Long Context LMs and Retrieval

Can Long-Context Language Models Subsume Retrieval, RAG, SQL, and More?
Long-Context LLMs Meet RAG: Overcoming Challenges for Long Inputs in RAG

Siddharth Gollapudi and Dongwei Lyu

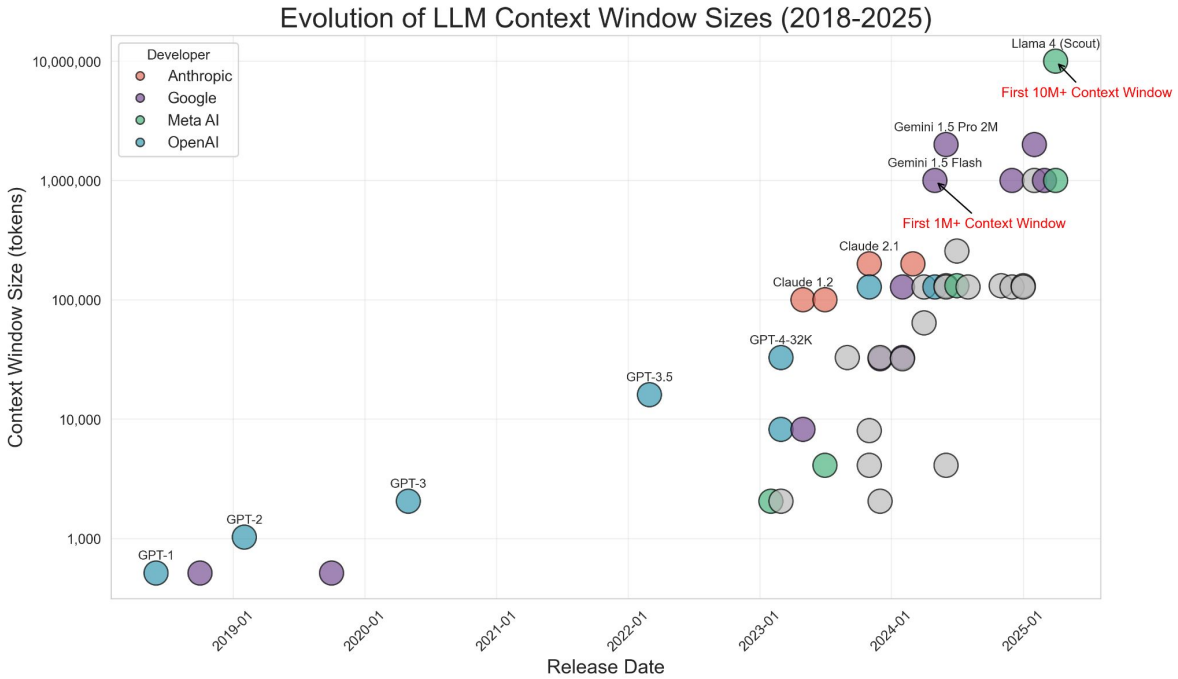
October 21th, 2025

Plan

1. Long context Language Models (LCLMs) primer
2. Paper 1
3. Paper 2
4. Discussion

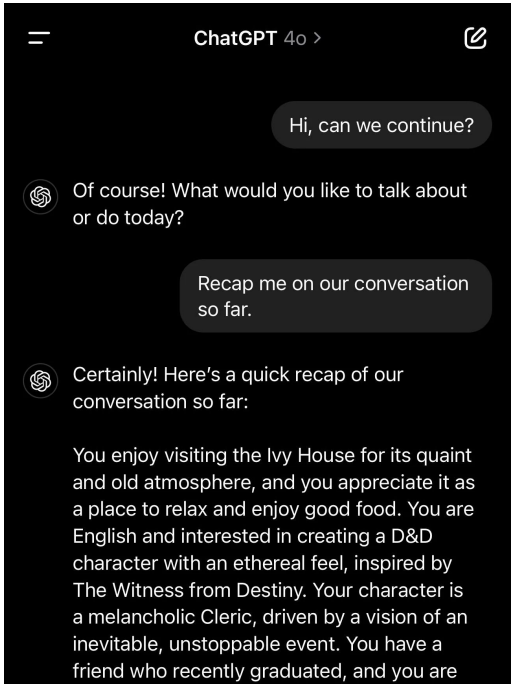
Long Context Language Modeling (LCLMing)

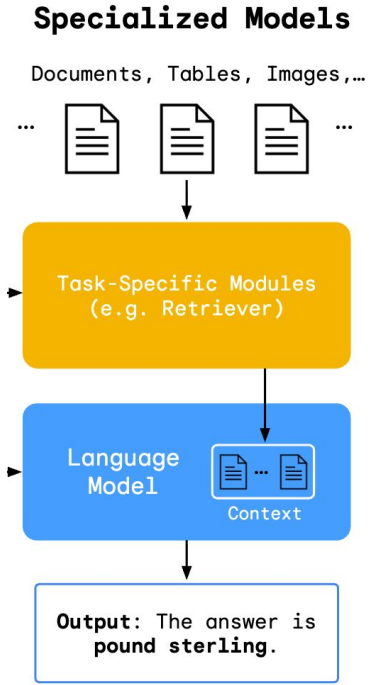
Every LLM has a **rated** context length!



https://www.meibel.ai/post/understanding-the-impact-of-increasing-llm-context-windows

What do we use the longer context for?





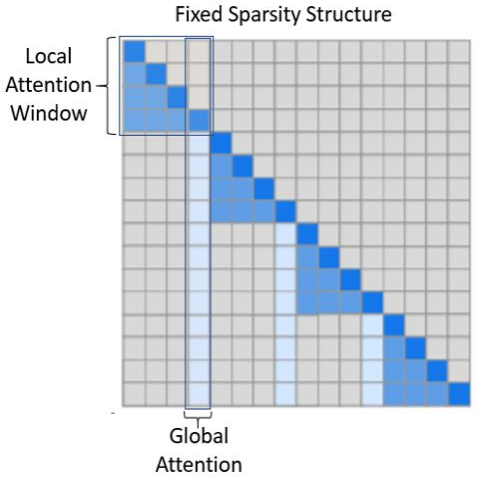
$\mathcal{Q}$: Beth bakes 4, or 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?
$\mathcal{R}$: Beth bakes 4 2 dozen batches of cookies for a total of $4 * 2 =<< 4 * 2 = 8 >>$ 8 dozen cookies. There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of $12 * 8 =<< 12 * 8 = 96 >>$ 96 cookies. She splits the 96 cookies equally amongst 16 people so they each eat $96/16 =<< 96/16 = 6 >>$ 6 cookies.
$\mathcal{A}$: 6

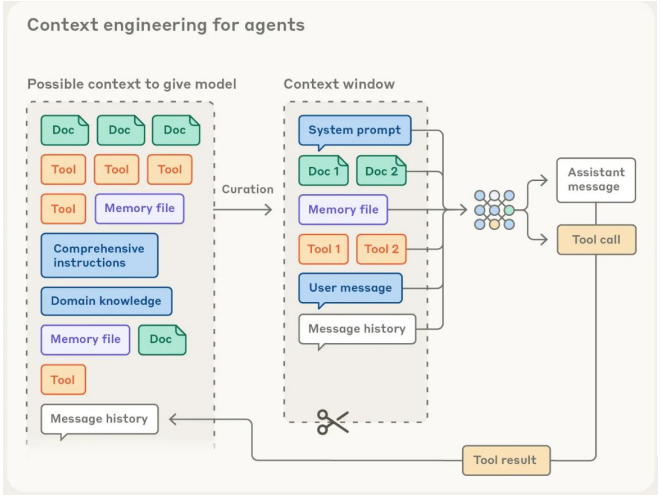
Ahn, et. al. 2024

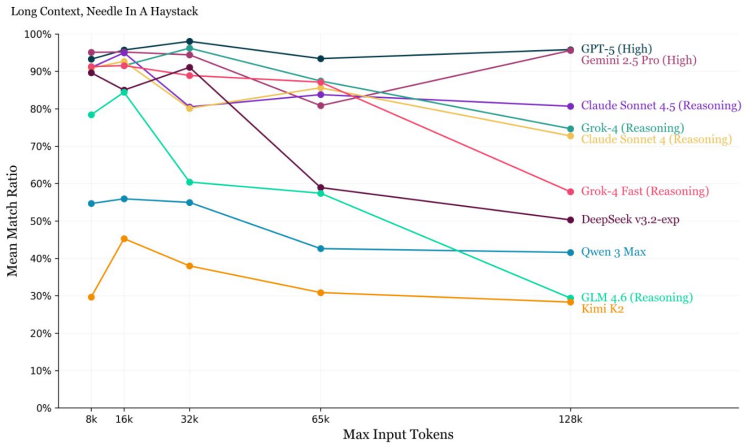
Issues with LCLMing

What do 1M and 500K context windows have in common? They are both actually 64K.

- Attention scales very poorly with context length
- Context Engineering: are models using their contexts properly?
- Poor benchmarking







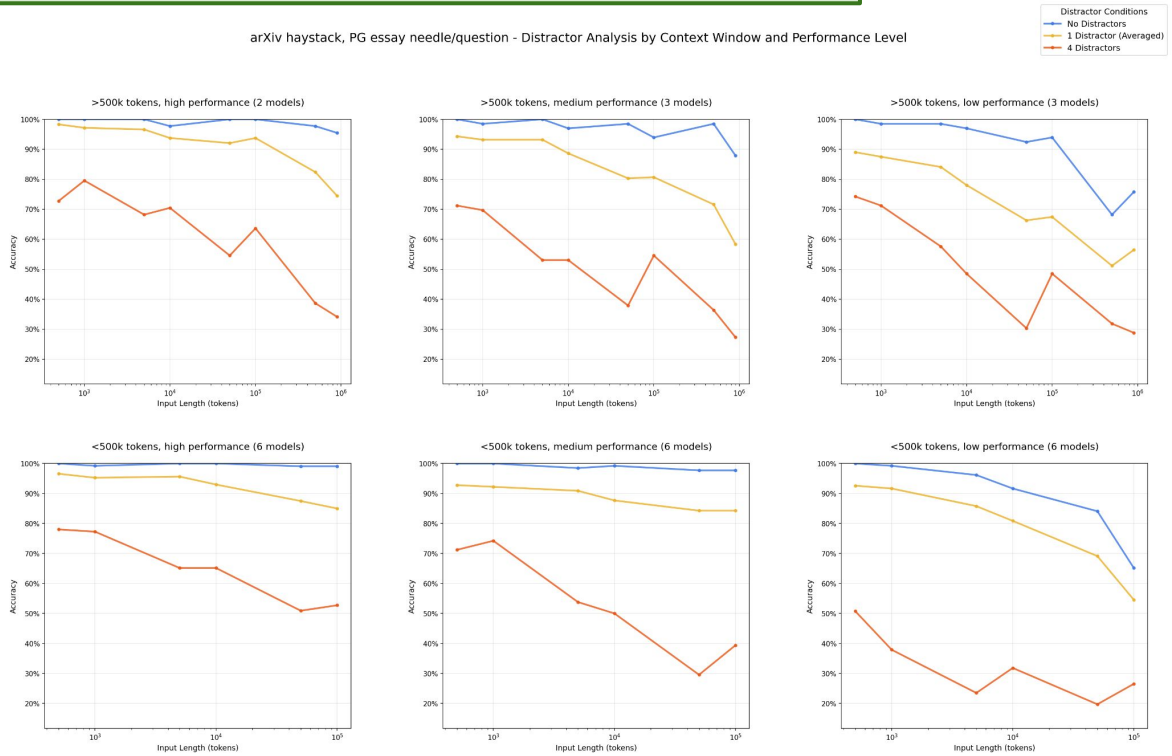
Context Engineering 1: Context Rot

Vibes: The **more** you use an LLM, the **worse** it gets!

Hong, et al. (2025)

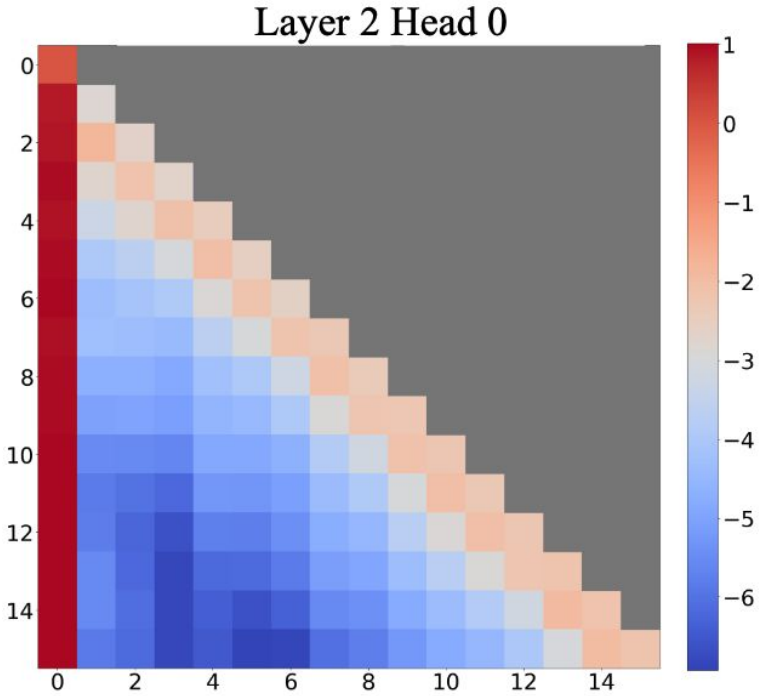
Needle-in-a-haystack (NIAH):

1. Question + highly relevant “needle”
2. Distractors
3. Model must “find” the needle from the “haystack”
4. *Can* the model attend to every token?
5. *Does* the model understand every token?

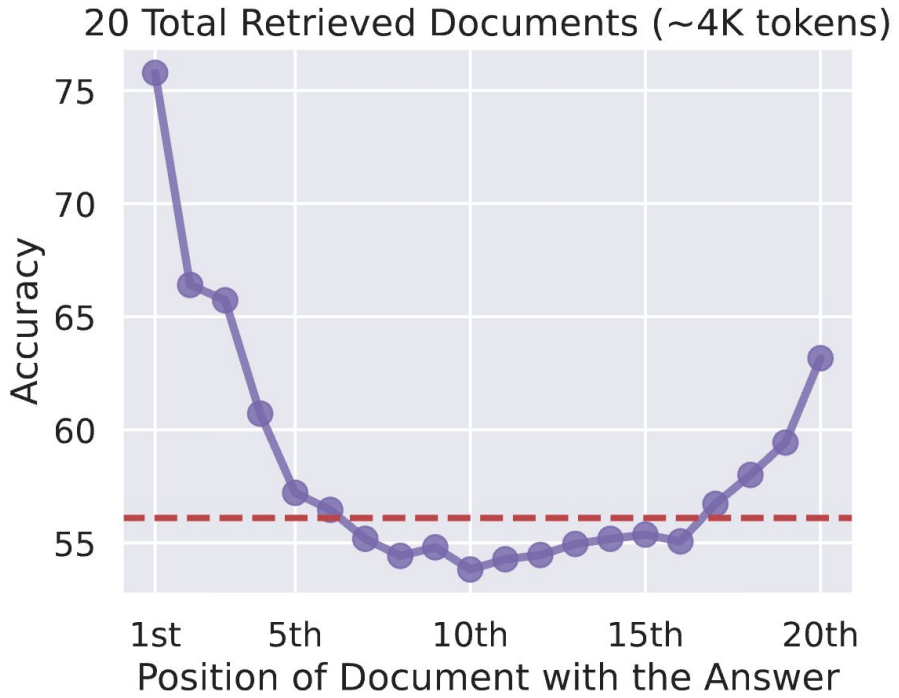


Context Engineering 2: Positional Biases

Claim: Pretraining introduces **biases** towards certain context positions



Attention sinks (Xiao, et al. 2023)



Lost-in-the-middle (Liu, et al. 2023)

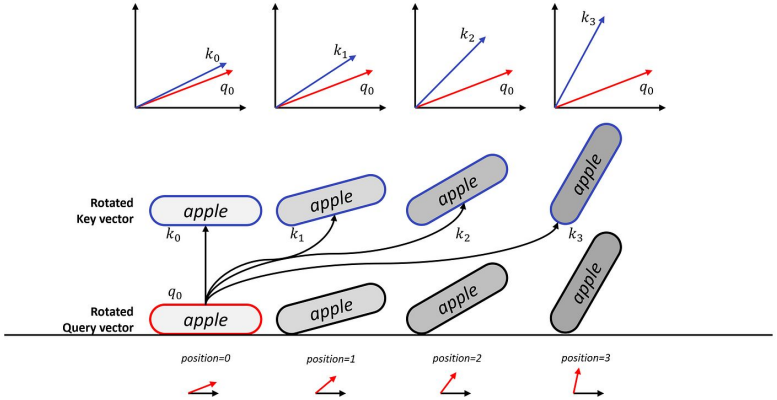
Context Engineering 3: Positional Encodings

$$f_{\{q,k\}}(\boldsymbol{x}_m, m) = \boldsymbol{R}^d_{\Theta,m} \boldsymbol{W}_{\{q,k\}} \boldsymbol{x}_m \tag{14}$$

where

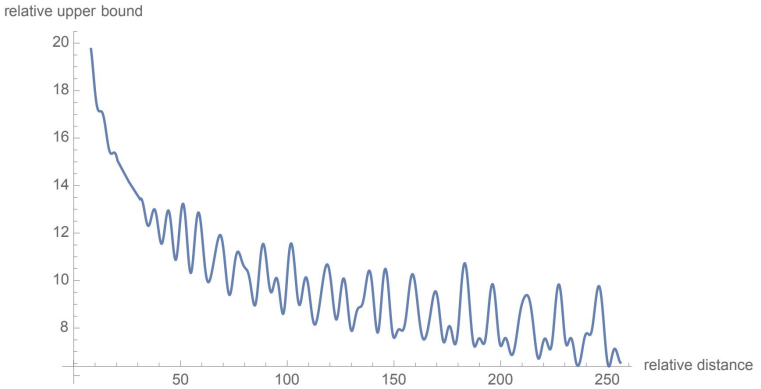
$$\boldsymbol{R}^d_{\Theta,m} = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2} & -\sin m\theta_{d/2} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{pmatrix} \tag{15}$$

Test sentence : *"apple apple apple apple"*



RoPE (Su, et al. 2021)

Long-term decay



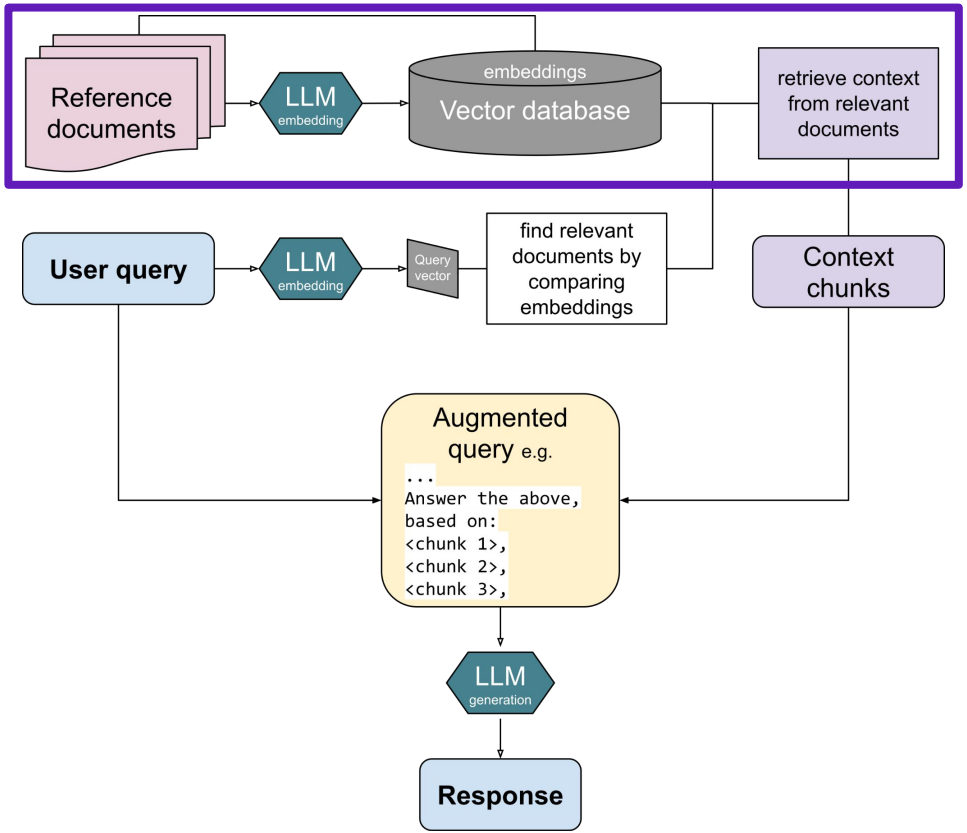
Position Interpolation

$$f'_{\mathbf{W}}(\mathbf{x}_m, m, \theta_d) = f_{\mathbf{W}}\left(\mathbf{x}_m, \frac{mL}{L'}, \theta_d\right)$$

L' >L, enables efficient length generalization

YaRN (Peng, et al. 2023)

Recap: Retrieval Augmented Generation



Wikimedia

Can Long-Context Language Models Subsume Retrieval, RAG, SQL, and More?

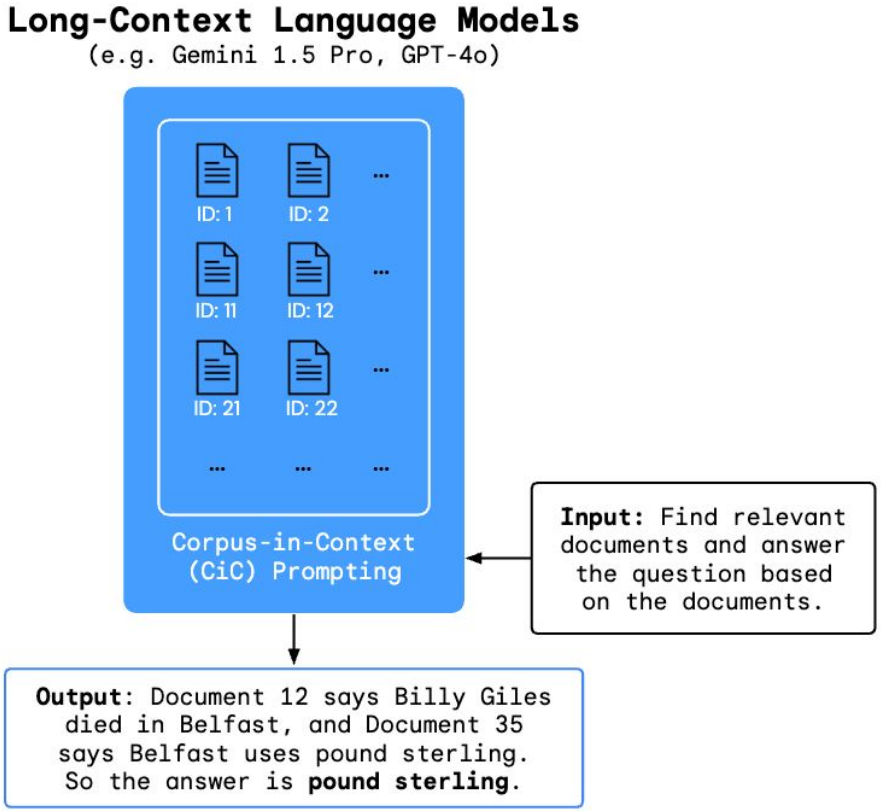
Motivation 1: Retrieval with Long Contexts?

Problem: RAG pipelines suffer from cascading errors

Hypothesis: Do it all with the language model?

Support:

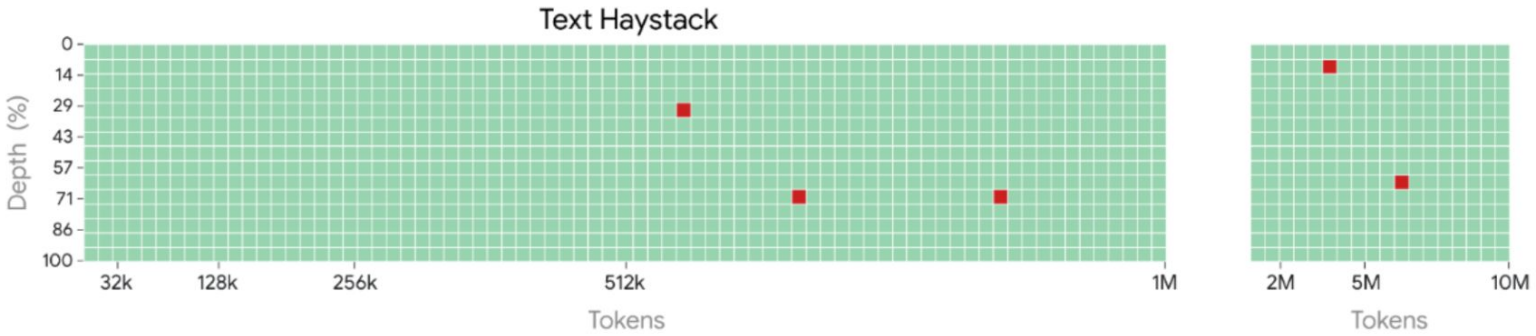
- We've gone from 8k *rated* context lengths to 1M!
- Give the language model as much info as possible



Motivation 2: Benchmark Quality

Existing Benchmarks are way too easy!

Text
Up to 7M words
(10M tokens)



Level 3: Can the model perform well on **real-world** tasks?

Contributions

Benchmark (LOFT):

1. Retrieval (Visual, Audio, text)
2. RAG
3. “SQL”
4. Many-many-shot ICL

Evaluation:

1. Frontier models
2. Existing Baselines
3. Ablations

(Text) Retrieval

Simplify: place corpus directly in LCLM's context, removing need for encoder

Enables: better reasoning, multihop QA, few-shot tasks, etc.

Task	Dataset	Description	Avg. Cand. Length	# Cand. (128k)	Candidates	Input	Target
Text Retrieval	ArguAna	Argument Retrieval	196	531	Passages	Query	Passage ID(s)
	FEVER	Fact Checking	176	588			
	FIQA	Question Answering	196	531			
	MS MARCO	Web Search	77	1,174			
	NQ	Question Answering	110	883			
	Quora	Duplication Detection	14	3,306			
	SciFact	Citation Prediction	301	357			
	Touché-2020	Argument Retrieval	330	329			
	TopiOCQA	Multi-turn QA	149	680			
	HotPotQA	Multi-hop QA	74	1,222			
	MuSiQue	Multi-hop QA	120	824			
	QAMPARI	Multi-target QA	132	755			
	QUEST	Multi-target QA	328	328			

RAG

Simplify: pipeline complexity, e.g. query decomposition

Enables: Better reasoning

Corpus: Shared across all queries, subset to fit 128k context limit

- Gold/random documents are shuffled to minimize positional bias

Task	Dataset	Description	Avg. Cand. Length	# Cand. (128k)	Candidates	Input	Target
RAG	NQ	Question Answering	110	883	Passages	Question	Answer(s)
	TopiOCQA	Multi-turn QA	149	680			
	HotPotQA	Multi-hop QA	74	1,222			
	MuSiQue	Multi-hop QA	120	824			
	QAMPARI	Multi-target QA	132	755			
	QUEST	Multi-target QA	328	328			

SQL

Simplifies: text-to-SQL, structured data processing

Enables: expressive querying, handling of noisy data, unified processing

Corpus: associated database of tables

- The chosen database is the largest possible that fits into the context

Task	Dataset	Description	Avg. Cand. Length	# Cand. (128k)	Candidates	Input	Target
SQL	Spider SParC	Single-turn SQL Multi-turn SQL	111k 111k	1 1	SQL Database	Question	Answer

How do we actually handle the data in-context?

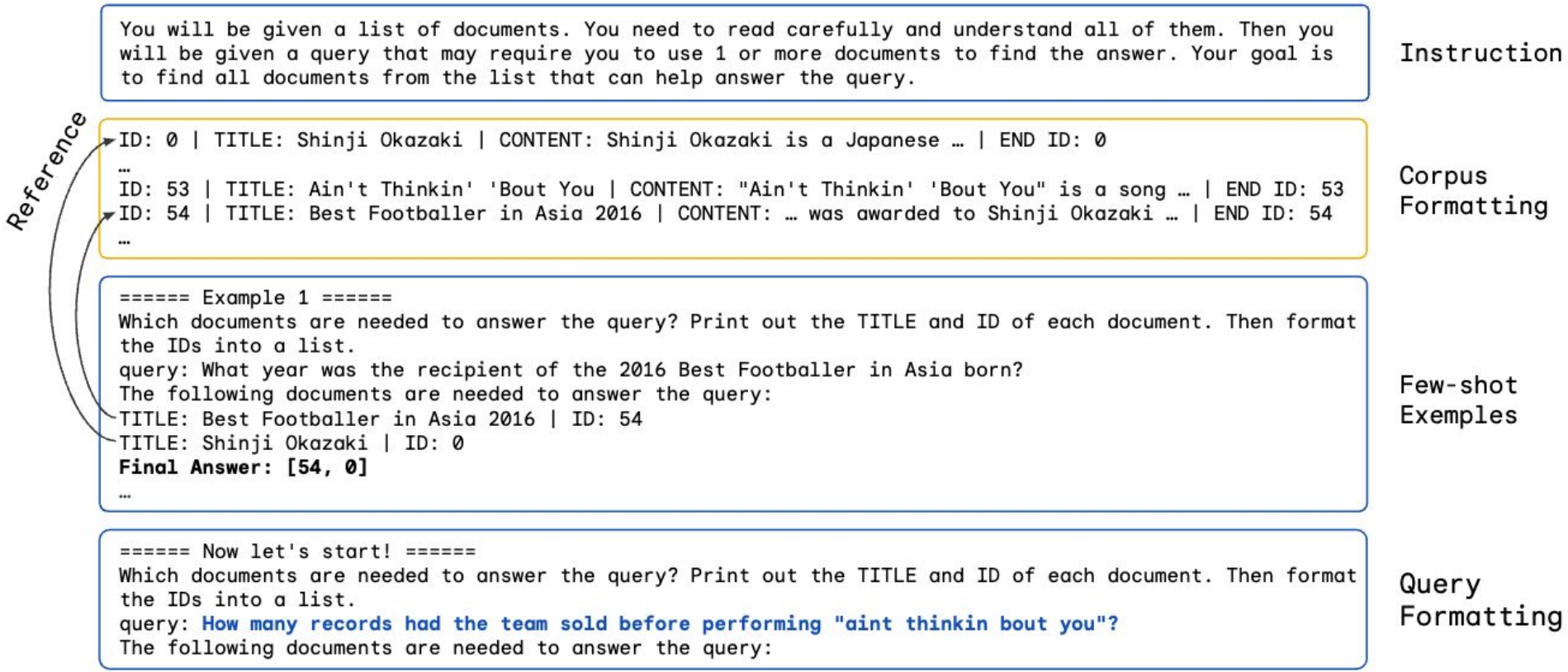
Corpus in Context Prompting:

1. Task specific instructions
2. Corpus Formatting: ID/Content/ID
3. Examples
4. Query Formatting

Prefill friendly - query is placed at the end

Carefully curated for rated context length

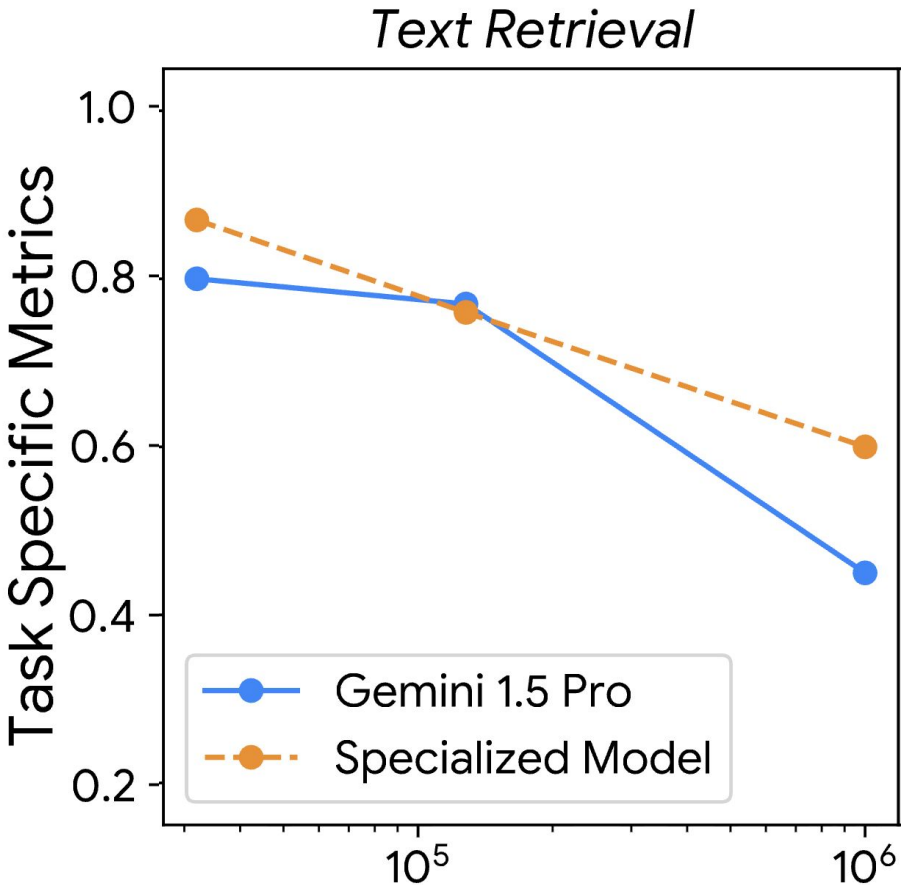
CIC Example


You will be given a list of documents. You need to read carefully and understand all of them. Then you will be given a query that may require you to use 1 or more documents to find the answer. Your goal is to find all documents from the list that can help answer the query.
Instruction
Reference
ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese … | END ID: 0
…
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song … | END ID: 53
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki … | END ID: 54
…
Corpus Formatting
====== Example 1 ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
TITLE: Best Footballer in Asia 2016 | ID: 54
TITLE: Shinji Okazaki | ID: 0
Final Answer: [54, 0]
…
Few-shot Exemples
====== Now let's start! ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: How many records had the team sold before performing "aint thinkin bout you"?
The following documents are needed to answer the query:
Query Formatting

LCLMs on Text Retrieval

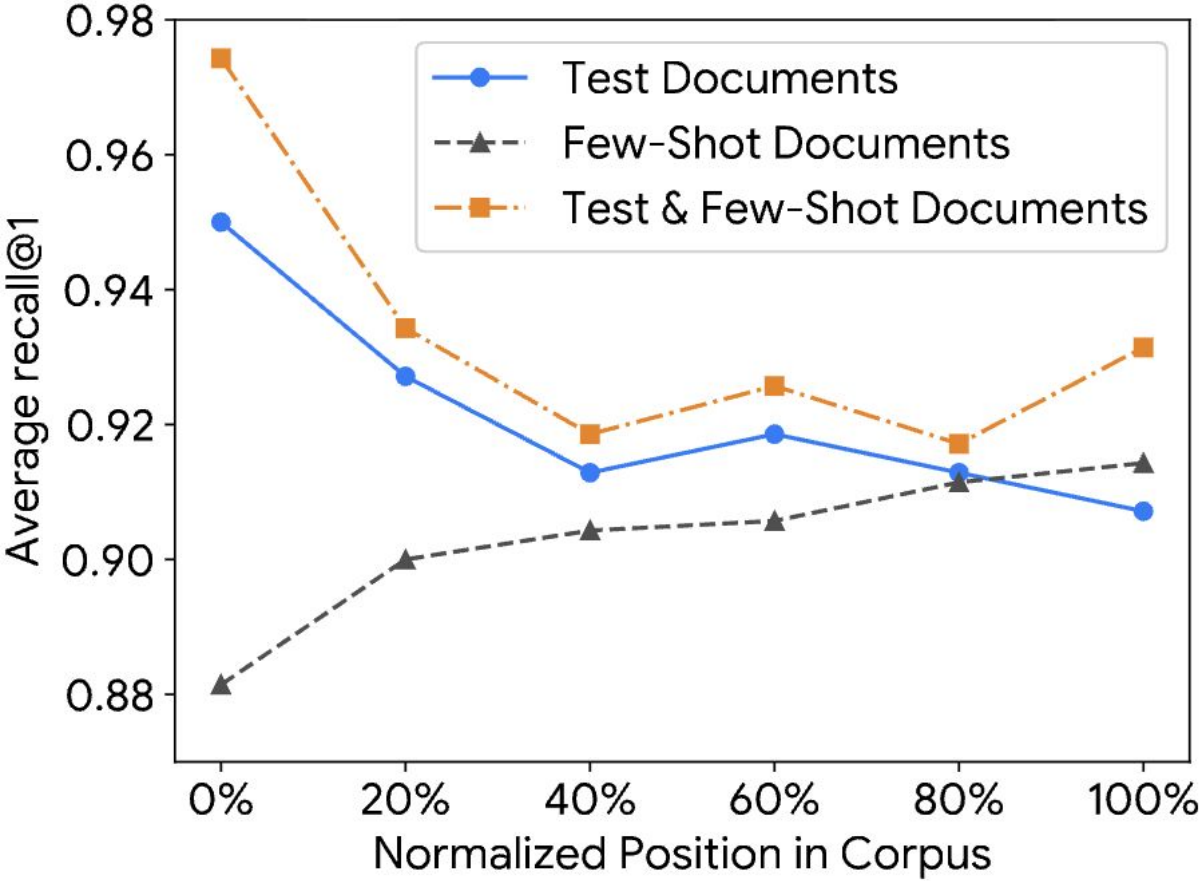
	Dataset	Gemini 1.5 Pro	GPT-4o	Claude 3 Opus	Specialized
Text Retrieval	ArguAna	0.84	0.85	0.74	0.75
	FEVER	0.98	0.96	0.94	0.97
	FIQA	0.79	0.82	0.61	0.83
	MS MARCO	0.95	0.87	0.93	0.97
	NQ	1.00	0.99	0.96	0.99
	Quora	0.93	0.93	0.94	1.00
	SciFact	0.88	0.88	0.73	0.85
	Touché-2020	0.91	0.88	0.71	0.88
	TopiOCQA	0.49	0.30	0.42	0.36
	HotPotQA†	0.90	0.82	0.83	0.92
	MuSiQue†	0.42	0.10	0.27	0.29
	QAMPARI†	0.61	0.18	0.20	0.57
	QUEST†	0.30	0.19	0.18	0.54
	Average	**0.77**	0.67	0.65	0.76

LCLMs on Text Retrieval 2



1. Baseline: Gecko, dual-encoder
2. Gemini 1.5 performs well, but context rot becomes a problem past 128k
3.

What if we move key documents around?



Hypotheses:

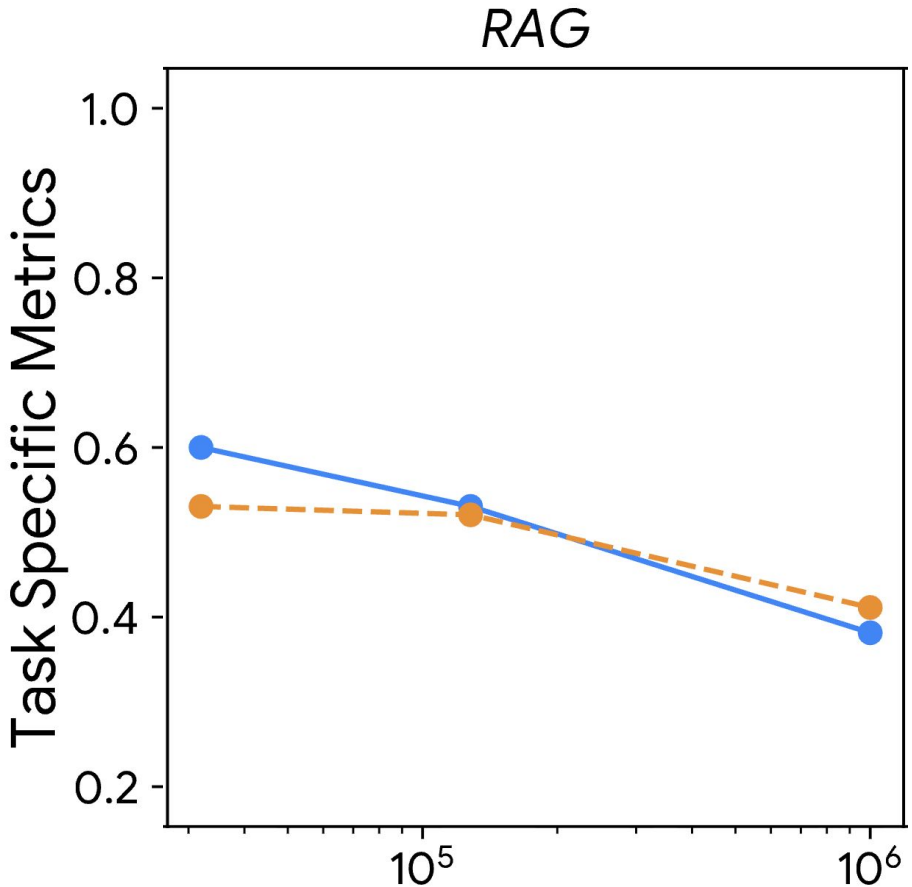
1. Attention sinks + Lost-in-the-middle effect

2. Few-shot docs actually impact position

LCLMs on RAG

	Dataset	Gemini 1.5 Pro	GPT-4o	Claude 3 Opus	Specialized
RAG	NQ	0.84	0.89	0.85	0.71
	TopiOCQA	0.34	0.33	0.37	0.35
	HotPotQA	0.75	0.72	0.74	0.70
	MuSiQue	0.55	0.47	0.45	0.45
	QAMPARI	0.44	0.27	0.25	0.55
	QUEST	0.28	0.20	0.15	0.35
	Average	**0.53**	0.48	0.47	0.52

LCLMs on RAG



1. Baseline: Gecko retrieves top 40 docs and feeds to Gemini
2. Strong performance by Gemini even at scale
3. Claim: more documents leads to better chain-of-thought, but context rot is still a problem

Quick Takeaways

1. Existing benchmarking for LCLMs is mediocre, “let’s do better”
2. Focusing on retrieval/RAG eval gets to the heart of what LCLMs do best: retrieve and generate

BUT

1. The actual benchmark is quite limited due to context size limitations (scalable)
2. Context scaling seems to be quite hard
   a. The paper is motivated by rated, not “real” context length
   b. Lot of intermediate problems that need to be addressed first
3. Is giving the model more information a good idea?
   a. Intuitively makes sense, but long context has its issues… more in the next paper

Long-Context LLMs Meet RAG: Overcoming Challenges for Long Inputs in RAG

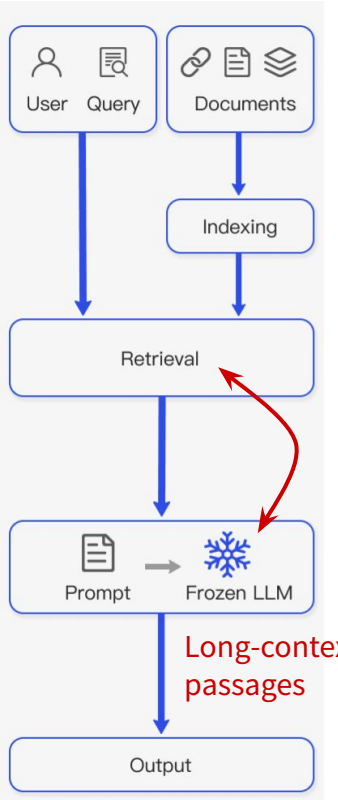
How to better combine Retriever with LLM?

Prior research:

- Tuning LLM for RAG on small retrieval set (usually <10 passages).
- Monotonic gain: more passages, better performance.

Key question:

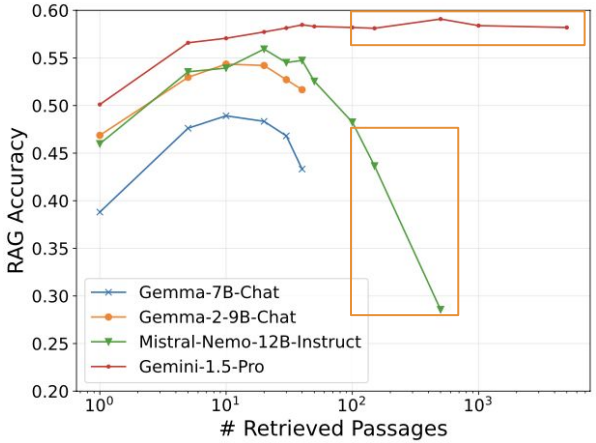
Does retrieving more passages always yield better performance?



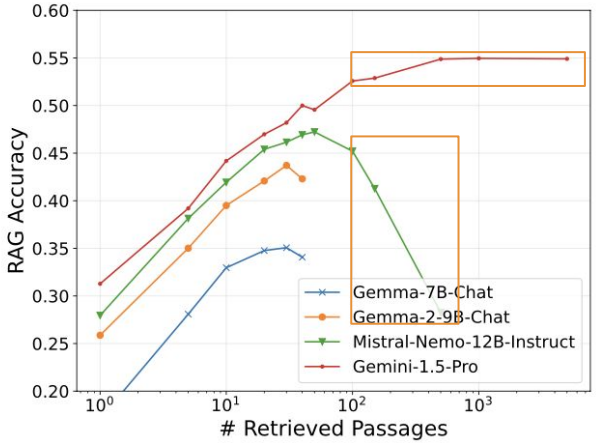
Contributions

1. Performance Analysis on Long-Input RAG
2. Robust Long-Input Methods
    a. Retrieval reordering
    b. Implicit fine-tuning with retrieval augmentation
    c. Explicit fine-tuning with reasoning
3. Fine-Tuning Strategy Analysis
    a. Data mixing
    b. Retriever transferability
    c. Retrieval-number ablation

Is stronger retriever always better?



(a) RAG performance with e5 retriever



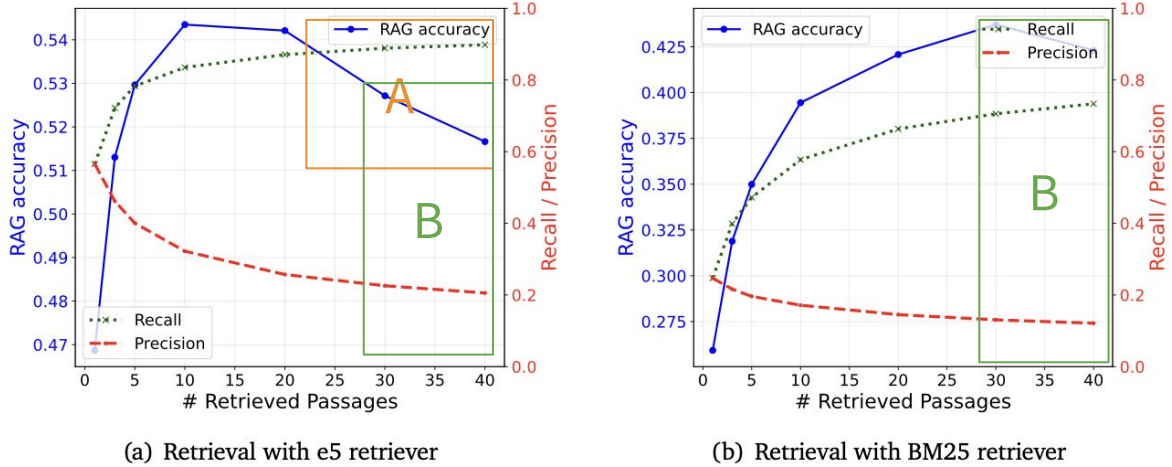
(b) RAG performance with BM25 retriever

1. "Inverted-U" performance trend
2. Stronger retriever: higher peak, sharper drop

BM25: sparse TF-IDF embedding; lexical match; no training.

e5: dense transformer embedding; cosine similarity; supervised contrastive training.

False promise of recall and precision



(a) Retrieval with e5 retriever

(b) Retrieval with BM25 retriever

irrelevant retrieved passages are defined as "*hard negatives*".

Conflict A: more passages → higher recall ↔ lower RAG acc

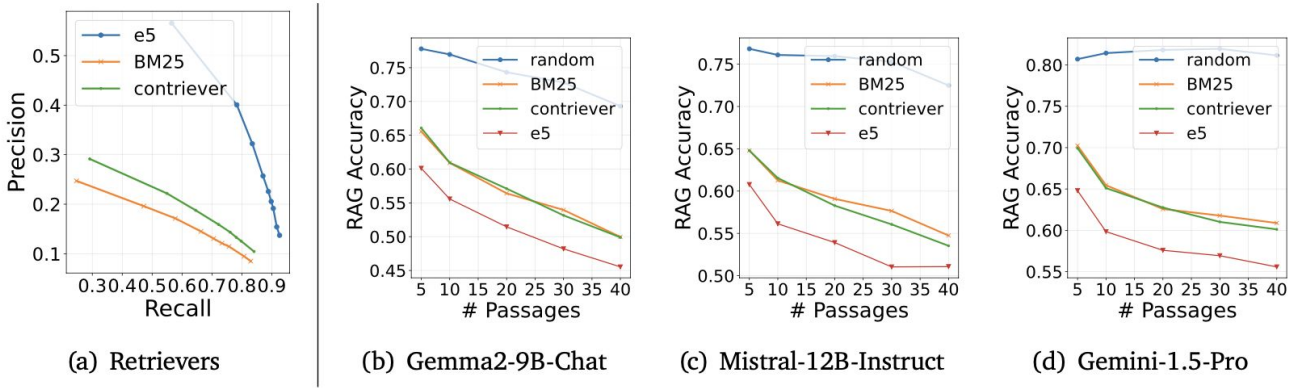
Irrelevant retrieved passages distract the LLM even with more relevant information present

Conflict B: less irrelevant passages → higher precision ↔ lower RAG acc

Type of irrelevant passages harms more than *quantity*.

Hard negatives hurts more than random samples

dataset construction (#k passages): [gold document] + (k-1)[retrieved document without answer]



(a) Retrievers (b) Gemma2-9B-Chat (c) Mistral-12B-Instruct (d) Gemini-1.5-Pro

Two types of "*hard negatives*" (irrelevant retrieved passages):

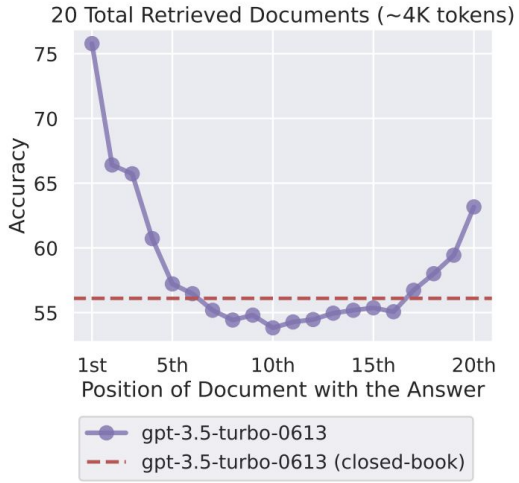
1. Related but Irrelevant (most distracting! portion-wise: E5 > bm25 > random)
2. Not Related

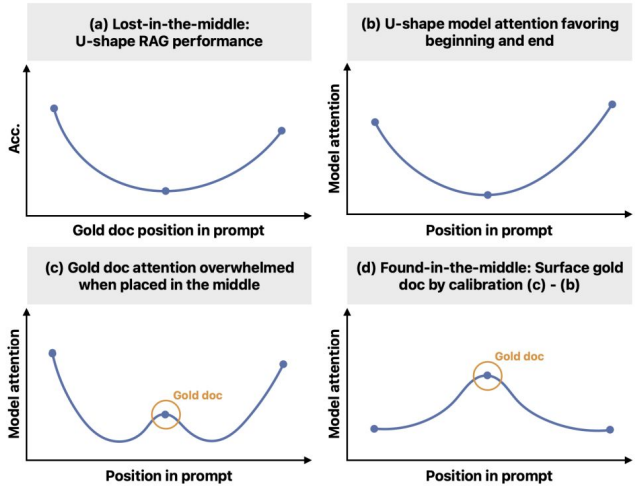
Simplest fix: reordering

"Lost in the middle" phenomenon in RAG

Root cause: positional bias in attention

What if we make use of that: amplify bias towards the end and suppress insured information in the middle?





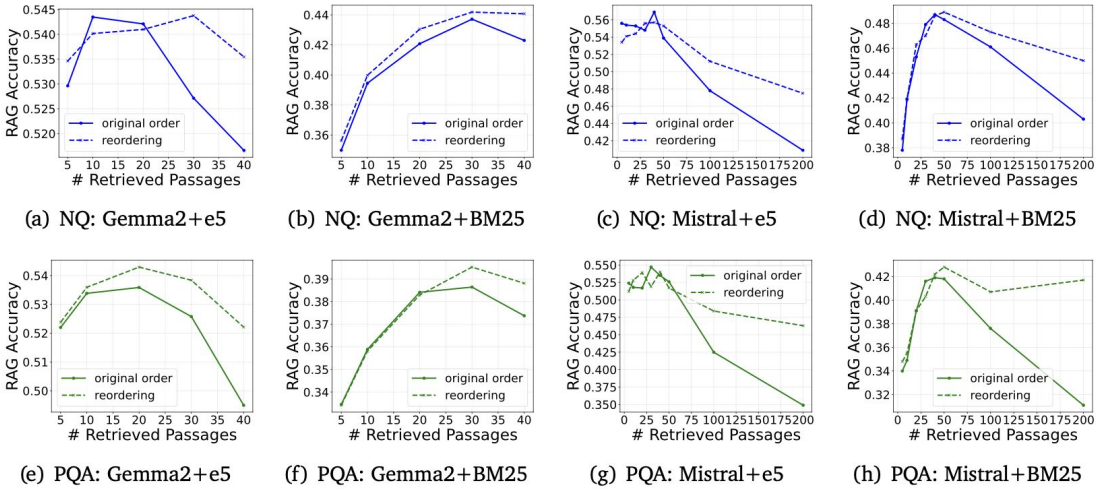
Found in the Middle [Hsieh, et al. 2024]

Simplest fix: reordering

[I, d1, d2, …, dk, q] → [I, d1, d3, …, d4, d2, q]

$$Order(d_i) = \begin{cases} \frac{i+1}{2} & \text{if } \mathrm{mod}(i, 2) = 1 \\ (k+1) - \frac{i}{2} & \text{if } \mathrm{mod}(i, 2) = 0 \end{cases}$$

‘U-shape’ of retrieval score



1. Simple reordering improve the performance
2. Consistency across dataset

More fundamental fix: fine-tuning

Training data

short-form
long-form
true/false
close-set

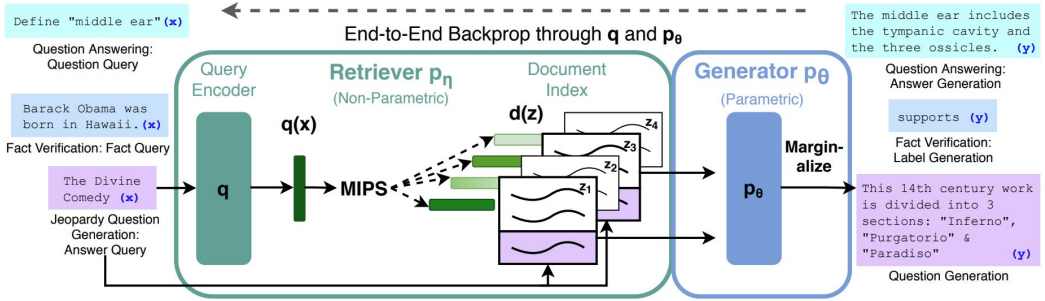
Dataset	the number of instances
Natural Question	12,500
Wizard of Wikipedia	12,500
FEVER	12,500
MMLU	12,500

Testing data

Dataset	Task	the number of instances
TriviaQA	QA	11,313
PopQA	QA	14,267
WebQuestions	QA	2,032
HotpotQA	Multi-Hop QA	7,405
2WikiMultiHopQA	Multi-Hop QA	12,576
Bamboogle	Multi-Hop QA	125
ASQA	Long-form QA	948
T-REx	Slot filling	5,000
Zero-shot RE	Slot filling	3,724

Fine-tuning as the norm before large LMs

Before LLMs: fine-tune retriever and generator (small LM) on domain specific task



RAG for NLP task [Lewis, et al. 2020]

After LLMs: fine-tunes large language models to ignore irrelevant information within retrieved passages (top-1 retrieval). [Yoran et al., 2023]

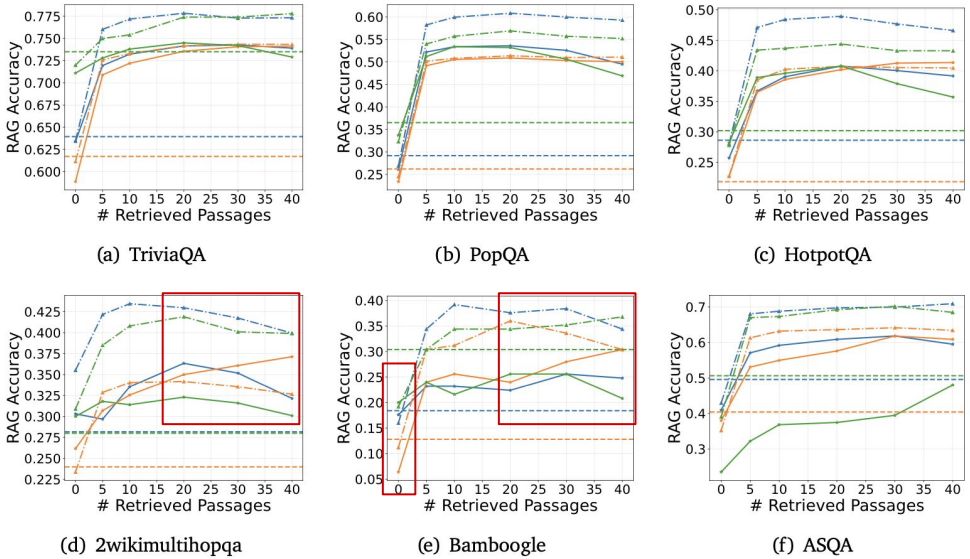
What's new here?: make use of context window of LCLM to handle large-scale retrieval (40 or more) during fine-tuning

Implicit retrieval augmented fine-tuning

Notation: Instruction I, query q, retrieved passages [d], target answer a

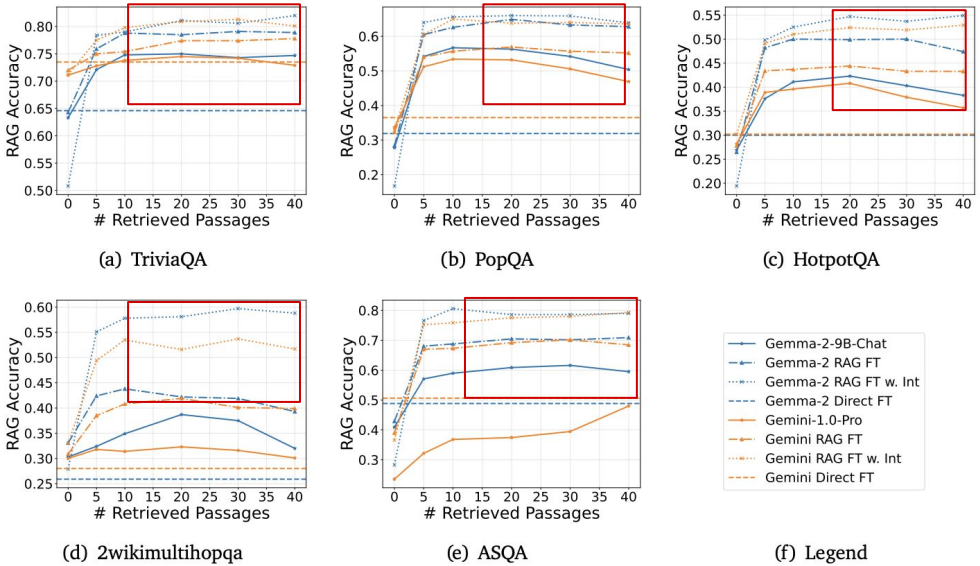
- Implicit RAG-specific fine-tuning:
  Input: [I, d1, d2, …, dk, q] → Output: a
- SFT baseline (simple Q&A):
  Input: [I, q]→ Output: a
- Gemma/Mistral: 4 epochs

  Gemini: 1 epoch



More robust to hard negatives

Explicit fine-tuning with reasoning

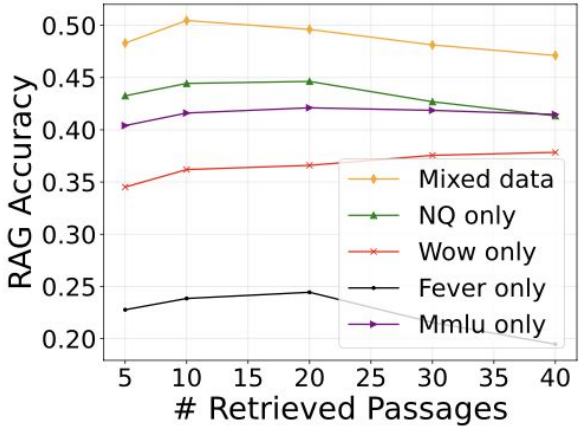


Explicit fine-tuning consistently perform better than implicit fine-tuning

Notation: Instruction I, query q, retrieved passages [d], target answer a, reasoning paragraph r

Explicit fine-tuning:
Input: [I, d1, d2, …, dk, q] → Output: [r, a]

Fine-tuning data strategy



Each dataset size is 50k

Mixed data: 12.5k from each

Test set: Hotpot QA

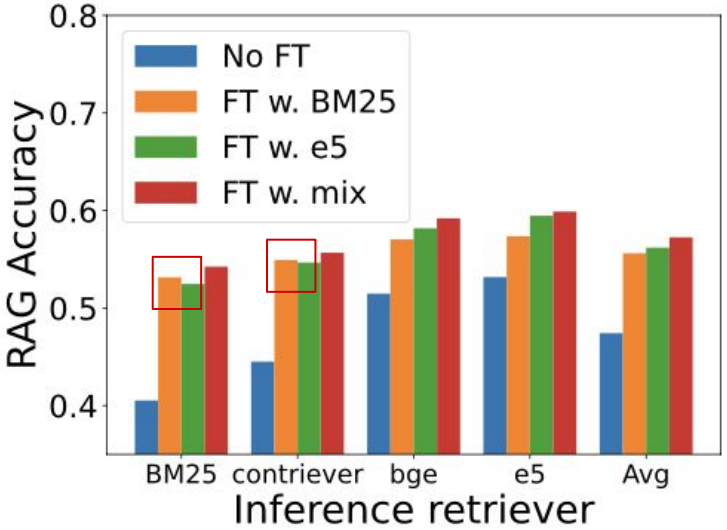
Datas diversity is important for RAG generalization

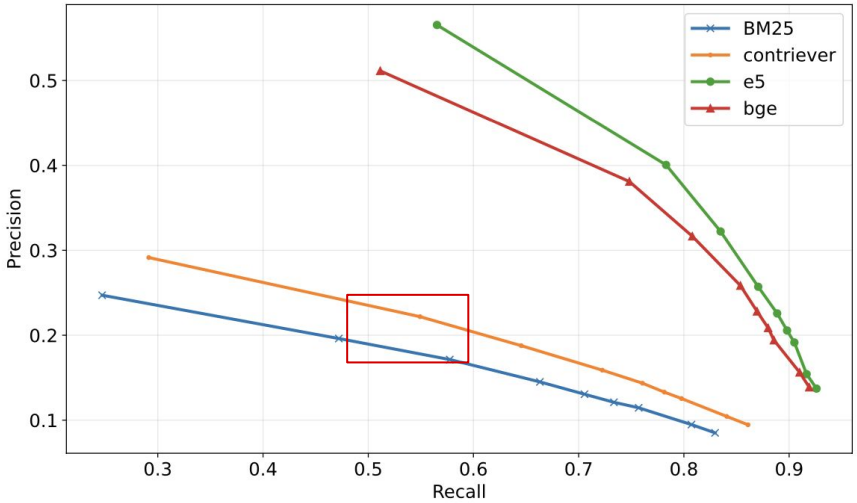
Number of Retrieval Passages	5k	20k	50k	200k
10	0.5942	0.5925	0.6058	0.6277
20	0.5909	0.5925	0.6078	0.6294
30	0.5787	0.5792	0.6072	0.6150
40	0.5582	0.5582	0.5859	0.5983
Avg.	0.5805	0.5806	0.6017	0.6176

Total samples count in mixed dataset

Increased training dataset size gives better performance

Does fine-tuning generalize across retrievers?

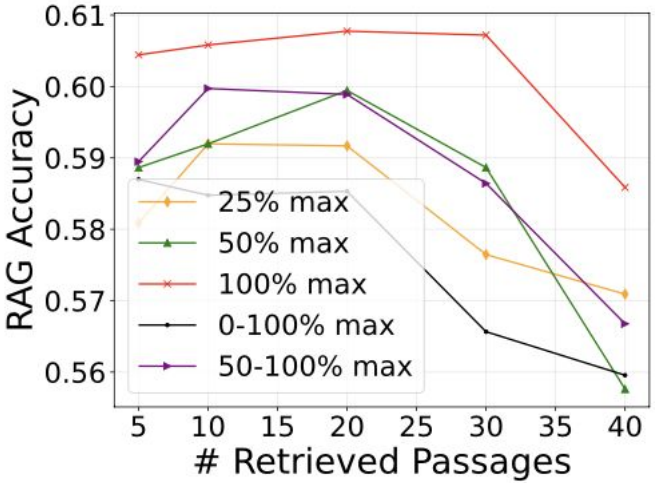




1. Mixing of retrievers during training gives the best performance.

2. Retriever similarity matters for generalization:
   E5 & bge; BM25 & contriever

Should we use full context window in training?



Fine-tuning with maximal input length is necessary.

Evaluation on Gemma-2-9B (maximal input length: 8192 tokens)

25% max: 10 retrieved passages

50~100% max: Dynamic 20 ~ 40 passages

Main Takeaways

More retrieved passages doesn’t always yield better results:

- “Hard-negative” passages is distracting LLM.
- Stronger retrievers fetch more related but irrelevant passages with larger retrieval sizes, causing sharper performance drops.

There are ways to fix:

- Hide less relevant context in the middle (reordering).
- Train LLM to ignore hard negatives (implicit fine-tuning).
- Train LLM to identify relevant passages (explicit reasoning fine-tuning).

Broader Impacts

Connections: Both papers explore how RAG and long-context LLMs complement each other.

- Paper 1: LCLMs have the capacity to process huge contexts and rival retrieval-based systems.
- Paper 2: performance actually declines as context grows: fine-tuning is needed to keep the model focused.

Best viewpoint:

Retrieval gives efficient coverage of a large external corpus.

LCLMs bring deep understanding and reasoning over the given context.

Long Context and Retrieval

Critics

Sidhika Balachandar & Prasann Singhal

10/21

Issue #1: Their naming sense

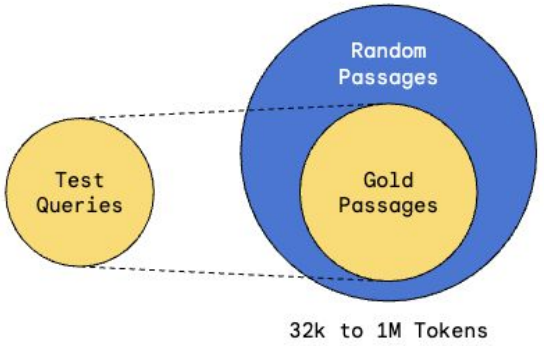
To address this, we introduce the **Lo**ng-Context **Fron**t**i**ers (LOFT) benchmark, a suite of six tasks consisting of 35 datasets which span text, visual, and audio modalities, to push LCLMs to their limits

context window. This unlocks a novel prompting-based approach for solving new and existing tasks, which we call **C**orpus-**i**n-**C**ontext prompting (**CiC**, pronounced "seek").

Issue #2: Unfair comparisons w/ retrieval

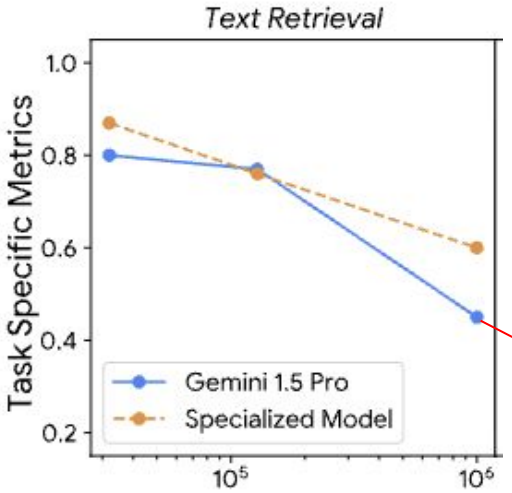
- Long-context models aren't actually long enough to handle more than small fractions of data-stores
- This isn't accounting for the much higher inference-time costs / time

All queries in each retrieval and RAG dataset share a single corpus, mimicking real retrieval applications. To create this shared corpus, we first include all gold passages from few-shot, development and the test queries, and then sample passages uniformly until reaching the desired context size (Figure 2). This construction ensures smaller corpora (e.g., 128k) are subsets of larger ones (e.g., 1M). Gold and random passages are shuffled to avoid positional biases. For fair comparison, specialized retriever models also use the same corpora for the evaluation.



Issue #2: Unfair comparisons w/ retrieval

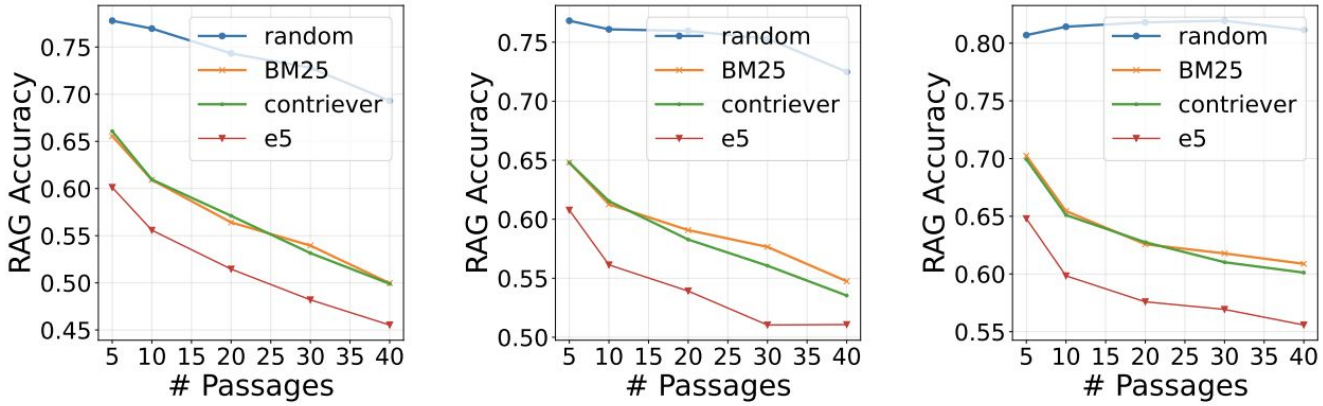
- More context lead to pretty big downgrades in performance



What might happen if you scale context more?

Issue #2: Unfair comparisons w/ retrieval

- Lee et al. only look at *random negatives*
- Jin et al. find that *hard negatives* reduce RAG accuracy



(b) Gemma2-9B-Chat (c) Mistral-12B-Instruct (d) Gemini-1.5-Pro

Issue #2: Unfair comparisons w/ retrieval

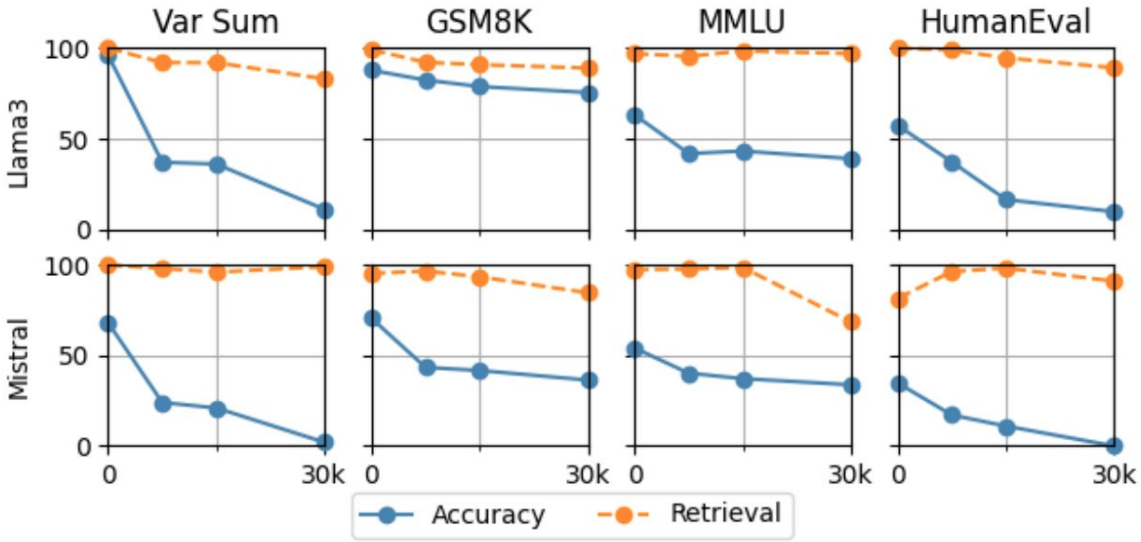
- Du et al. (2025): Even when the model can perfectly retrieve evidence, accuracy still drops as context length increases

Context Length Alone Hurts LLM Performance Despite Perfect Retrieval

Yufeng Du [1*], Minyang Tian [1*], Srikanth Ronanki [2], Subendhu Rongali [2], Sravan Bodapati [2], Aram Galstyan [2,3], Azton Wells[4], Roy Schwartz[5], Eliu A Huerta[4,6], Hao Peng[1]
[1]University of Illinois at Urbana-Champaign, [2]Amazon.com Inc.,
[3]USC Information Sciences Institute, [4]Argonne National Laboratory,
[5]The Hebrew University of Jerusalem, [6]University of Chicago

Issue #2: Unfair comparisons w/ retrieval

- Du et al. (2025): Even when the model can perfectly retrieve evidence, accuracy still drops as context length increases



Issue #3: 2/4 core contributions unconvincing

SQL	Spider	0.40	0.14	0.19	0.74
	SParC	0.36	0.13	0.21	0.55
	Average	0.38	0.13	0.20	**0.65**

Way worse than cheaper baseline

- **SQL**: We explore LCLMs' database querying and byp potentially enables more e Importantly, it can also be other types of structured dat graphs that often require be
- **Many-Shot ICL**: LCLMs context learning setup to hu of few-shot examples to use

No Clear Result? Prior work has already shown ICL is helpful. + missing error bars

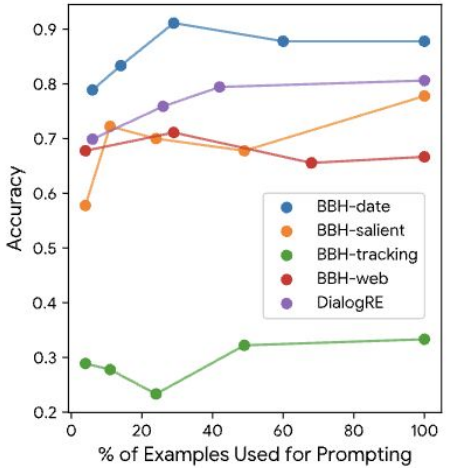


Figure 7: **ICL Performance** as we scale the percentage of examples used up to 100%.

Issue #4: Limited Evaluation

Task	Dataset	# Queries (Few-shot / Development / Test)	Supported Context Length	# Candidates
Text Retrieval	ArguAna	5 / 10 / 100	32k / 128k / 1M	123 / 531 / 3,891
	FEVER	5 / 10 / 100	32k / 128k / 1M	154 / 588 / 6,031
	FIQA	5 / 10 / 100	32k / 128k / 1M	148 / 531 / 4,471
	MS MARCO	5 / 10 / 100	32k / 128k / 1M	302 / 1,174 / 9,208
	NQ	5 / 10 / 100	32k / 128k / 1M	214 / 883 / 6,999
	Quora	5 / 10 / 100	32k / 128k / 1M	820 / 3,306 / 25,755
	SciFact	5 / 10 / 100	32k / 128k / 1M	86 / 357 / 2,753
	Touché-2020	5 / 10 / 34	32k / 128k / 1M	77 / 329 / 2,843
	TopiOCQA	5 / 10 / 100	32k / 128k / 1M	170 / 680 / 5,379
	HotPotQA	5 / 10 / 100	32k / 128k / 1M	319 / 1,222 / 10,005
	MuSiQue	5 / 10 / 100	32k / 128k / 1M	210 / 824 / 6,650
	QAMPARI	5 / 10 / 100	32k / 128k / 1M	186 / 755 / 5,878
	QUEST	5 / 10 / 100	32k / 128k / 1M	87 / 328 / 2,858

Tiny test sets and no error bars!

Everything w.r.t 1 closed model

Issue #4: Limited Evaluation

- Compare this with Jin et al. who evaluate across different LLMs, retrievers, and evaluation datasets
  - 3 different LLMs (Gemma, Mistral, Gemini)
  - 3 different retrievers (BM25, Contriever, e5)
  - Multiple large evaluation datasets (TriviaQA, PopQA, HotpotQA, etc.)

Issue #5: Role of Closed-Book Priors

Titles Only
-
-
0.91
0.84
0.10
0.09
0.05
0.23
0.39
0.09
0.02
0.30
(-0.30)

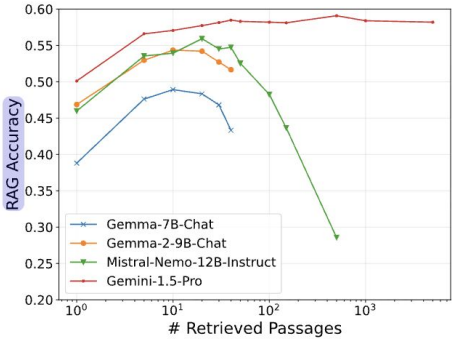
Close-book is worse, but still pretty good! Contamination? Are we really measuring text processing, or just elicitation of prior knowledge

Dataset	Dev (32k)	Test (128k)
NQ	0.60 (-0.10)	0.37 (-0.47)
HotPotQA	0.60 (-0.30)	0.33 (-0.42)
MuSiQue	0.20 (-0.60)	0.10 (-0.45)

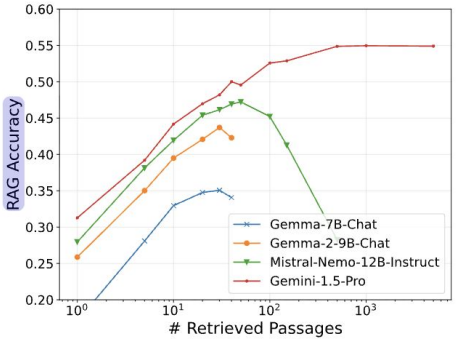
Table 3: **Gemini's closed-book performance on RAG**. Red indicates the performance difference compared to the CiC prompting.

Issue #6: Is Naive Context Scaling the right focus?

Close source / open source sort of in same ball-park? Both plateau / stop getting better at a pretty low number of documents. May want to focus on something else…



(a) RAG performance with e5 retriever



(b) RAG performance with BM25 retriever

Drowning in Documents: Consequences of Scaling Reranker Inference

Mathew Jacob[1,2,†], Erik Lindgren[1], Matei Zaharia[1], Michael Carbin[1], Omar Khattab[1] and Andrew Drozdov[1,*]

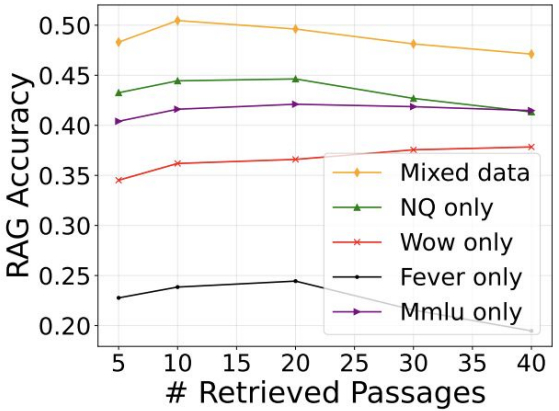
Issue #6: Is Naive Context Scaling the right focus?

- Jin et al. propose solutions for LCLMs other than just naive scaling:
  - Retrieval reordering
  - Implicit fine tuning (RAG FT)
  - Explicit reasoning (RAG FT w. Intermediate Reasoning)

Issue #6: Is Naive Context Scaling the right focus?

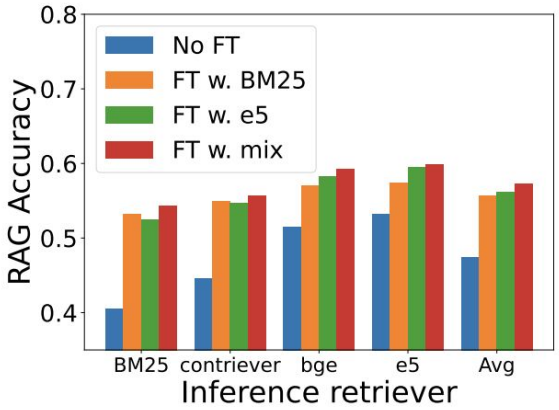
Jin et al. also provides advice on how to run the fine tuned models

What data to use?



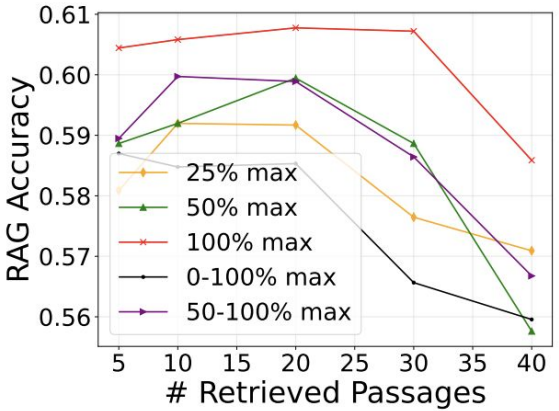
(a) Analysis of training data distribution. (Test: HotpotQA)

What retriever to use?



(b) Influence of retriever variations on fine-tuning effectiveness. (NQ)

How many passages to retrieve?



(c) Investigation of the optimal number of passages for training.

Defend for “Can Long-Context Subsume RAG”

Yichuan Wang

Contribution #1: Systematically compare RAG w LCLM

	Dataset	Gemini 1.5 Pro	GPT-4o	Claude 3 Opus	Specialized
Text Retrieval	ArguAna	0.84	0.85	0.74	0.75
	FEVER	0.98	0.96	0.94	0.97
	FIQA	0.79	0.82	0.61	0.83
	MS MARCO	0.95	0.87	0.93	0.97
	NQ	1.00	0.99	0.96	0.99
	Quora	0.93	0.93	0.94	1.00
	SciFact	0.88	0.88	0.73	0.85
	Touché-2020	0.91	0.88	0.71	0.88
	TopiOCQA	0.49	0.30	0.42	0.36
	HotPotQA†	0.90	0.82	0.83	0.92
	MuSiQue†	0.42	0.10	0.27	0.29
	QAMPARI†	0.61	0.18	0.20	0.57
	QUEST†	0.30	0.19	0.18	0.54
	Average	**0.77**	0.67	0.65	0.76
Visual Retrieval	Flickr30k	0.84	0.65	-	0.75
	MS COCO	0.77	0.44	-	0.66
	MSR-VTT	0.76	0.72	-	0.64
	OVEN	0.93	0.89	-	0.79
	Average	**0.83**	0.68	-	0.71
Audio Retrieval	FLEURS-en	1.00	-	-	0.98
	FLEURS-es	0.99	-	-	0.99
	FLEURS-fr	1.00	-	-	1.00
	FLEURS-hi	1.00	-	-	0.74
	FLEURS-zh	1.00	-	-	1.00
	Average	**1.00**	-	-	0.94
RAG	NQ	0.84	0.89	0.85	0.71
	TopiOCQA	0.34	0.33	0.37	0.35
	HotPotQA	0.75	0.72	0.74	0.70
	MuSiQue	0.55	0.47	0.45	0.45
	QAMPARI	0.44	0.27	0.25	0.55
	QUEST	0.28	0.20	0.15	0.35
	Average	**0.53**	0.48	0.47	0.52
SQL	Spider	0.40	0.14	0.19	0.74
	SParC	0.36	0.13	0.21	0.55
	Average	0.38	0.13	0.20	**0.65**
Many-Shot ICL	BBH-date	0.88	0.81	0.92	-
	BBH-salient	0.78	0.64	0.69	-
	BBH-tracking7	0.33	0.81	0.54	-
	BBH-web	0.67	0.57	0.83	-
	LIB-dialogue	0.76	0.67	0.72	-
	Average	0.68	0.70	**0.74**	-

1. Constructing a specialized pipeline for each task is tedious, but they manage to cover them all.

2. And the dataset they built can be reused annually to benchmark the progress of long-context capabilities versus advances in embedding models and indexing methods.

Contribution #2:Multimodal Coverage

4.2 Visual Retrieval

We employ CLIP-L/14 [47], a widely used text-to-image retrieval model, as our specialized model. For Flickr30k and MS-COCO, CLIP performs text-to-image retrieval. For MSR-VTT, it performs text-to-video retrieval by averaging scores across frames. For OVEN, due to the lack of suitable open-source image-to-text models, we approximate image-to-text retrieval by using CLIP's text-to-image retrieval. Evaluation of Claude 3 Opus on this task was not feasible due to the current limitation of 20 images per API request.

Results Gemini 1.5 Pro outperforms GPT-4o across all four visual benchmarks (Table 2). Notably, as shown in Figure 5, Gemini 1.5 Pro maintains a performance advantage over CLIP across all visual benchmarks and context lengths.

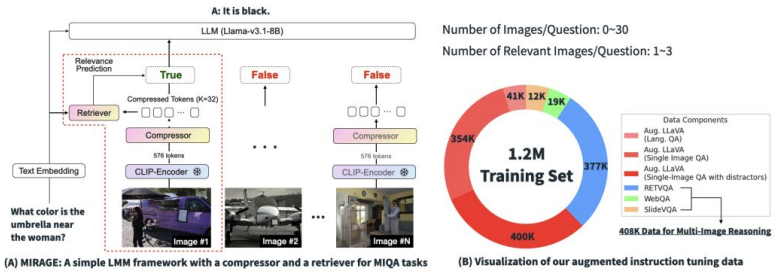
4.3 Audio Retrieval

We choose PaLM 2 DE [19] as a specialized model, which is a dual-encoder trained to maximize the similarity between audio and their transcription and has achieved previous state-of-the-art on the FLEURS datasets. Currently, GPT-4o and Claude 3 Opus do not support audio input.

Results Gemini 1.5 Pro demonstrates comparable performance to PaLM 2 DE across all 5 languages (Table 2). We notice that Gemini 1.5 Pro notably surpasses PaLM 2 DE in Hindi; this advantage likely stems from differences in pre-training data between Gemini and PaLM. Figure 5 further confirms Gemini 1.5 Pro's robust performance across various context length, highlighting the current capabilities of LCLMs while also indicating the need for more challenging audio datasets.

Method	RetVQA	VQAv2	GQA	TextVQA	POPE	MMB	MMB-CN	MME	SEED	MM-Vet
GPT-4o	34.6	77.2	-	78.0	87.2	-	-	1614.2	-	-
Gemini v1.5 Pro	32.2	73.2	-	73.5	88.2	-	-	1562.4	-	-
LLaVA-v1.5-7B	30.6	78.5	62.0	58.2	85.9	64.3	58.3	1510.7	58.6	31.1
LWM	-	55.8	44.8	18.8	75.2	-	-	-	-	9.6
MIRAGE-8.3B (Ours)	67.6	76.6	59.1	56.2	85.4	69.2	66.9	1437.9	59.0	33.4

VS



(A) MIRAGE: A simple LMM framework with a compressor and a retriever for MIQA tasks

(B) Visualization of our augmented instruction tuning data

Visual Haystacks

https://arxiv.org/pdf/2407.13766

Contribution #3: Practical advice

- Insight 1: LCLMs can replace retrieval/pipeline systems in many cases — but still struggle with deep compositional reasoning
  - When dealing with tasks requiring complex SQL-like reasoning (multiple operators, inequality, aggregation) the LCLMs still lagged significantly behind specialized pipelines. For example, in the SQL tasks the average accuracy for LCLMs was ~0.38 vs ~0.65 for the specialized pipeline
    - Challenge here: maybe the comparison is unfair
- Insight 2: Prompt structure and placement matter significantly when working with very long contexts
  - Similar to the second paper

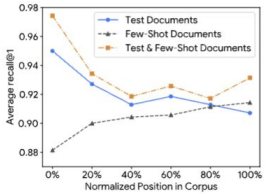


Figure 4: **Positional Analysis**. We vary gold document positions of queries within the corpus (0% = beginning, 100% = end).

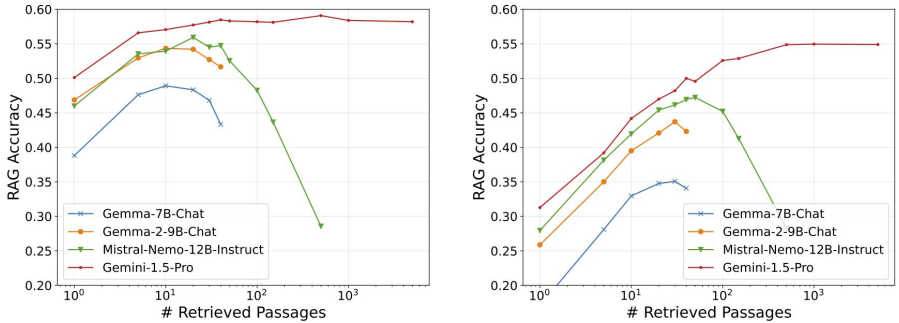
Exploiting this "lost-in-the-middle" behavior, we consider a simple and effective strategy: reordering the retrieved passages based on their relevance scores calculated by the retriever. Given a query q and a set of retrieved passages $d_1, d_2, \ldots, d_k$ with decreasing relevance scores, the standard input sequence construction for an LLM with instruction I would be: $[I, d_1, d_2, \ldots, d_{k-1}, d_k, q]$. Retrieval reordering modifies this to prioritize passages with higher scores at the beginning and end: $[I, d_1, d_3, \ldots, d_4, d_2, q]$ where the position of passage d_i is determined by

$$Order(d_i) = \begin{cases} \frac{i+1}{2} & \text{if } \mathrm{mod}(i, 2) = 1 \\ (k+1) - \frac{i}{2} & \text{if } \mathrm{mod}(i, 2) = 0 \end{cases} \quad (1)$$

Critic for “Long-Context Meets RAG”

Proponent: Critic for paper 2

- Demonstration of differences in different models. We might be able to distill the long context ability from the better models.
- Example, they may have reasoning patterns that is designed for integrating multiple sources.
- Not enough discussion



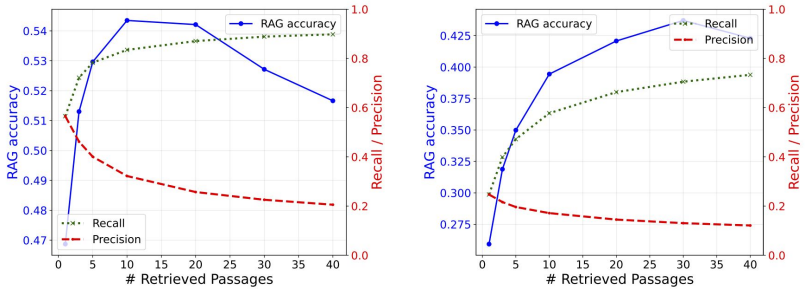
(a) RAG performance with e5 retriever

(b) RAG performance with BM25 retriever

Proponent

- Different turning points for different retrievers. Need to dig deeper

- This reveals more interesting patterns. Maybe BM25's accuracy turning point comes later because it is not very accurate so adding more passages still increases total amount of relevant documents retrieved?



(a) Retrieval with e5 retriever

(b) Retrieval with BM25 retriever