How to Train Long-Context Language Models

Pristina Wang, Mahler Revsine

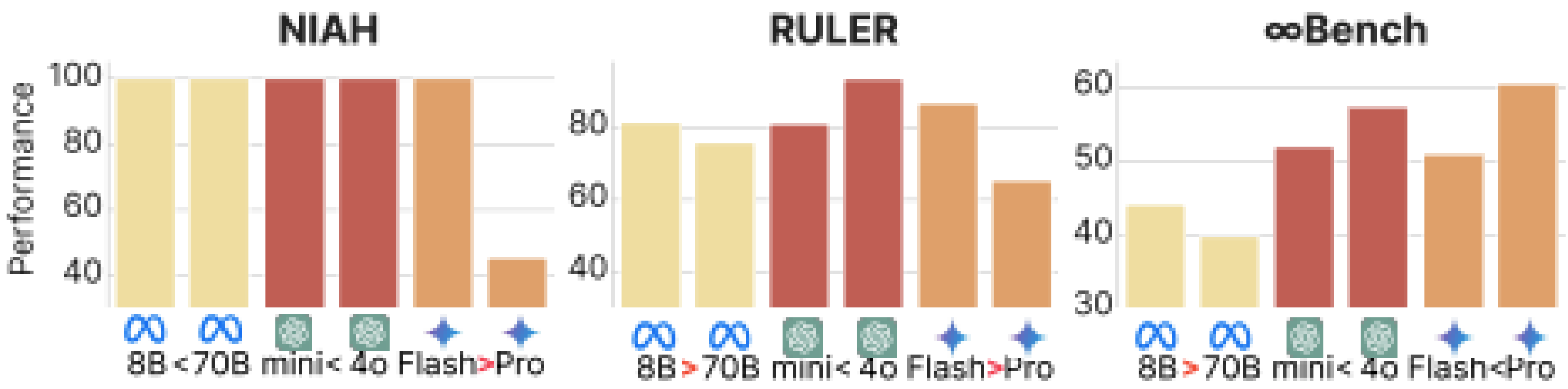
JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Evaluation

- ~~Needle-in-a-hay-stack and RULER~~
- HELMET benchmark

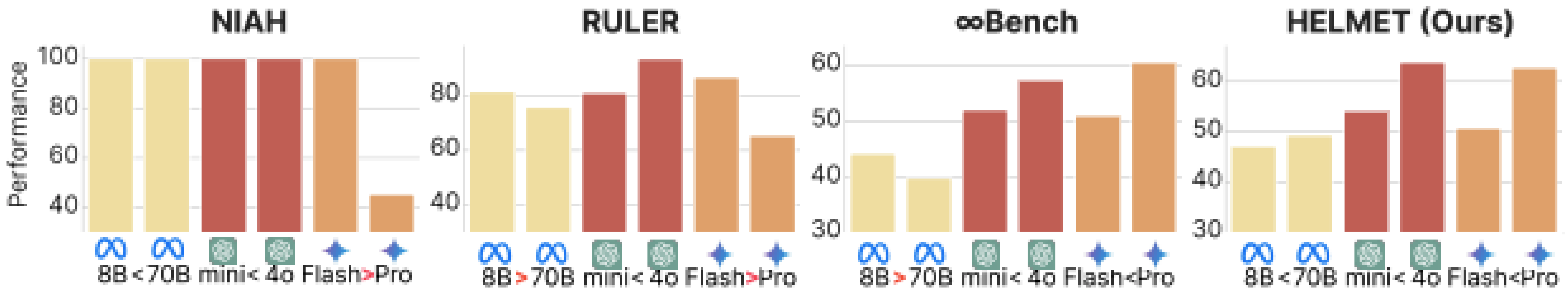
Why not current benchmark

- Results are saturated



Why not current benchmark

- HELMET has consistent ranking



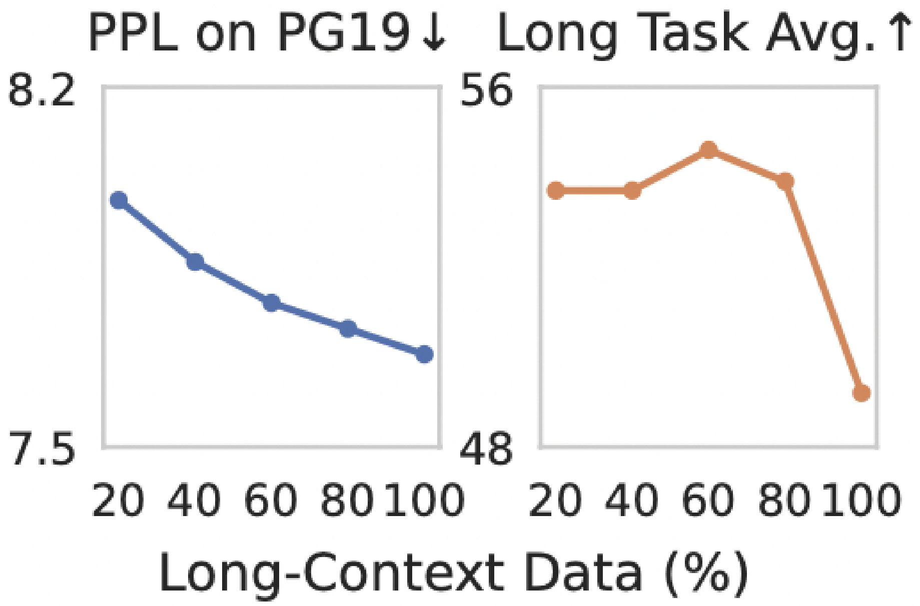
HELMET

- 7 diverse application-centric categories
- Recall
- RAG
- Re-ranking
- ICL
- QA
- Summarization

	Type of tasks							Benchmark features		
	Cite	RAG	Re-rank	Long-QA	Summ	ICL	Synthetic Recall	Robust Eval.	$L \geq 128k$	Controllable L
ZeroSCROLLS	✗	✗	✗	✓	✓	✗	✗	✗	✗†	✗
LongBench	✗	✓	✗	✓	✓	✓	✓	✗	✗†	✗
L-Eval	✗	✓	✗	✓	✓	✗	✗	✓‡	✗†	✗
RULER	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
∞BENCH	✗	✗	✗	✓	✓	✗	✓	✗	✓	✓
HELMET (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

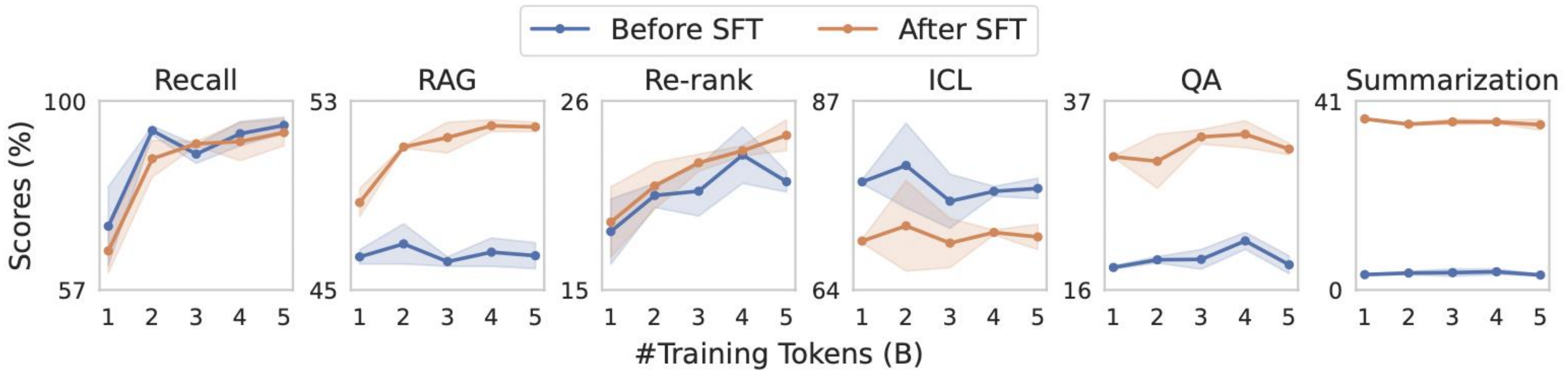
Why not perplexity?

- Empirical results
- While using more long data continues to improve PPL, using 100% long data hurts downstream long-context performance



Effects of Supervised Finetuning

- SFT shows that the models continue to improve with more training tokens on RAG, re-ranking while unclear when evaluated before
- SFT enables evaluation on QA and summarization



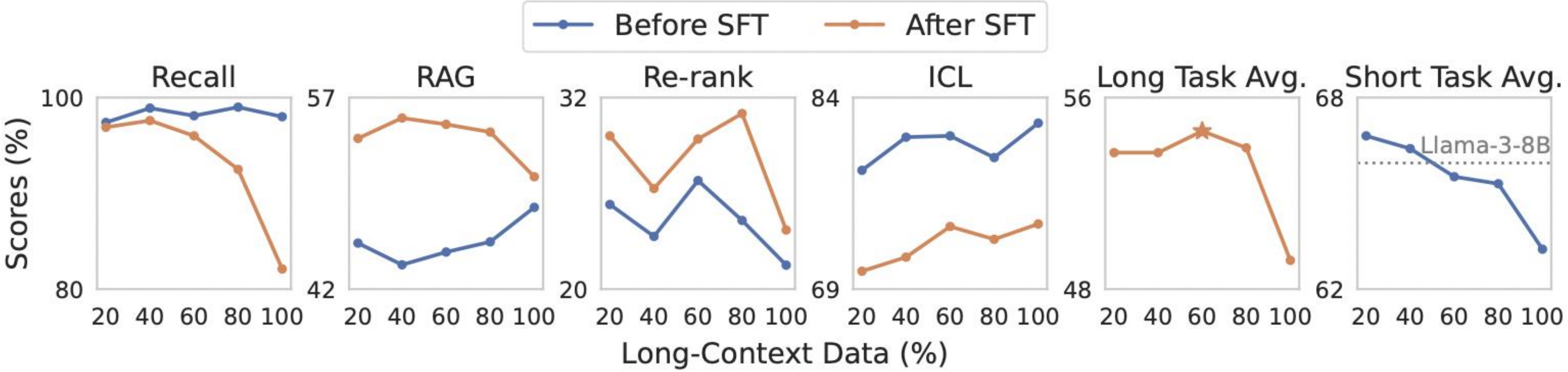
Short context performance

- Previous techniques lower short-context performance
- Position extrapolation(PE) and fine-tuning on long-context mixture SlimPajama

	HSwag	MMLU	ARC-c	WG	GSM8K
Llama-3-8B	82.1	66.5	59.4	77.1	44.7
+ PE	81.5	64.7	58.1	75.5	40.1
+ SlimPajama	81.0	63.1	57.8	75.1	40.6

Long-context data curation

- Mixture of long and short
- 60% long and 40% short



Long context source

- Book/code 1:1 is the best

Long Data (60%)	Long-Context							Short-Context
	Recall	RAG	Re-rank	ICL	QA	Summ.	Avg.	Avg.
CommonCrawl	84.1	53.3	28.1	67.5	35.2	37.0	50.9	66.5
Books	94.9	53.9	**30.7**	72.2	33.2	37.7	53.8	65.5
Code Repos	**99.2**	53.8	29.0	61.2	34.7	36.2	52.3	65.9
Books/Repos 1:1	96.0	**54.9**	29.4	**73.9**	**35.7**	**37.9**	**54.6**	65.5

Short context source

- More knowledge-intensive downstream-related data is the best
- Retained llama-3-8b math ability the most

Short Data (40%)	Long-Context	Short-Context					
	Avg.	HellaS.	MMLU	ARC-c	WG	GSM8K	Avg.
Original model (Llama-3-8B)	-	82.1	66.5	59.4	77.1	44.7	66.0
SlimPajama	52.9	81.2	63.0	58.5	76.2	41.9	64.2
FineWeb-Edu	53.0	81.0	62.6	57.7	74.4	39.4	63.0
DCLM-Baseline	52.0	82.0	65.6	59.6	77.4	39.4	64.8
ProLong ShortMix	**54.6**	81.6	65.3	58.0	76.2	46.6	**65.5**

Components	%
FineWeb	25
FineWeb-Edu	25
Wikipedia	10
Tulu-v2	10
StackExchange	10
ArXiv	10
OpenWebMath	10

Train for more steps

- Train **20B** tokens of length **64K**, then **20B** tokens of length **512K**
- Improves performance on all long-context tasks

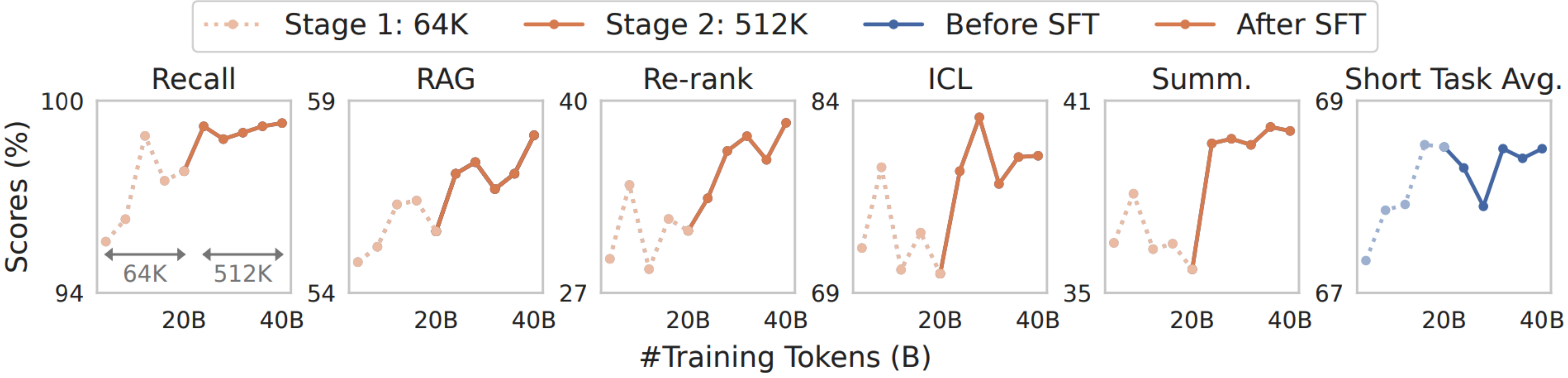


Figure 4: Performance (avg. of 32K and 64K) of our ProLong model throughout training.

Train with a longer context than you evaluate on

Table 7: Impact of training models on different sequence lengths. All the results are evaluated at a sequence length of 64K. We see that training at a maximum length beyond the evaluation context window consistently improves the long-context performance.

Max Seq. Length	Recall	RAG	Re-rank	ICL
ProLong 64K training (20B)	96.5	52.7	22.8	70.6
+4B 64K training	95.0	56.4	28.0	78.8
+4B 512K training	**98.5**	**56.9**	**32.9**	**79.2**

- Training on **longer** texts improves performance on **shorter** ones
- Better results on all **64K** inference tasks when trained on **512K**

Should we fine-tune with synthetic data?

- **Previous work** suggests synthetic data **improves** SFT
- Instruction datasets have **short** prompts (**1-4K** tokens)
- Could **supplement** them with **synthetic long** instruction data
- 40% QA (Question - Answer)
  - Llama-3-8B-Instruct generates QA pairs from chunks of long documents
- 30% RAG (Retrieval-Augmented Generation)
  - Reuse QA pairs, add random chunks from the document as faux retrievals
- 30% Synthetic summarization
  - Generate summaries of books via recursive summarization

Synthetic data harms model performance

Table 8: Effect of different ratios of synthetic SFT data (mixed with UltraChat). We report the 32K-and-64K-averaged performance except tasks marked with †, which are evaluated at 512K for stress testing. The number of percentage is based on #tokens, not #samples.

% Synthetic Data	JsonKV†	RAG	Re-rank	ICL	QA†	Summ.†	Avg.
0%	65.7	58.1	38.5	80.3	49.7	42.1	**55.7**
1%	61.5	57.0	38.3	80.8	45.3	41.5	54.1
3%	62.0	56.4	37.9	80.6	44.8	39.5	53.5
10%	70.3	55.5	36.1	80.6	41.7	39.4	53.9
50%	45.8	48.8	18.8	70.5	42.3	33.3	43.3

- Best performance with **0% synthetic** data
- Opposite trend found in previous work

Discrepancy with prior work

- This study found that synthetic fine-tuning data was **detrimental** to task performance
- Previous studies showed **benefits** to fine-tuning with **synthetic** data

- The authors suggest two reasons for this difference:
  1. Prior models had **insufficient** long-context training
     - Synthetic fine-tuning samples acted as additional training data
  2. Prior approaches used **much larger** fine-tuning datasets
     - Synthetic samples helped prevent model degeneration

The final recipe for ProLong

- Start with **Llama-3-8B-Instruct** model
- 20B **64K**-length tokens, then 20B **64 / 512K** tokens
  - **Increase RoPE base** from 10^6 to 10^8 for longer tokens
  - Include 3% **textbooks**
- SFT via **UltraChat** dataset
- Disable **cross-document** attention

Table 9: The training recipe for ProLong.

	Continued Long-context Training	
Data	30% code repos, 30% books, 3% textbooks, 37% ShortMix	
	ShortMix:	27% FineWeb-Edu, 27% FineWeb, 11% Tulu-v2, 11% StackExchange, 8% Wikipedia, 8% OpenWebMath, 8% ArXiv
Length Curriculum	Stage 1 (64K):	Code repos, books, and textbooks at length 64K
	Stage 2 (512K):	Code repos: 50% at length 512K, 50% at length 64K Books: 17% at length 512K, 83% at length 64K Textbooks at length 512K
Steps	Stage 1: 20B tokens (2.2K H100 hours), Stage 2: 20B tokens (12.2K H100 hours)	
Model	Initialization: RoPE: Attention:	Llama-3-8B-Instruct (original RoPE base freq. 5×10^5) Stage 1: 8×10^6, Stage 2: 1.28×10^8 Full attention with cross-document attention masking
Optim.	AdamW (weight decay = 0.1, $\beta_1 = 0.9$, $\beta_2 = 0.95$) LR: Batch size:	 $1e-5$ with 10% warmup and cosine decay to $1e-6$, each stage 4M tokens for stage 1, 8M tokens for stage 2
	Supervised Fine-tuning (SFT)	
Data	UltraChat	
Steps	1B tokens	
Optim.	AdamW (weight decay = 0.1, $\beta_1 = 0.9$, $\beta_2 = 0.95$) LR = $2e-5$ (cosine decay to $2e-6$), warmup = 5% Batch size = 4M tokens	

ProLong is the best model of its size

Table 10: Our main evaluation results on HELMET (Yen et al., 2024b; details in §A.1). All results are averaged over sequence lengths of 32K, 64K, and 128K. For all models, we use the corresponding instruction version. ProLong is one of the best performing 10B-scale LMs. The complete set of results can be found in §C.

Model	Max Len.	Recall	RAG	ICL	Re-rank	QA	Summ.	Avg.
ProLong (8B)	512K	**99.4**	**66.0**	81.1	**33.2**	**40.8**	40.5	**60.2**
MegaBeam-Mistral (7B)	512K	**99.4**	58.1	**82.1**	22.1	33.7	43.6	56.5
Meta-Llama-3.1 (8B)	128K	98.7	62.8	79.7	26.6	40.4	**46.1**	59.0
Qwen2 (7B)	128K	34.4	43.4	54.8	4.6	23.3	38.5	33.2
Phi-3-small (7B)	128K	74.8	60.6	82.0	18.5	34.1	42.4	52.1
Mistral-Nemo (12B)	128K	24.9	48.1	82.0	4.7	37.7	37.0	39.1
Jamba-1.5-Mini (12B/52B)	256K	87.7	61.3	88.4	25.9	42.0	38.6	57.3
Meta-Llama-3.1 (70B)	128K	98.5	65.9	80.0	39.4	47.2	51.1	63.7
Claude-3.5-Sonnet	200K	99.4	44.0	79.3	19.9	38.1	49.2	55.0
Gemini-1.5-Pro	2M	94.2	71.4	78.9	65.3	44.4	56.2	68.4
GPT-4o	128K	99.9	71.5	86.7	59.6	47.0	55.7	70.1

- Outperforms Llama-3.1 with **5%** as many training examples

ProLong gets better at longer contexts

Table 11: ProLong at 512K.

	32K	64K	128K	512K
QA	31.7	43.7	46.7	**49.7**
Summ	40.4	39.8	41.5	**42.1**

- ProLong performs best at 512K
- By contrast, many other models cannot run inference on more than 128K tokens

ProLong excels on the NoCha benchmark

Table 12: Results on the NoCha benchmark (Karpinska et al., 2024).[9] ProLong is the only model that achieves above-random performance in the <75K category and it consistently beats Llama-3.1. Different from the original NoCha leaderboard, we report the average accuracy over all test instances without filtering the test examples based on the model's context window lengths.

Model	Max Len.	<75K	75K-127K	127K-180K	>180K
ProLong (8B)	512K	**28.4**	17.0	13.1	**20.3**
MegaBeam-Mistral (7B)	512K	19.8	**18.3**	**17.5**	15.6
Meta-Llama-3.1 (8B)	128K	17.3	16.4	0.0	0.0
Mistral-Nemo (12B)	128K	13.6	0.4	0.0	0.0
Jamba-1.5-Mini (12B/52B)	256K	27.2	28.0	24.4	6.2
Meta-Llama-3.1 (70B)	128K	42.0	25.0	0.0	0.0
Gemini-1.5-Pro	2M	24.7	38.8	35.3	46.9
GPT-4o	128K	55.6	58.4	0.0	0.0

- Claim verification dataset
- Human-curated, global **reasoning** tasks

Main takeaways

- Much of the **existing** literature on long-context training is **wrong**
- Sufficient to change the data **composition**, training **regimen**, and **hyperparameters**
- Improvement in this domain is possible!