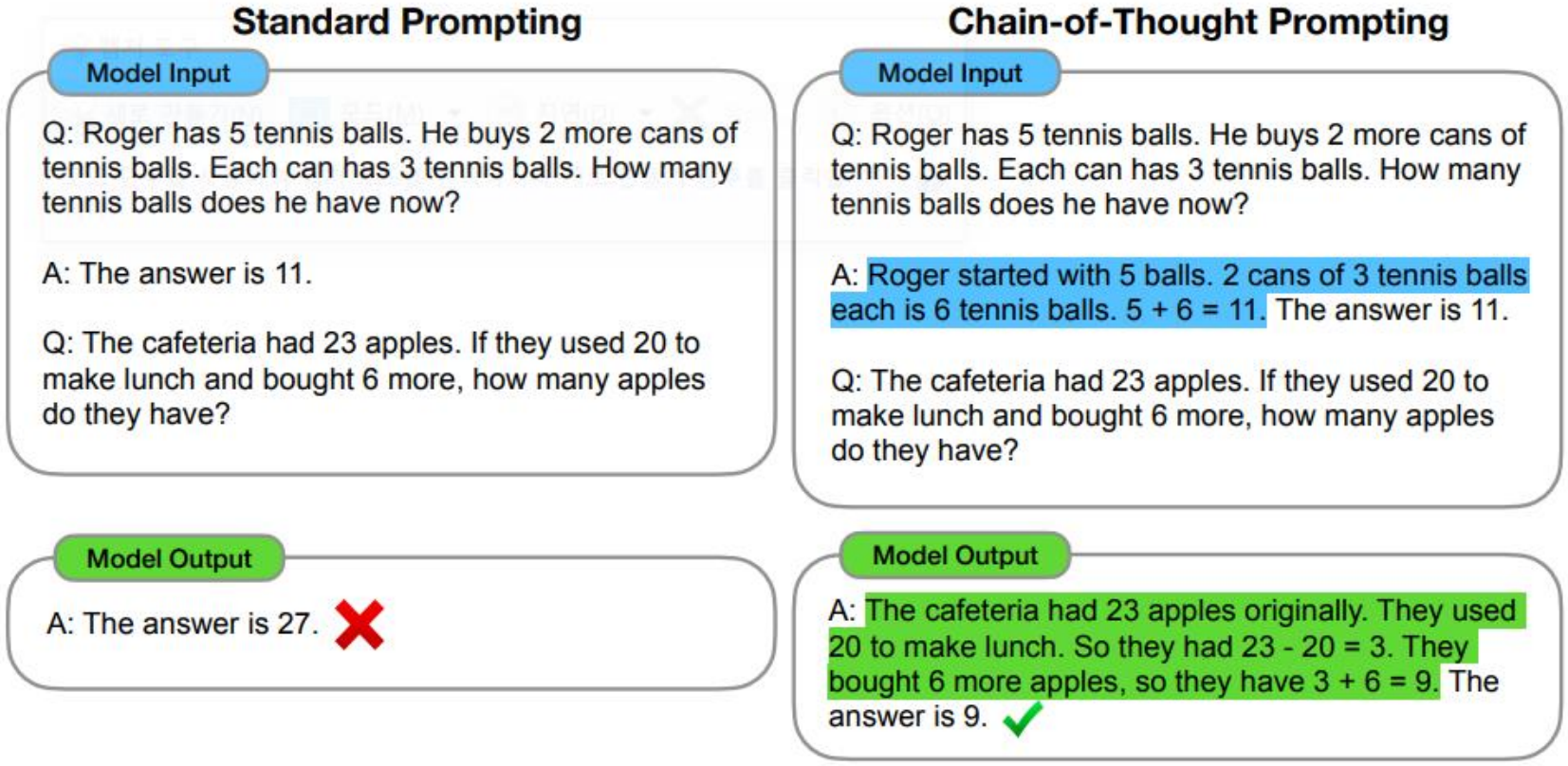


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

[R1] Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." *Advances in neural information processing systems* 35 (2022): 24824-24837.

Chain of Thought (CoT)

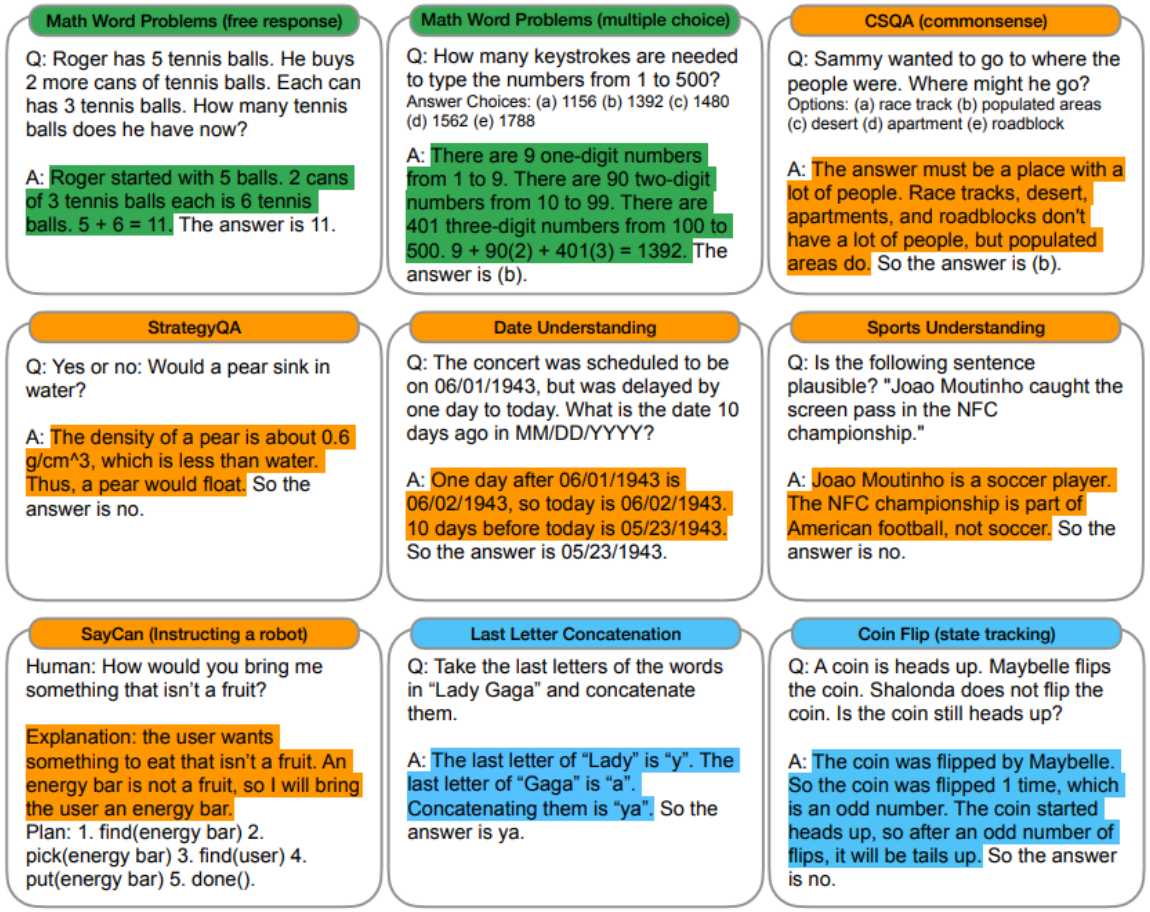


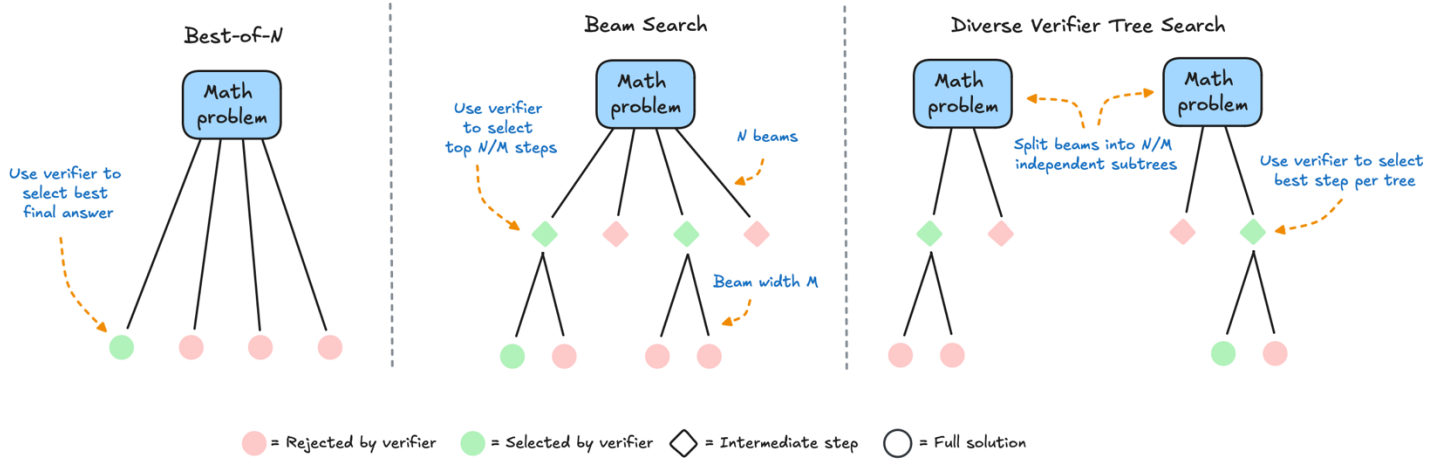
Figure 3: Examples of ⟨input, chain of thought, output⟩ triples for arithmetic, commonsense, and symbolic reasoning benchmarks. Chains of thought are highlighted. Full prompts in Appendix G.

- **Various prompt example**
  - Arithmetic (green)
  - Common sense (orange)
  - Symbolic reasoning (blue)
- **Features**
  - Step-by-steps response
  - Generates longer sequence than original cases

[R1] Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." *Advances in neural information processing systems* 35 (2022): 24824-24837.

Test time compute with CoT

- **Using more resource in inference step**
  - Generate different result of CoTs
  - Select proper response various stratage
    - majority voting
    - Using learnable reward models for each CoTs
- **Disadvantage**
  - Generate longer CoT sequences than normal cases
    → **Require high cost and memory usage**



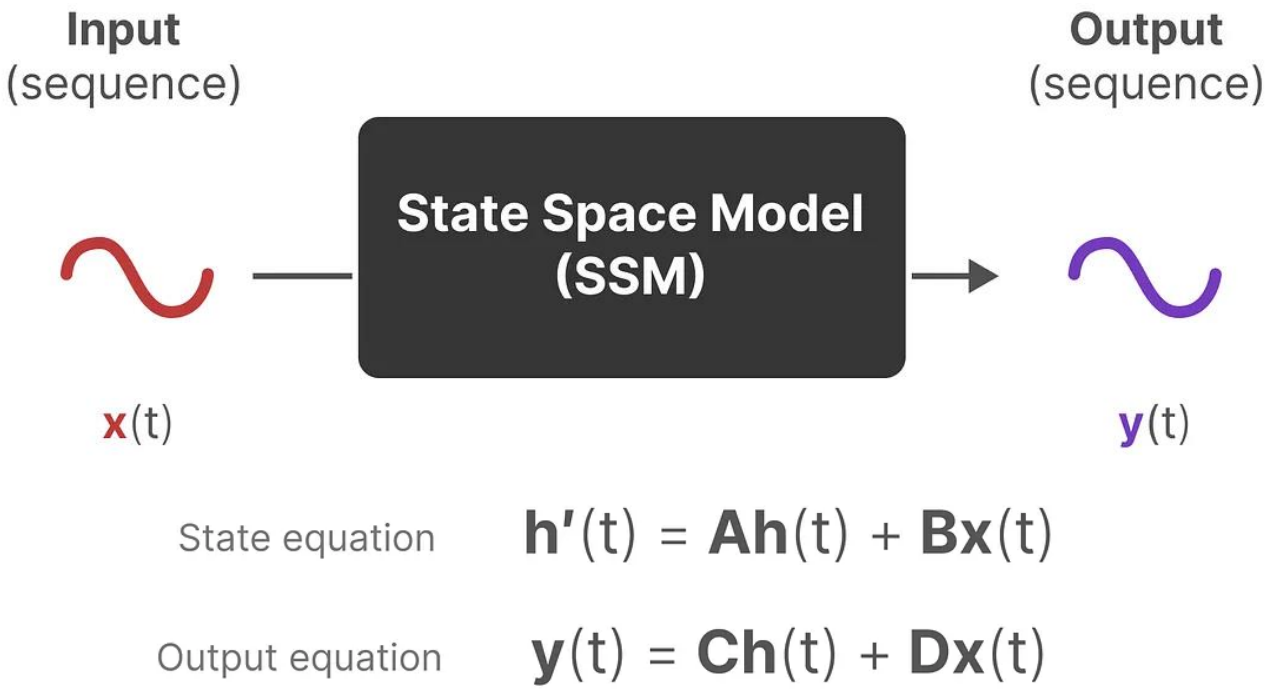
Subquadratic model

- **Transformer**
  - Self-attention mechanism, which is main reason of performance of Transformer architecture, refer all tokens in each steps.
  - Its computational complexity is $O(L^2)$ for input sequence of length L.
- **Subquadratic computational complexity**
  - Less than $O(L^2)$ computational complexity for input sequence of length L

  → **Better than transformer to adopt Test-time compute**

Subquadratic model

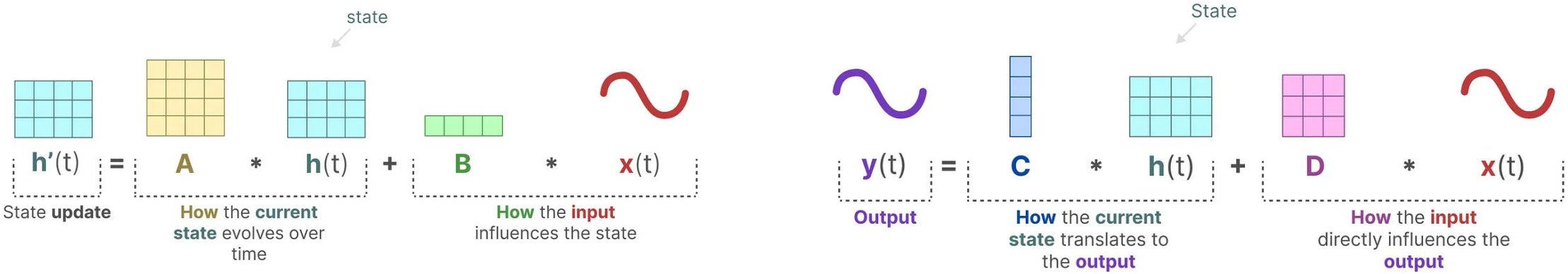
- **Mamba [R2]**
  - One of the highlighted subquadratic architecture of LLM
  - State Space Model (SSM) based recurrence model

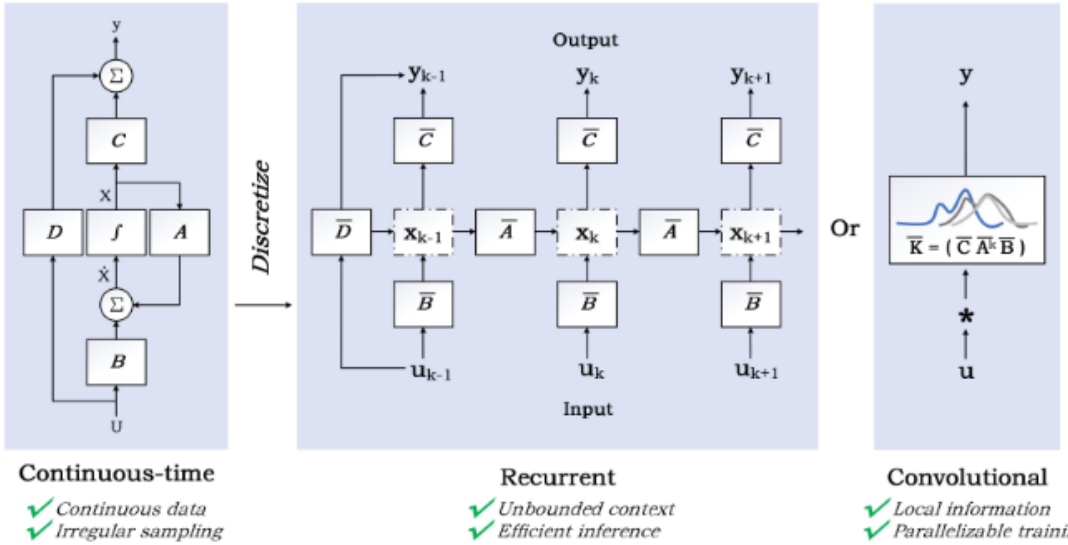


[R2] Gu, Albert, and Tri Dao. "Mamba: Linear-time sequence modeling with selective state spaces." *arXiv preprint arXiv:2312.00752* (2023).

SSM block of Mamba

- **Update state vector and discretize**





[R2] Gu, Albert, and Tri Dao. "Mamba: Linear-time sequence modeling with selective state spaces." *arXiv preprint arXiv:2312.00752* (2023).

SSM block of Mamba

- **Remove Linear Time-Invariant (LTI) constraint to SSM**
  - Matrix A contributes major performance of SSM, but it has LTI constraints.
  - Using High-order Polynomial Projection Operators (HiPPO)
  - Construct recent token more precisely than first token.

HiPPO Matrix A_{nk}

$$\begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & \text{everything below the diagonal} \\ n+1 & \text{the diagonal} \\ 0 & \text{everything above the diagonal} \end{cases}$$

HiPPO Matrix

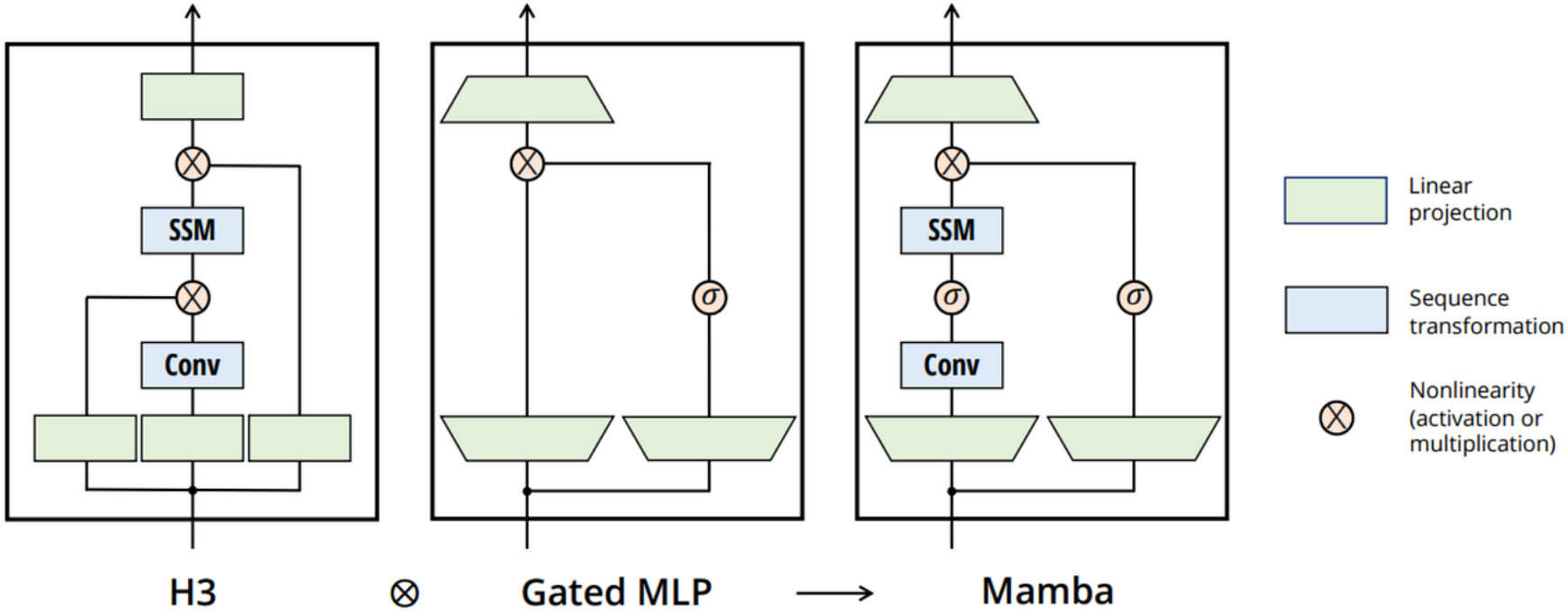
1	0	0	0
1	2	0	0
1	3	3	0
1	3	5	4

k (rows) × n (columns)

[R2] Gu, Albert, and Tri Dao. "Mamba: Linear-time sequence modeling with selective state spaces." *arXiv preprint arXiv:2312.00752* (2023).

SSM block of Mamba

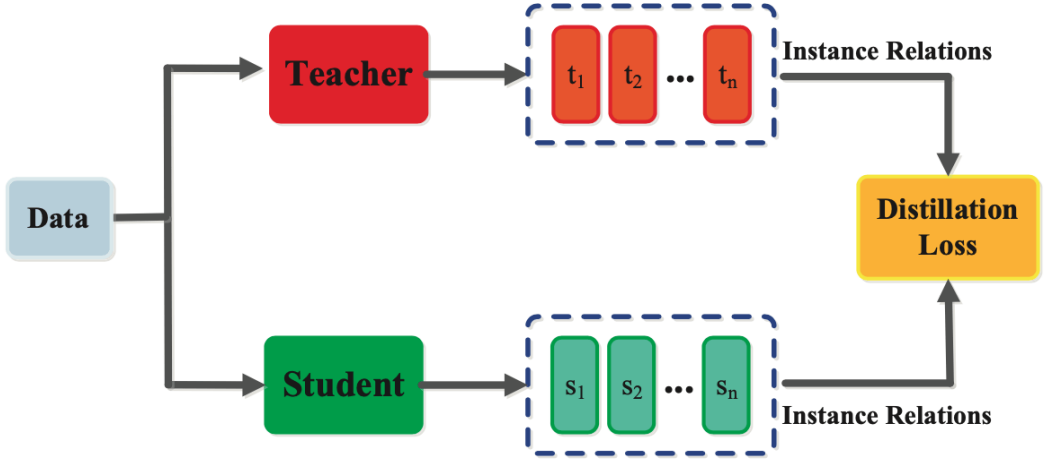
- **Selectiveness**
  - Fusion of H3 and Gated MLP



[R2] Gu, Albert, and Tri Dao. "Mamba: Linear-time sequence modeling with selective state spaces." *arXiv preprint arXiv:2312.00752* (2023).

Distillation

- **Definition**
  - Transfer the knowledge large model to small model.
  - Efficient compression of LLMs
- **Cross-architecture distillation**
  - Transformer to RNN[R3], Linear Attention[R4], SSM[R5] etc.
  - In this paper, the author distill the knowledge transformer to mamba.

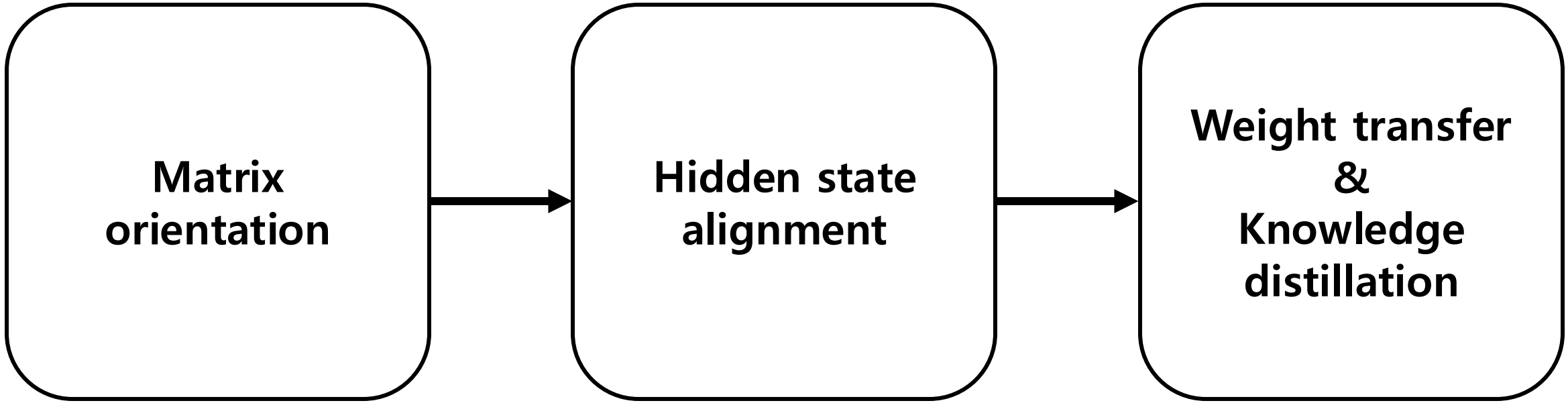


[R3] Kasai, Jungo, et al. "Finetuning pretrained transformers into rnns." *arXiv preprint arXiv:2103.13076* (2021).
[R4] Zhang, Michael, et al. "The Hedgehog & the Porcupine: Expressive Linear Attentions with Softmax Mimicry." *ICLR, 2024*
[R5] Wang, J., Paliotta, D., May, A., Rush, A., and Dao, T. "The mamba in the llama: Distilling and accelerating hybrid models." *NeurIPS, 2025*

Method

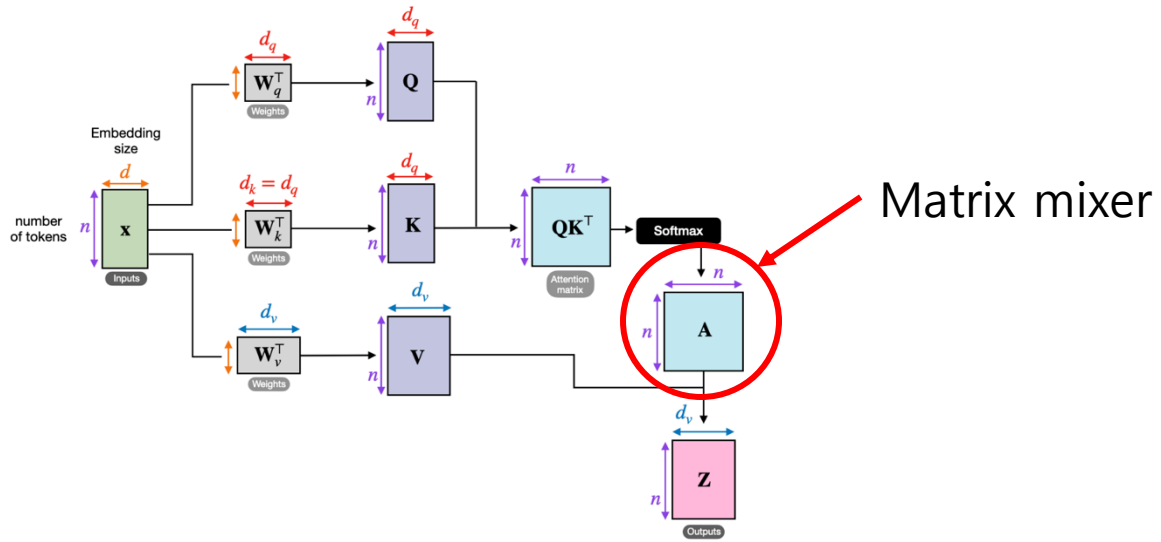
MOHAWK: Distill Llama to Mamba

- **MOHAWK is composed of three stages:**



Sequence transformation & matrix mixer

- **Definition 1: Sequence transformation**
  - Sequence transformation refers to parameterized map on some sequence
  - i.e. sequence transformation combine tokens at various time steps
- **Definition 2: Matrix mixer**
  - Some sequence transformation can be represented by $\mathrm{Y} = \mathrm{MX}$
  - In this case, M is called as **matrix mixer**
  - In a attention mechanism, $\mathrm{Softmax}(\mathrm{QK}^{\mathrm{T}})$ is a matrix mixer



Causal variants of sequence transformation

- **Definition 3: Causal variants of sequence transformation**
  - By multiplying a lower triangular matrix filled with 1s (**L**), we can get casual variants of sequence transformation. That is, if we multiply L to the attention matrix, we can get causal variants of attention matrix.

$$\mathbf{L} \circ \mathrm{Softmax}(\mathbf{QK}^\top) \longrightarrow \begin{bmatrix} \alpha_1 & 0 & 0 & \cdots & 0 \\ \alpha_{2:1} & \alpha_2 & 0 & \cdots & 0 \\ \alpha_{3:1} & \alpha_{3:2} & \alpha_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{n:1} & \alpha_{n:2} & \alpha_{n:3} & \cdots & \alpha_n \end{bmatrix}$$

Relation between Mamba-2 & causal linear attention

- **Mamba-2 is a time-varying state-space model, and defined as follow:**

$$h_{t+1} = \mathbf{A}_t h_t + \mathbf{B}_t x_t$$
$$y_t = \mathbf{C}_t h_t$$

- **Fixing $A_t = I$ results in the formulation of causal linear attention with the matrices B, C representing the projections of the key and the query, respectively.**

Why Mamba-2 can be the causal linear attention?

- **Mamba-2 is a time-varying state-space model, and defined as follow:**

$$h_t = a_t\, h_{t-1} + B_t\, x_t,$$
$$y_t = C_t^\top\, h_t.$$

- **If we unroll the state update with $h_0 = 0$, we can get …**

$$h_t = a_t\, h_{t-1} + B_t x_t = \sum_{i=1}^{t} \left(\prod_{j=i+1}^{t} a_j \right) B_i\, x_i.$$

Same function with causal masking

Why Mamba-2 can be the causal linear attention?

- **In linear attention, by using kernel-mapping, the output is calculated as follow :**

$$y_t = \phi(q_t)^\top \sum_{i=1}^{t} \phi(k_i)\, v_i$$

- **By computing $y_t = C_t^\top h_t$ with previously unrolled results, we can get**

$$y_t = C_t^\top h_t = \sum_{i=1}^{t} \left[C_t^\top (B_i\, x_i) \right] \prod_{j=i+1}^{t} a_j$$

Why Mamba-2 can be the causal linear attention?

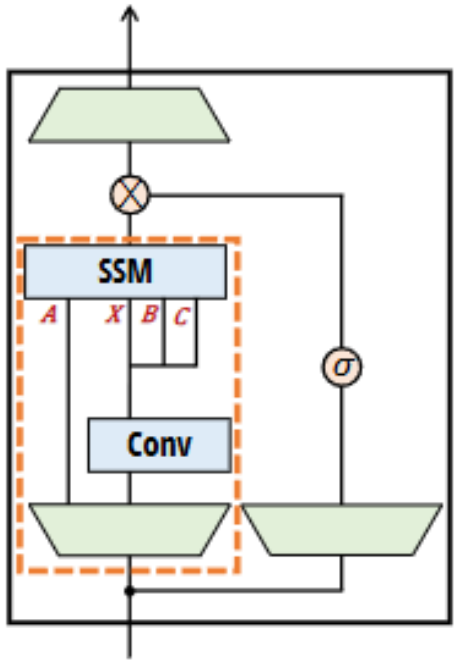
- **In summarize, attention mechanism's query, key, value matrices can be mapped into Mamba-2 SSM as below:**

Attention mechanism	Mamba-2 SSM
$kernelized\ key_i = \phi(W_k x_i)$	$B_i x_i$
$value_i = W_v x_i$	(Intrinsically included in B_i)
$kernelized\ query_t = \phi(W_q x_t)$	C_t

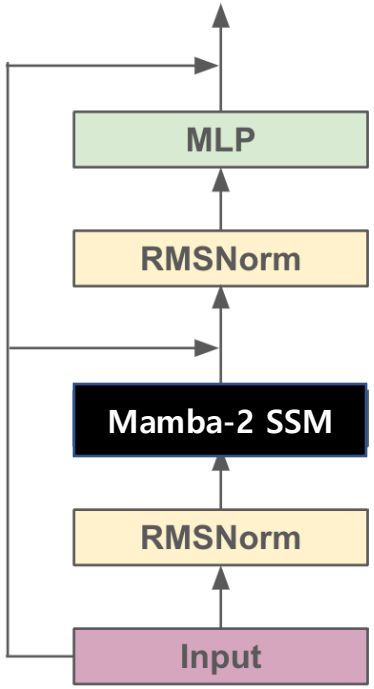
Please refer to Katharopoulos et al. & Bick et al. for more detailed explanation

Distillation Llama into Llamba

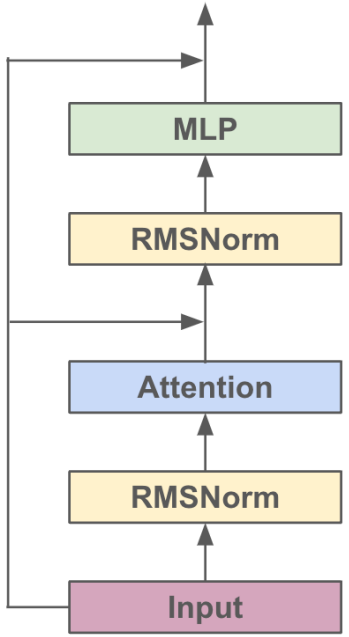
- **Goal: Distillation Llama to Llamba, which means Llama model replacing its attention to Mamba-2 SSMs.**



Mamba-2



Student: Llamba



Teacher: Llama

MOHAWK Stage 1: Matrix Orientation

- **Goal: Align each student SSM block's SSM to its teacher's attention matrix.**

$$M_\ell^{(T)} = \mathrm{softmax}(Q_\ell K_\ell^\top) \qquad y_t = C_t^\top \sum_{i=1}^{t} \Big(\prod_{j=i+1}^{t} A_j \Big) B_i \, x_i = \sum_{i=1}^{t} \underbrace{C_t^\top \Big(\prod_{j=i+1}^{t} A_j \Big) B_i}_{M_{t,i}^{(S)}} \, x_i.$$

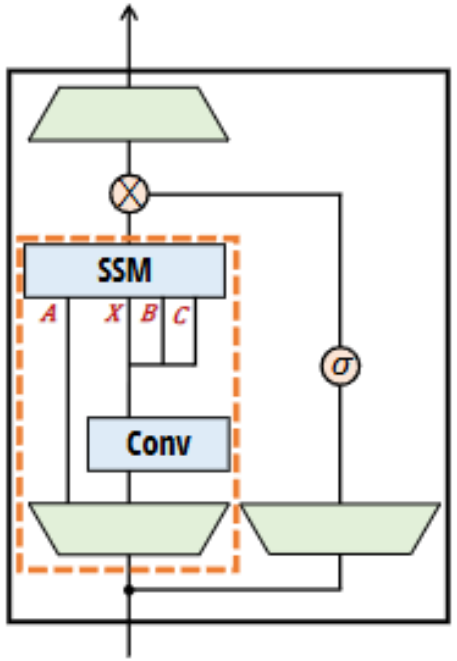
Loss of stage 1: $\mathcal{L}_{\text{matrix}} = \| M_\ell^{(T)}(u_\ell) - M_\ell^{(S)}(u_\ell) \|_F^2$

- **Key insight: Matching mixing matrices first, to ensure the student mirrors the teacher's long-range range information flow before any hidden-state alignment**

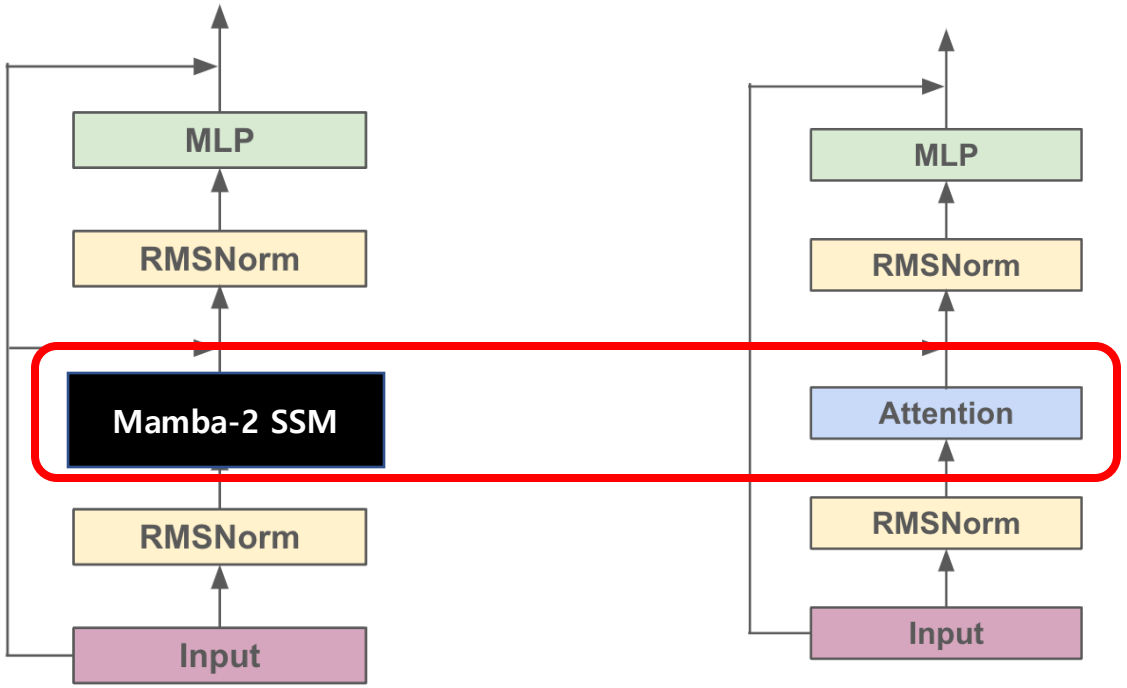
Distillation Llama into Llamba

- **Goal: Align each student SSM block's SSM to its teacher's attention matrix.**

Loss of stage 1: $\mathcal{L}_{\text{matrix}} = \left\| M_\ell^{(T)}(u_\ell) - M_\ell^{(S)}(u_\ell) \right\|_F^2$



Mamba-2



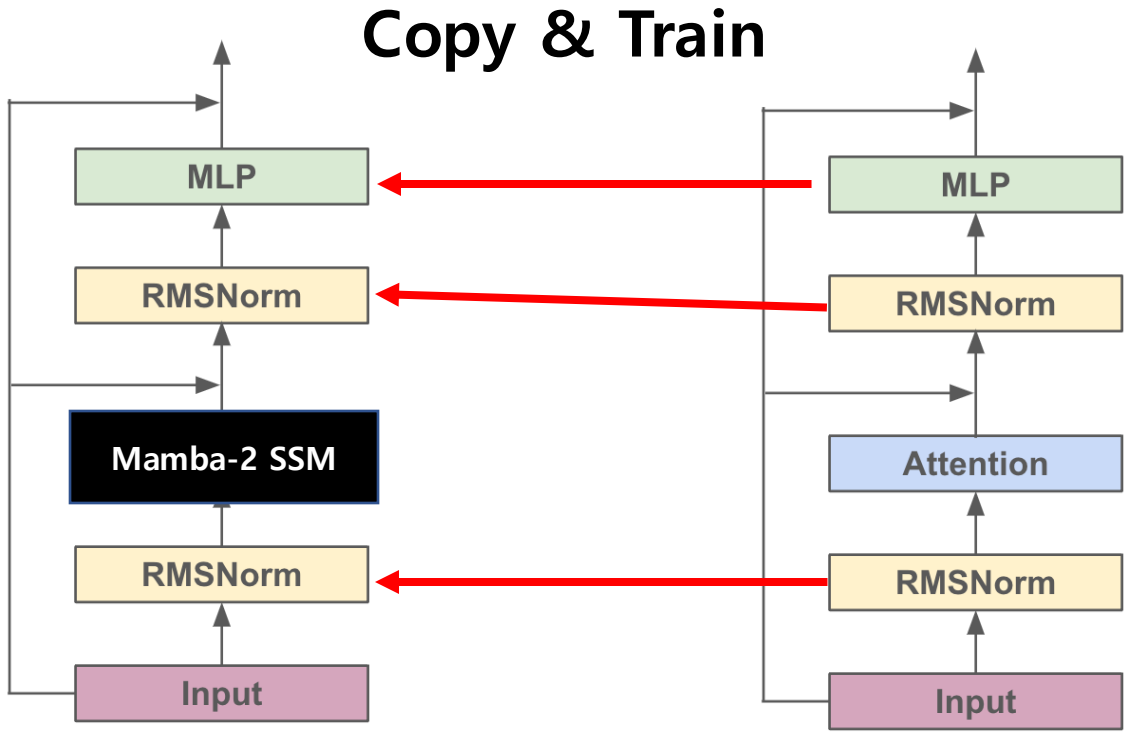
Student: Llamba

Teacher: Llama

MOHAWK Stage 2: Hidden-State Alignment

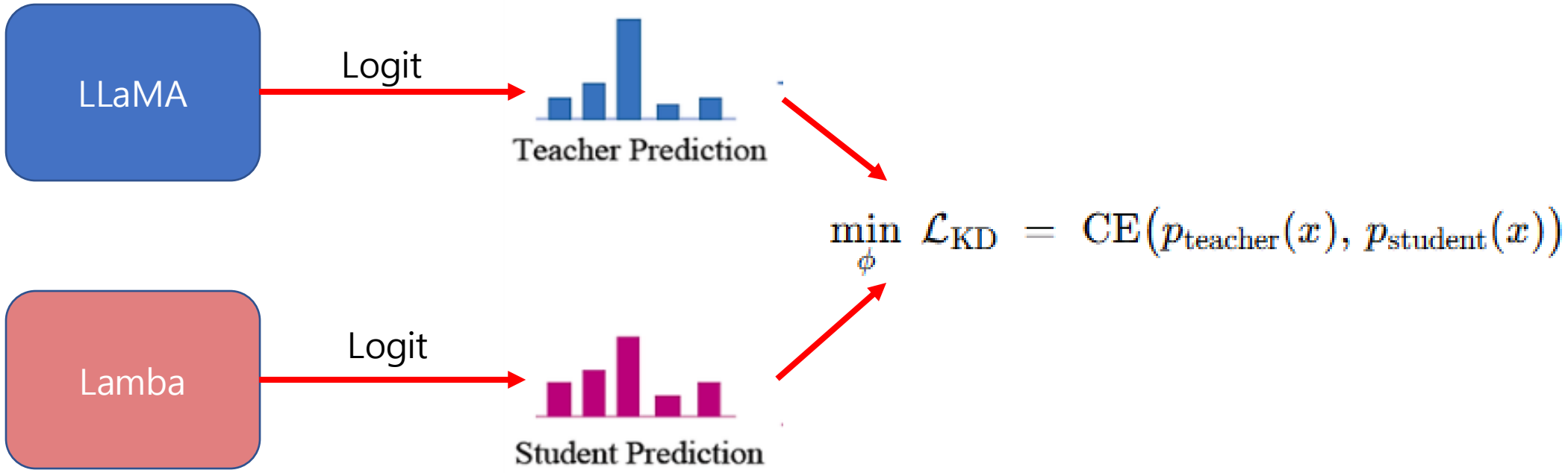
- **Goal: Bring the student's internal representations in each mixer block into close agreement with the teacher's output of attention block.**

$$\min_{\phi} \|\mathrm{AttnBlock}(\mathbf{u}) - \mathrm{StudentMixerBlock}_{\phi}(\mathbf{u})\|_2$$



MOHAWK Stage 3: Weight transfer & End-to-End KD

- **Goal: Finalize the student by**
- **(a) inheriting compatible teacher weights**
- **(b) training on final outputs so that its predictions match the teacher's over a small corpus.**



Experimental Results

Experimental Results Overview

- **Authors tried to show..**

  (1) Inference speedup of distilled models is better!

  (2) This speedup can result in better scaling for a given inference time budget!

Inference Time Results

- **Experiment protocols**
  - Dataset: MATH and GSM8K
    - Realistic setup that matches the prompt and CoTs length
  - Varying batch size, tasks of generating 512 tokens from a prompt with 512 tokens
  - Prefilling time is not included.
    - It depends on the given prompt
    - Only interested in the time to generate multiple completions given one prompt
  - Done on a single NVIDIA H100 GPU

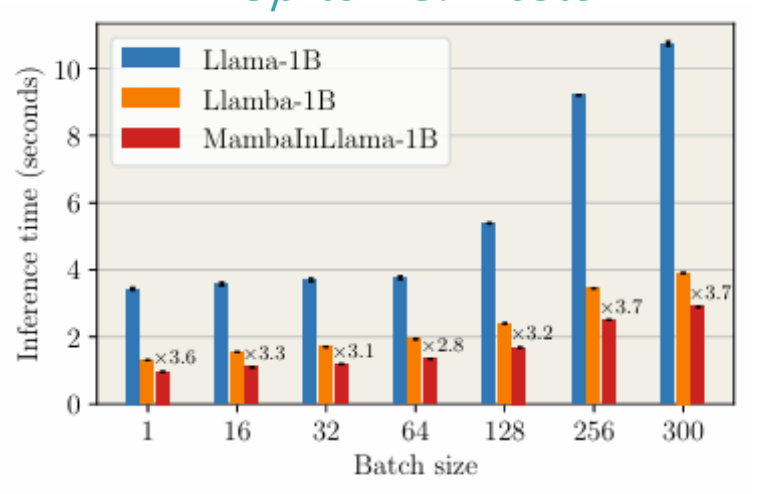
MATH System Prompt: *Solve the following math problem efficiently and clearly:\n\n- For simple problems (2 steps or fewer):\nProvide a concise solution with minimal explanation.\n\n- For complex problems (3 steps or more):\nUse this step-by-step format:\n\n## Step 1: [Concise description]\n[Brief explanation and calculations]\n\n## Step 2: [Concise description]\n[Brief explanation and calculations]\n\n...\n\nRegardless of the approach, always conclude with:\n\nTherefore, the final answer is: $\boxed{answer}$. I hope it is correct.\n\nWhere [answer] is just the final number or expression that solves the problem.*

GSM8K System Prompt: *\n\nGiven the following problem, reason and give a final answer to the problem.\nYour response should end with "The final answer is [answer]" where [answer] is the response to the problem.\nProblem:*

Inference Time Results

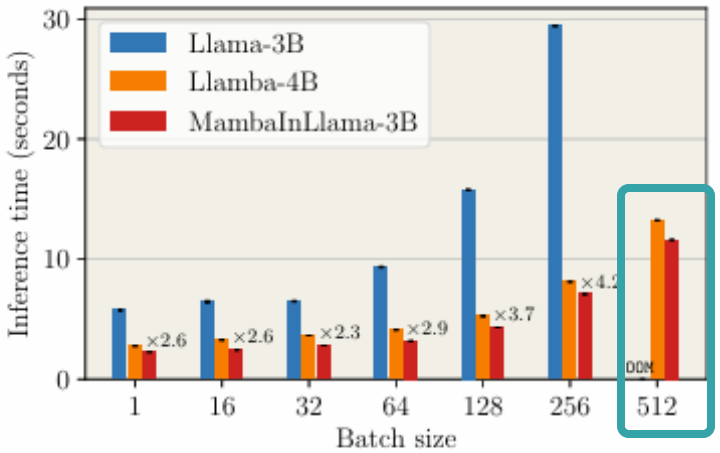
- **Faster generation of distilled models**
  - Distilled models were faster than Llama baselines.
  - MambaInLlama models are slightly faster than Llamba
    - Smaller SSM state size: MambaInLlama(16) < Llamba(64)

Up to ×3.7 faster



(a) 1B Models.

Up to ×4.2 faster



(b) 3B Models.

Out-of-memory error on baseline

Results on Reasoning Tasks

- **Experiment Protocol**
  - Teacher models: Llama-3.2-1B-Instruct & 3B-Instruct
  - Distilled Mamba Students: MambaInLlama-1B & 3B, Llamaba-1B & 4B
  - 500 sample subset of MATH and GSM8K
  - Evaluated coverage and accuracy

Results on Reasoning Tasks

- **Experiment Protocol**
  - Coverage vs Accuracy
    - Coverage: Probability that the generated set contains the correct answer(upper bound)
    - Accuracy: Probability that the selected answer is correct (final output quality)

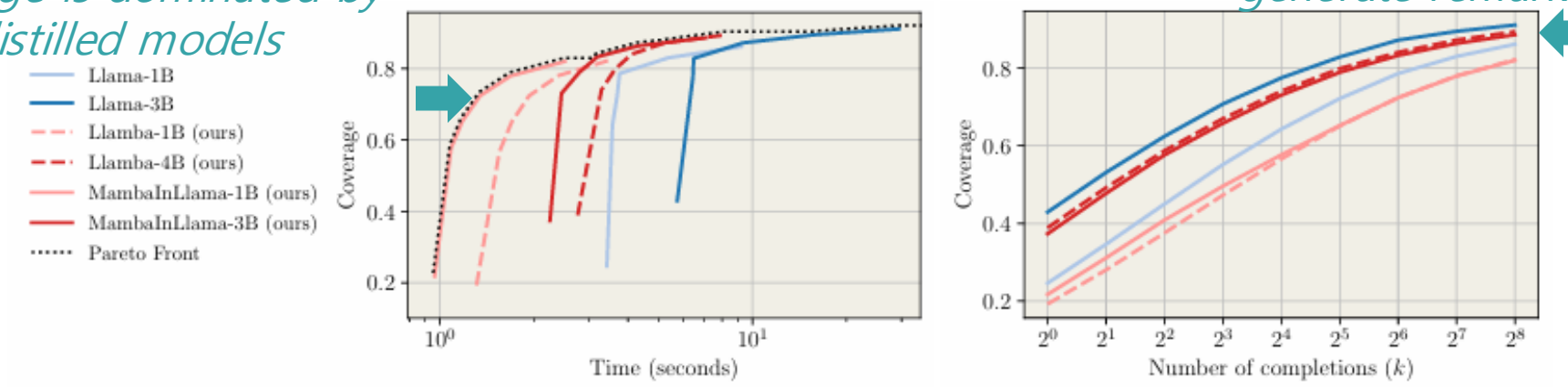
Coverage: $$\text{pass@k} = \frac{1}{\#\text{ of problems}} \sum_{i=1}^{\#\text{ of problems}} \left(1 - \frac{\binom{N-C_i}{k}}{\binom{N}{k}}\right)$$

Results on Reasoning Tasks

- **Distilled Models can Cover like Teachers**
  - "Does distilled model can generate meaningful results?"
    - Previous distilled models: could not achieve similar coverage with teachers…☹
  - ** Time budget: faster models can generate more candidates*
  - Observed scaling of coverage as:
    - (a) a function of time budget, (b) the number of generation k increases

Generate correct answer fast
∴ Overall Pareto front for coverage is dominated by distilled models

Similar coverage with teachers
∴ Distilled model has an ability to generate remarkable candidates



(a) Scaling with time.

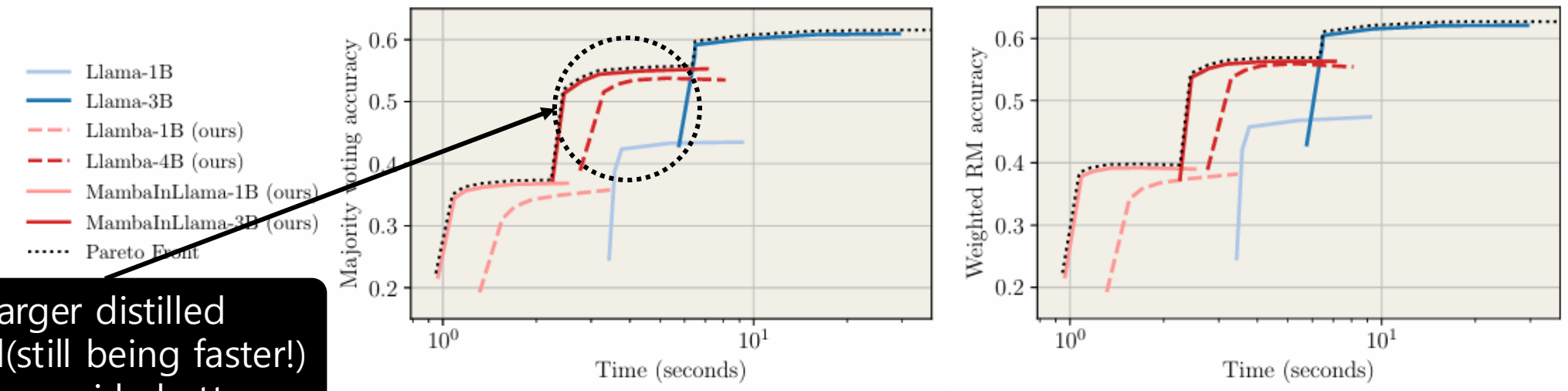
(b) Scaling with number of completions.

Results on Reasoning Tasks

- **Distilled Models Achieve Competitive Accuracy Under Fixed Time**
  - Selection methods
    - **Majority voting**: Selects the most frequently occurring final answer
    - **Weighted Best-of-N**: Selects the answer with the highest score based on a reward model or evaluation function. (In this paper, Llama-3.1-8B-based reward model)

Small time budget: Distilled models dominate

Large time budget: Teacher models dominate



(a) Majority voting.

(b) Weighted Best-of-N.

Larger distilled model(still being faster!) can provide better accuracy

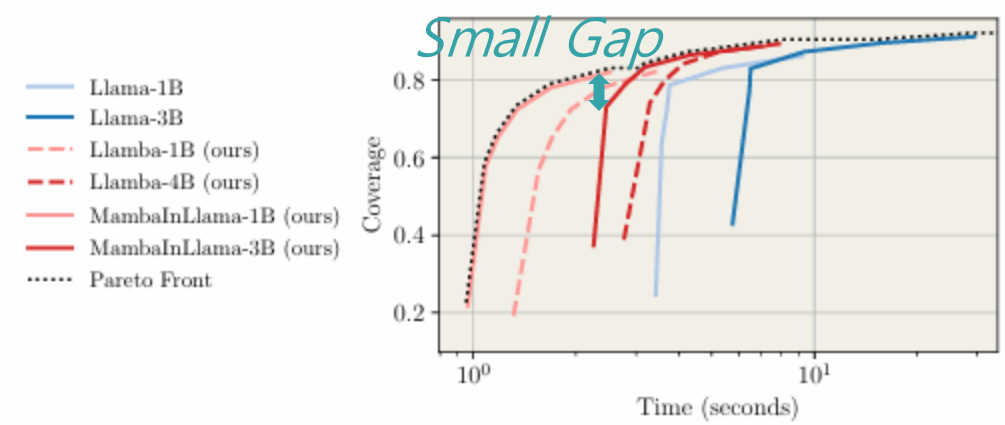
Result Analysis

- **Larger Students are Better than Smaller Teachers**
  - 3B Subquadratic models are faster than 1B Transformer models
  - Proposed MambaInLlama-3B and Llamba-4B model outperforms Llama-1B baseline, in terms of coverage and accuracy while being faster

Result Analysis

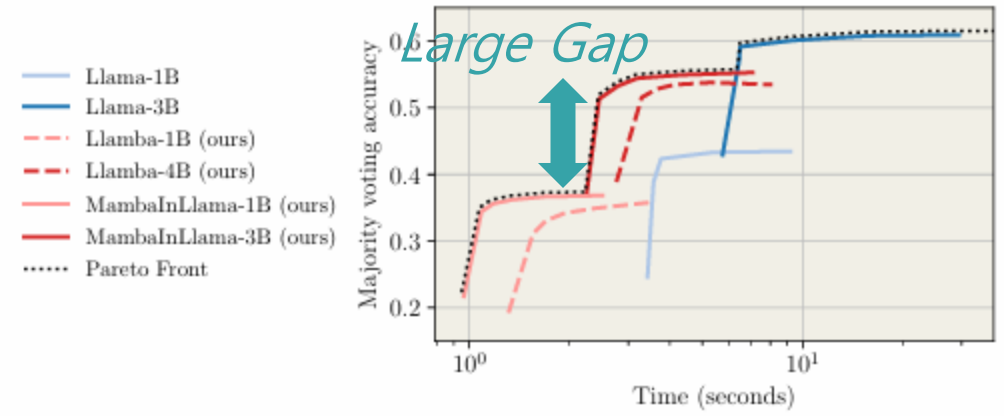
- **Smaller Models have possibility to achieve better accuracy**
  - Smaller models have great coverage, but gap between larger in accuracy test.
  - Smaller models have the ability to generate the correct answer → Developing better reward model can be helpful
  - For the tasks that coverage matters most(e.g. easily verifiable coding, mathematical proofs), smaller models can be preferred

MambaInLlama 1B vs 3B
Small Gap



MATH coverage

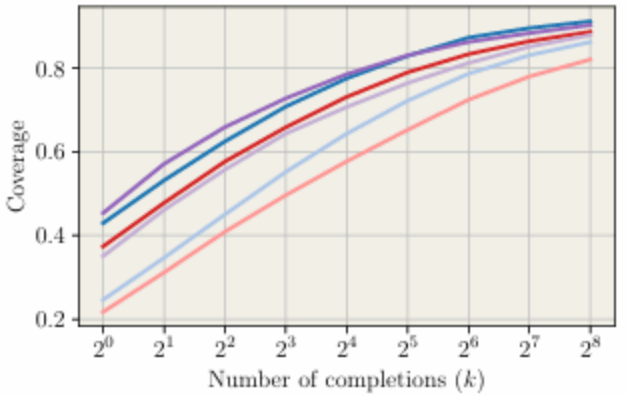
MambaInLlama 1B vs 3B
Large Gap



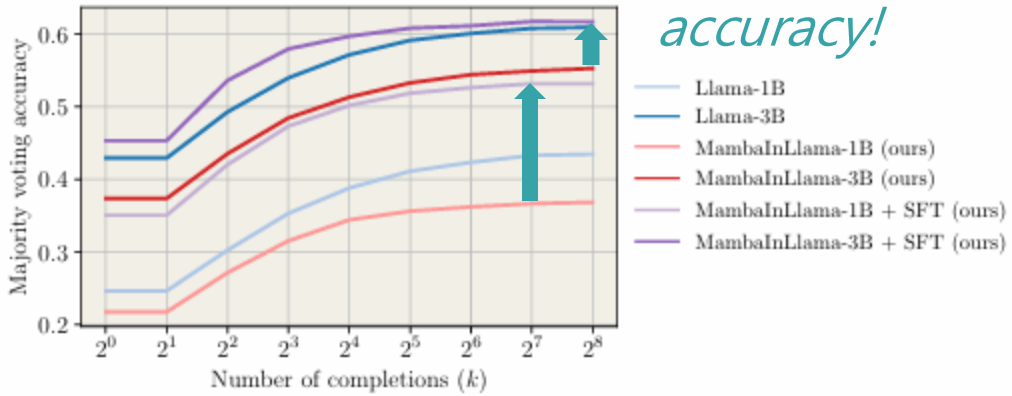
MATH accuracy

Supervised Fine-Tuning

- **SFT improves the models significantly**
  - While distillation effectively transfers knowledge from the teacher model, SFT further refines and aligns the model's capabilities
  - With SFT, subquadratic architectures can surpass teachers!



(a) Coverage.



(b) Majority voting.

High improvement in accuracy!

Conclusion

Conclusion

- Investigated whether lower-complexity models can leverage their superior generation throughput to outperform similarly sized Transformers
- Focused on reasoning tasks to scale test-time computation
- In fixed memory and computation source, proposed models achieve better coverage and accuracy for most time budgets compared to Transformers
- The paper highlights the potential of Mamba and other attention alternatives

Limitation and Future Research

- **Lack of proposed method**
  - Propose new distilling method, Develop better reward model, …
- **Need to compare with existing KD models**
  - What is the exact limitation of existing models?
  - What is the difference between the existing models and proposed models?
- **Need to analyze the reason of improvement**
  - Why the Mamba model is superior?
  - Why was the proposed model able to achieve this performance?

Thank You!

Chanhee Lee, Yongjun Kim, Jiwoo Kim