TAD
The Center for AI & Data Science
Tel Aviv University

Massachusetts Institute of Technology

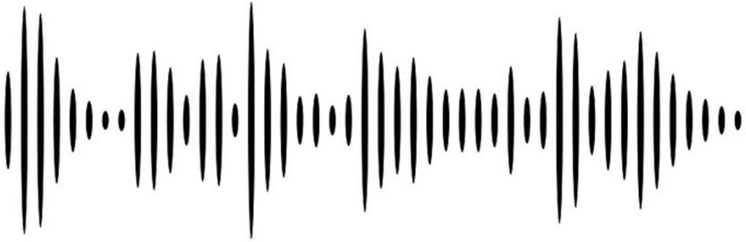
IBM

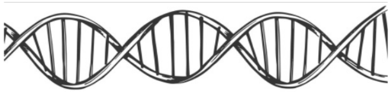
COLM

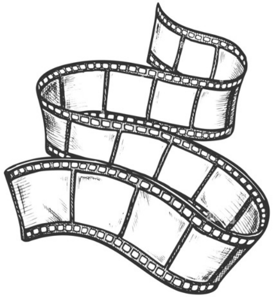
Overflow Prevention Enhances Long-Context Recurrent Models

Assaf Ben-Kish, Itamar Zimerman, Jehanzeb Mirza,
James Glass, Leonid Karlinsky, Raja Giryes

Long Sequences Are All Around Us







Transformers:

- Short Sequences - SOTA
- Long Sequences - Limited due to quadratic complexity w.r.t input length

The Long-Sequence Architecture Zoo

Sub-Quadratic Recurrent Models

State-Space Models

DeltaNet

RWKV

Recurrent Gemma

...

Transformer-Based Solutions

Flash Attention

Native Sparse Attention

YaRN

...

Strong Recurrent LLMs Are Starting to Appear

Model Name	ARC-25	HellaSwag-10	MMLU-5	Winogrande-5	TruthfulQA-0	GSM8K-5	Average
RWKV models							
RWKV-v6-Finch-7B*	43.86	75.19	41.69	68.27	42.19	19.64	48.47
RWKV-v6-Finch-14B*	47.44	78.86	52.33	71.27	45.45	38.06	55.57
Transformer models							
Falcon2-11B	59.73	82.91	58.37	78.30	52.56	53.83	**64.28**
Meta-llama-3-8B	60.24	82.23	**66.70**	78.45	42.93	45.19	62.62
Meta-llama-3.1-8B	58.53	82.13	66.43	74.35	44.29	47.92	62.28
Mistral-7B-v0.1	59.98	**83.31**	64.16	78.37	42.15	37.83	60.97
Mistral-Nemo-Base-2407 (12B)	57.94	82.82	64.43	73.72	49.14	**55.27**	63.89
Gemma-7B	61.09	82.20	64.56	79.01	44.79	50.87	63.75
Hybrid SSM-attention models							
RecurrentGemma-9b**	52.00	80.40	60.50	73.60	38.60	42.60	57.95
Zyphra/Zamba-7B-v1*	56.14	82.23	58.11	**79.87**	52.88	30.78	60.00
Pure SSM models							
TRI-ML/mamba-7b-rw*	51.25	80.85	33.41	71.11	32.08	4.70	45.52
FalconMamba-7B (pre-decay)*	49.23	80.25	57.27	70.88	37.28	21.83	57.29
FalconMamba-7B*	**62.03**	80.82	62.11	73.64	**53.42**	52.54	64.09
Avg Length:	55.3 tokens	86.7 tokens	80 tokens	27.9 tokens	71.4 tokens	156.7 tokens	

[1] Falcon Mamba: The First Competitive Attention-free 7B Language Model; Zuo et. al 2024; TII

What about long contexts?

Falcon-Mamba-Inst-7B over long-context benchmarks - HotPotQA:

1. Baseline - Process the full context
2. Random - Process only a random chunk from the context

Method	0-4K	4K-8K	8K+
Baseline	36.35	21.18	18.4
Random	35.53	23.02	27.62

➤ Context Length = ~10,000 Tokens; Chunk Size = 3000 Tokens

What about long contexts?

❌ SoTA Recurrent models have yet to close the performance gap with Transformers in long-context tasks

This is despite having “good conditions” for long-context processing:

- ✓ Match SoTA Transformers on **<u>short-sequence</u>** tasks
- ✓ Parameter count is large enough for in-context learning (7 Billion)
- ✓ Large hidden states
- ✓ Trained on long sequences (some on 32K!)

Problem Identification

Problem Investigation

Recurrent Memory Capacity
Are recurrent LLMs able to store, and later retrieve, all the relevant information in a long context?

Associative Recall (AR)

- Crucial for strong language modeling capabilities [1,2]

- Definition:

  "The ability to learn and remember the relationship between unrelated items".

- Example - names and professions in a story: *Alice - Lawyer, Bob - Doctor...*

- AR can be quantified via a synthetic task:

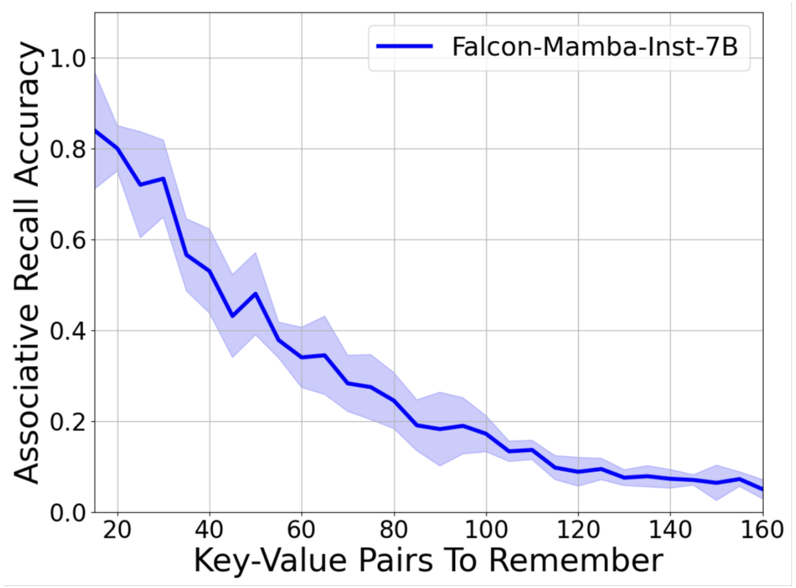
$$K_1 \; V_1 \; <pad> \; <pad> \; K_2 \; V_2 \; <pad> \; <pad> \; \cdots \; K_M \; V_M \; Q \; \textbf{?}$$

[1] In-context learning and induction heads; Olsson et. al.; Anthropic 2022
[2] Zoology: Measuring and Improving Recall in Efficient Language Models; Arora et. al.; ICLR 2024

How good are SoTA recurrent LLMs in AR?

- The recurrent memory capacity is **<u>significantly smaller</u>** than expected.
- Performance **<u>degrades fast</u>** as the amount of information increases beyond the model's limit.

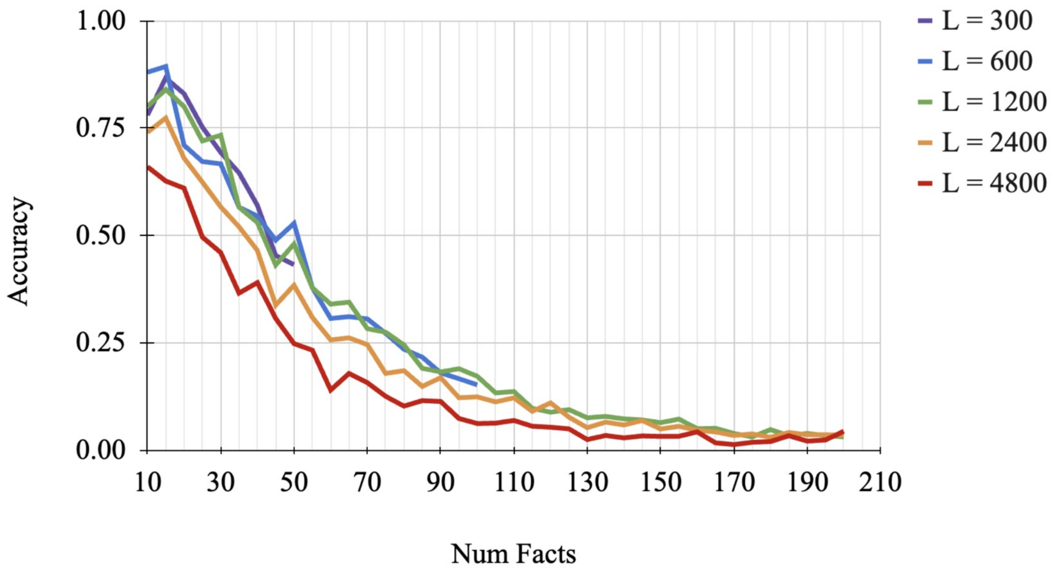
Recurrent Memory Overflows (RMOs)



➤ Sequence length is always 1200 tokens << Training length

RMOs are a fundamental long-context limitation

Recurrent memory overflows show little correlation with sequence length. They are primarily dictated by the **information content** of the context.



Existing Solutions

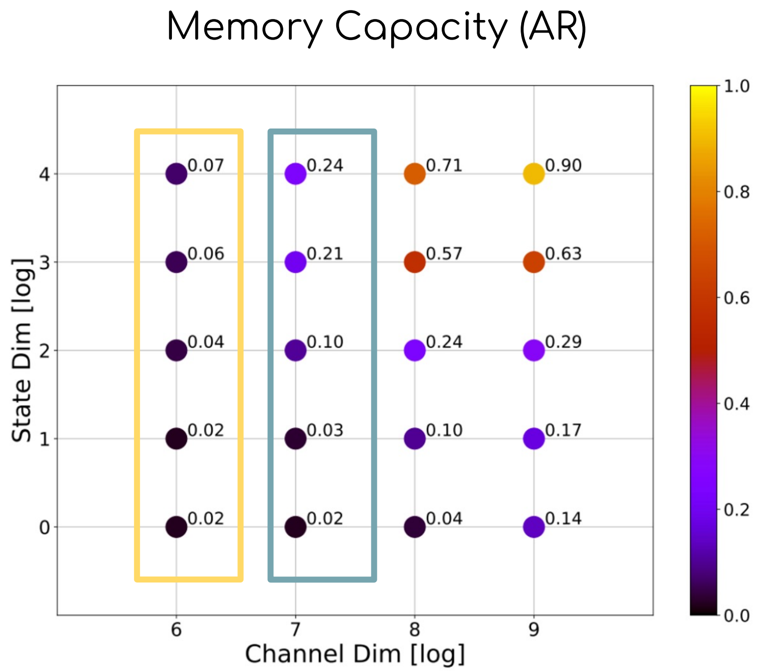
Existing Solutions - Recurrent Memory Overflows

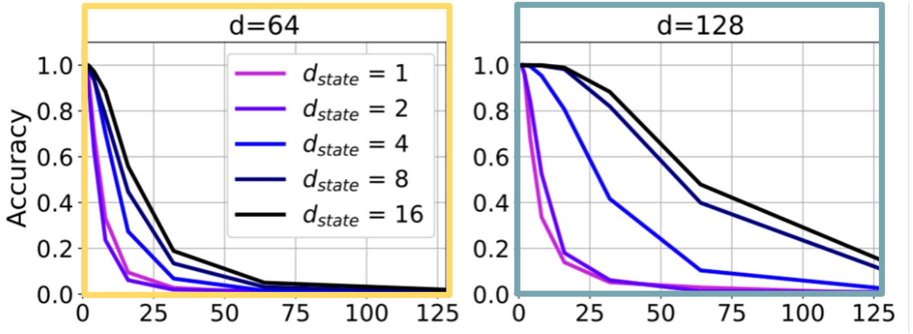
Previous works propose to:

- Increase the state's size (e.g. Mamba2, xLSTM, HGRN2, etc.)
- Manage the memory more effectively (e.g. DeltaNet)

Is this enough?

Toy Example - Expanding the State's Size





Conclusion: Static recurrent memory allocation is not robust!

- ➤ 2 layers, all models are trained to retrieve 128 key-value pairs
- ➤ Mamba state size = ssm_state_dim x channel_dim

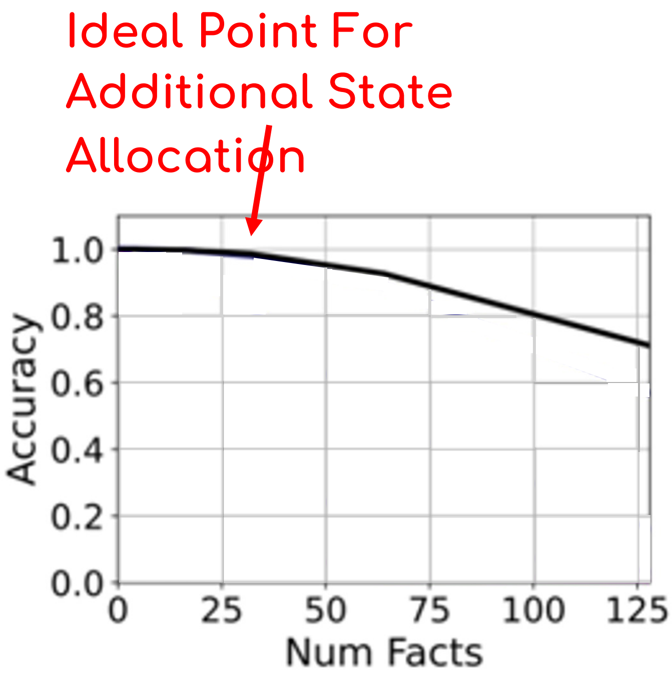
OPRM

Overflow Prevention for Recurrent Models

Overflow Prevention for Recurrent Models (OPRM)

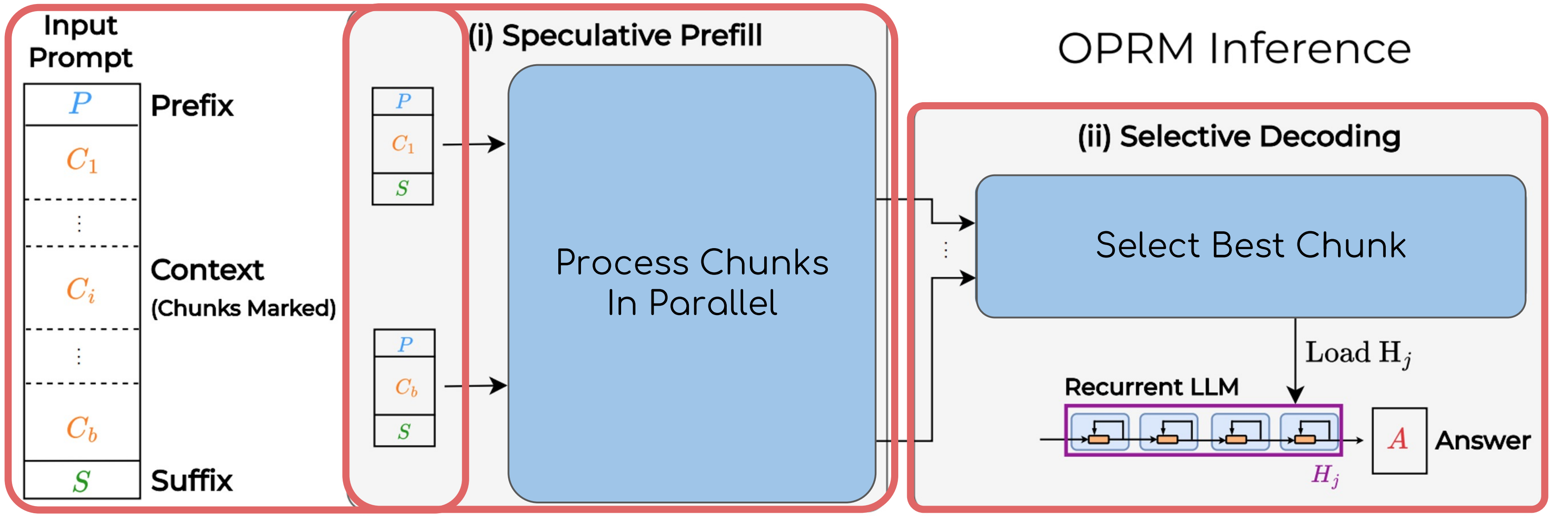
- OPRM:
  A training-free inference algorithm.

- Motivation - “Malloc”
  Dynamically allocate recurrent memory:
  - Allocate more states as information grows.
  - Each state should not process more information than it can reliably store.



Overflow Prevention for Recurrent Models

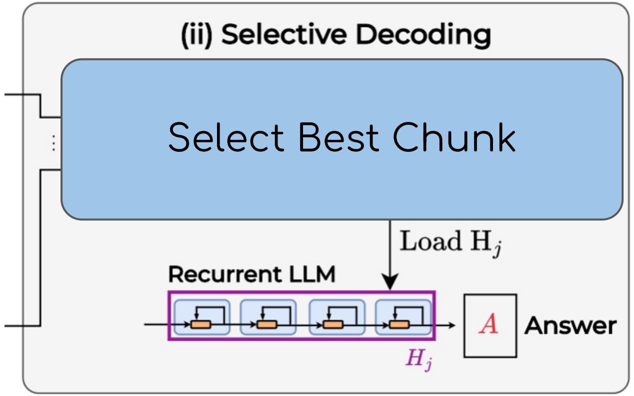
We augment the two inference stages - Pre-fill and Decoding:



Selective Decoding

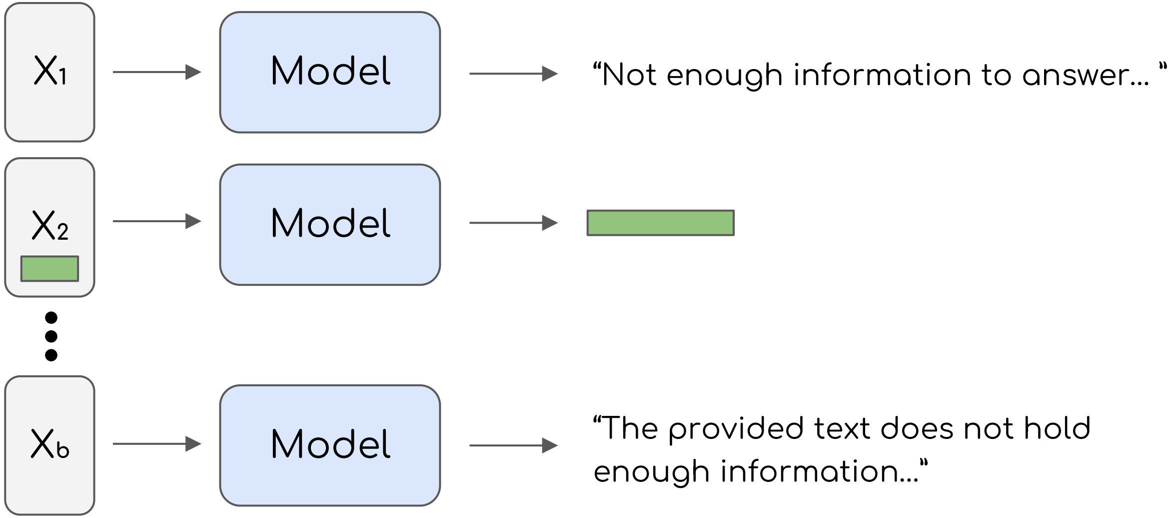
- The chunk is selected according to an entropy criterion.
  Given that we have b chunks X_i, $i \in [b]$, the selected chunk j is given by:

$$j = \arg\min_i \{E_i \mid i \in [b]\}, \quad E_i = \sum_{v \in V} Pr(v \mid X_i) \cdot \log_2 Pr(v \mid X_i)$$

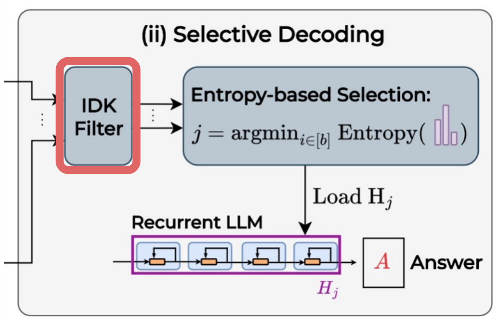


Selective Decoding - IDK Filter

Problem: The entropy criterion selects the most confident chunk. However, since most chunks do not contain the answer, a good model will confidently predict that the answer is not there.



Selective Decoding - IDK Filter

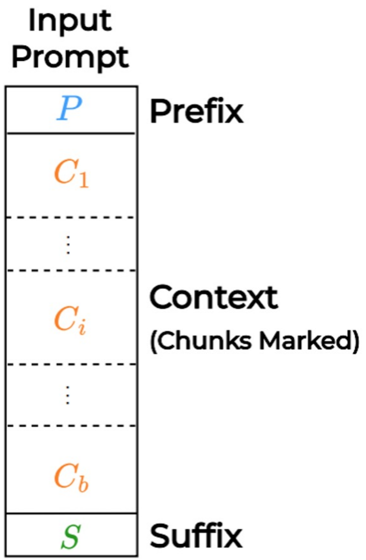


Solution - simple prompting technique.

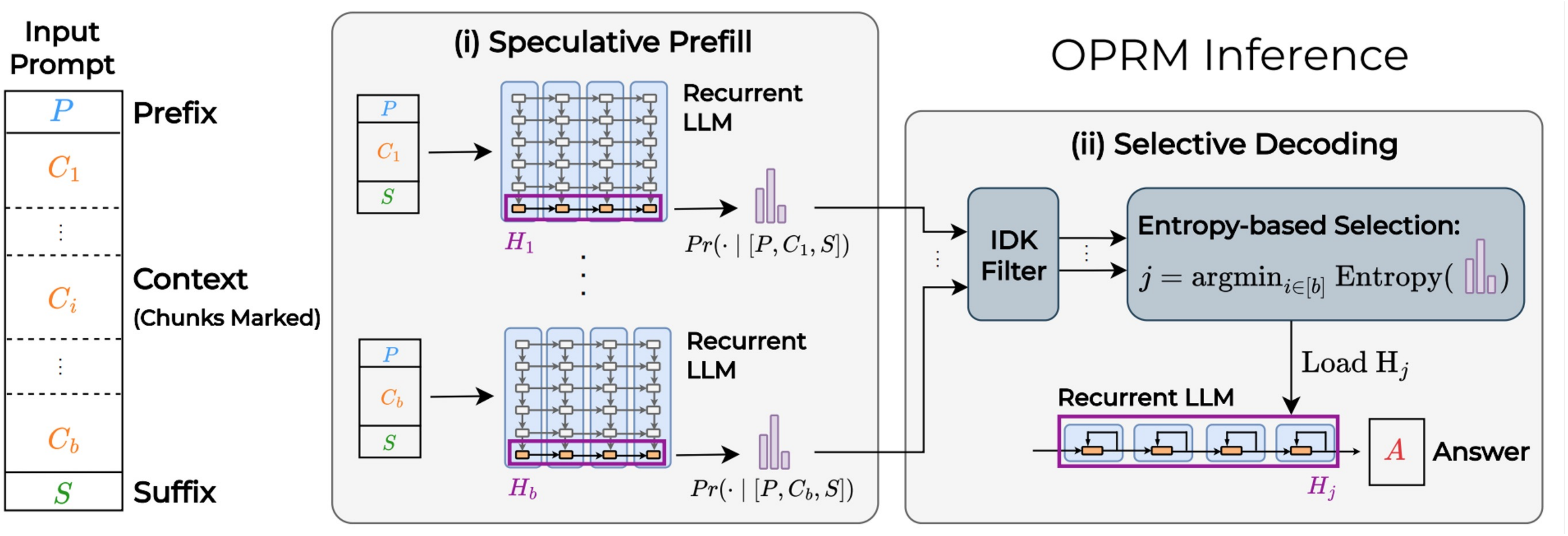
Question: Which town near the county border with North Yorkshire was this Lancashire mill (closed in 1979 and demolished) located? **If the answer is unknown, respond "Error".**

Chunk Size

- Constant (E.g. L=3000 tokens)
- Hyperparameter
- Robust, works well in practice, generalizes to all architectures



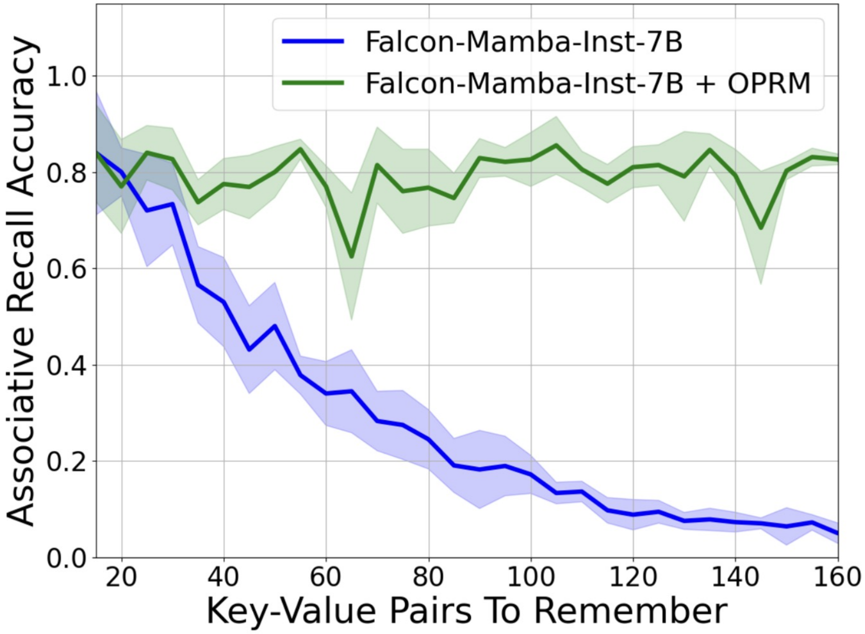
Overflow Prevention for Recurrent Models



Results

Synthetic Tasks - Associative Recall

OPRM practically solves AR:



Real-World Long-Context Tasks - LongBench

Falcon-Mamba-Inst-7B over long-context benchmarks - HotPotQA:

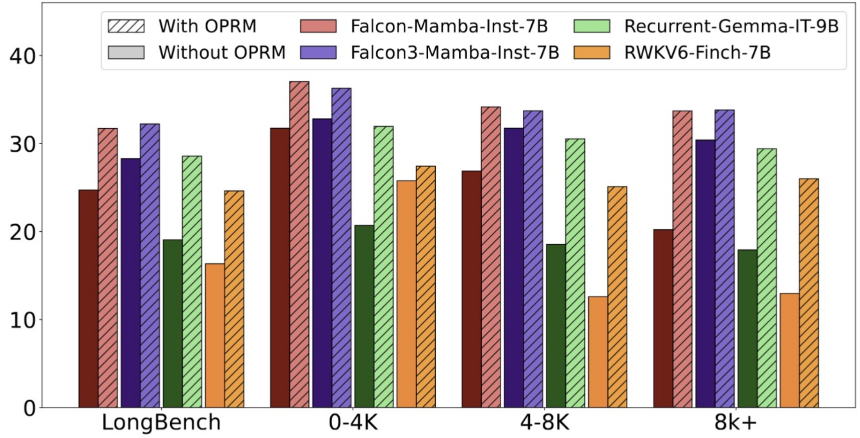
1. Baseline - Process the full context
2. Random - Process only a random chunk from the context
3. OPRM - Select best chunk

Method	0-4K	4K-8K	8K+
Baseline	36.35	21.18	18.4
Random	35.53	23.02	27.62
Min Entropy (Ours)	**39.71**	**37.1**	**35.18**

➤ Context Length = ~10,000 Tokens; Chunk Size = 3000 Tokens

Real-World Long-Context Tasks - LongBench

- Single-Document QA, Multi-Document QA, Summarization, Few-Shot Learning, Synthetic Tasks, and Code Completion.
- Recurrent LLMs + OPRM:
  Average improvement of **35%** across a variety of SoTA recurrent models
- Significant improvements on **Multi-Hop Reasoning** benchmarks



Multi-Hop Reasoning

Benchmark	Method	0-4K	4K-8K	8K+	LB
HotPotQa (2 hops)	Baseline	27.97	21.57	17.21	22.17
	+ OPRM	**38.68**	**34.37**	**36.07**	**35.09**
	Improvement	*38.3%*	*59.3%*	*109.7%*	*58.3%*
MuSiQue ($\leq$ 4 hops)	Baseline	N/A	N/A	N/A	8.37
	+ OPRM	N/A	N/A	N/A	**18.4**
	Improvement	N/A	N/A	N/A	*119.8%*
2WikiMQA ($\leq$ 5 hops)	Baseline	25.26	25.33	16.61	21.39
	+ OPRM	**30.37**	**28.88**	**27.01**	**25.08**
	Improvement	*20.2%*	*14.0%*	*62.7%*	*17.2%*

Real-World Tasks - LongBench v2 (< 1M Tokens)

Model Type	Model	#Params	LB_v2	Difficulty		Length		
				Easy	Hard	0-32k	32k-128k	128k+
–	Random Chance	–	25.0	25.0	25.0	25.0	25.0	25.0
	Human	–	53.7	100.0	25.1	47.2	59.1	53.7
Transformer (Large)	Llama-3.1-Inst-70B	70B	31.6	32.3	31.2	41.1	27.4	24.1
	Mistral-Large-Inst-2411	123B	34.4	38.0	32.2	41.7	30.7	29.6
	Qwen-2.5-Inst-72B	72B	39.4	43.8	36.7	44.4	34.0	41.7
	Qwen-2.5-Inst-72B + YaRN	72B	42.1	42.7	41.8	45.6	38.1	44.4
Transformer (Medium)	Llama-3.1-Inst-8B	8B	30.0	30.7	29.6	35.0	27.9	25.9
	GLM-4-Chat-9B	9B	30.2	30.7	**29.9**	33.9	29.8	25.0
	Qwen-2.5-Inst-7B	7B	27.0	29.2	25.7	36.1	23.7	18.5
	Qwen-2.5-Inst-7B + YaRN	7B	30.0	30.7	29.6	**40.6**	24.2	24.1
Hybrid	RecurrentGemma-IT-9B + OPRM	9B	26.2	26.0	26.4	26.1	22.8	**33.3**
Recurrent	RWKV6-Finch-7B + OPRM	7B	22.7	16.5	16.2	18.3	27.0	21.3
	Falcon-Mamba-Inst-7B + OPRM	7B	29.4	30.2	28.9	27.8	31.2	28.7
	Falcon3-Mamba-Inst-7B + OPRM	7B	**30.8**	**34.4**	28.6	29.4	**32.6**	29.6

Real-World Tasks - Ultra-Long Contexts

- InfiniteBench – Contexts of up to 200K tokens.
- Recurrent LLMs + OPRM:
  - Outperform same-size Transformers
  - Rival Transformers that are an order of magnitude larger

Model	Kimi-Chat	Claude2	GPT4	**Mistral-YaRN**	**Falcon3-Mamba**	**Falcon3-Mamba + OPRM**
Model Type # Params	Transformer >70B	Transformer >70B	Transformer >70B	Transformer 7B	Recurrent 7B	Recurrent 7B
Ret.PassKey	98.1	97.8	100.0	92.7	0.0	**99.8**
Ret.Number	95.4	98.1	100.0	56.6	0.0	**100.0**
Ret.KV	53.6	65.4	89.0	0.0	0.0	**31.3**
En.Sum	17.9	14.5	14.7	9.1	20.1	**22.08**
En.QA	16.5	12.0	22.2	9.6	11.0	**23.2**
En.MC	72.5	62.9	67.3	28.0	45.4	**59.4**
En.Dia	11.5	46.5	8.5	7.5	4.0	**8.0**
Code.Dbg	18.0	2.3	39.6	0.8	**27.1**	24.56
Code.Run	2.0	2.5	23.3	**1.3**	0.0	0.0
Math.Calc	0.0	0.0	0.0	**0.0**	**0.0**	**0.0**
Math.Find	12.6	32.3	60.0	17.1	26.86	**33.14**
Avg	36.2	39.5	47.7	20.2	12.2	36.5

Efficiency

OPRM is **faster** than vanilla inference, with **negligible** memory overhead.

	Model	Context Length						
		2K	4k	8K	16K	32K	64K	128K
Time [s]	Falcon3-Mamba-Inst-7B + OPRM	2.2	2.5	3.2	4.7	7.8	14.0	26.9
	Falcon3-Mamba-Inst-7B	2.0	2.5	3.5	5.7	10.2	18.9	36.2
Space [GB]	Falcon3-Mamba-Inst-7B + OPRM	15.3	15.6	16.3	17.8	20.6	26.2	37.3
	Falcon3-Mamba-Inst-7B	14.9	15.3	15.9	17.2	19.9	25.2	35.7

Increase of 4.4% at the most extreme case

Explanation:
Speculative pre-fill allows **parallelization** in the sequence axis.
GPU memory used by a recurrent state is just **1/1000** of the model size.

Context Extension

Explanation:

overflow prevention → context extension

context extension ~~→~~ overflow prevention

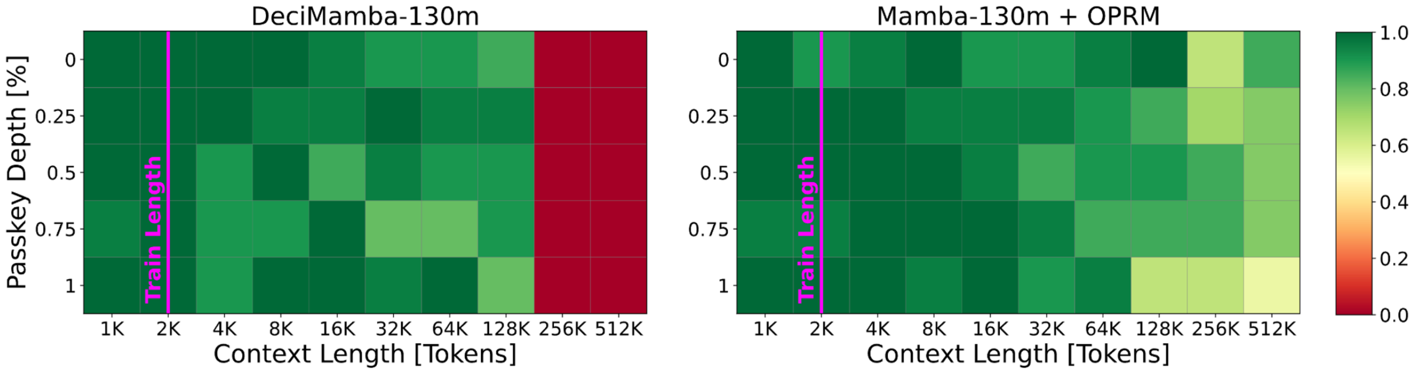
- Overflow prevention methods naturally extend the context lengths that models can handle.
- OPRM is better than dedicated methods at context extension!

LongBench

Method	SD-QA	MD-QA	Summ	Few-Shot	Syn	Code	Avg
Mamba-1.4b	5.94	5.92	7.80	12.56	3.00	12.92	9.35
DeciMamba-1.4b	6.42	6.19	9.78	17.54	**3.12**	35.17	15.25
LongMamba-1.4b	6.76	7.57	10.34	28.21	2.88	42.78	17.33
Mamba-1.4b + OPRM	**11.36**	**8.92**	**20.22**	**35.58**	2.60	**43.25**	**21.22**

Training Length = 2K tokens

Needle-In-A-Haystack



Comparison to RAG

- Long-context vs. RAG remains a persistent debate.
- Recurrent LLMs + OPRM strengthen the case for long-context modeling:

Document QA Benchmarks - LongBench_e

Method	0-4K	4K-8K	8K+	Avg	
Falcon-Mamba-Inst-7B	34.21	26.94	19.11	26.75	
+ Dragon	35.85	30.05	32.74	32.88	RAG (External Retriever)
+ PRP	36.27	32.52	34.61	34.47	RAG (LLM Retriever - Slow)
+ OPRM	**37.41**	**34.49**	**36.25**	**36.05**	

Possible explanation: “Multi-Hop Recall” [1]

[1] Retrieval meets long context large language models; Zu et. al.; ICLR 2024

Limitations

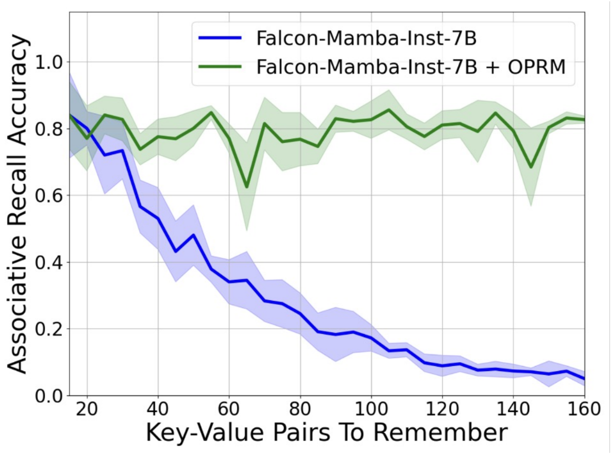
Limitations

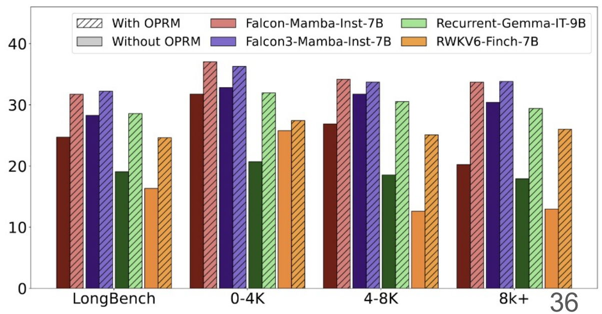
- OPRM does not perform cross-chunk information aggregation.
- OPRM is training-free - it relies on the base-model's abilities.
  E.g. IDK filter can be improved if explicitly trained on.

Recap

Recap

- Recurrent memory overflows fundamentally limit long-context performance of recurrent LLMs.
- Existing solutions compress the whole context into a fixed-size state - don't fully prevent RMOs.
- OPRM prevents RMOs by dynamically allocating more recurrent memory.
- OPRM is general, effective and efficient:
  - Boosts long-context performance
  - Runs faster than vanilla inference
  - Enables significant length generalization
  - Training-free, architecture-agnostic





Thank You!

OPRM on Github