

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

ACL 2022

Jinhyuk Lee\*, Anthony Chen\*, Zhuyun Dai\*

Dheeru Dua, Devendra, Singh Sachan, Michael Boratko, Yi Luan,  
Sébastien M. R. Arnold, Vincent Perot, Siddharth Dalmia, Hexiang Hu,  
Xudong Lin, Panupong Pasupat, Aida Amini, Jeremy R. Cole,  
Sebastian Riedel, Iftekhhar Naim, Ming-Wei Chang, Kelvin Guu

**Google DeepMind**

## Part 1. Introduction

---

- **Long-context language models (LCLMs)**
  - Hold the promise of reshaping artificial intelligence by enabling entirely new tasks and applications
  - Eliminate the reliance on tools and complex pipelines previously necessary due to context length limitations
  - Consolidate complex pipelines into a unified model:  
LCLMs ameliorate issues
    - Cascading errors, cumbersome optimization: A streamlined end-to-end approach
    - Adding instructions
    - Incorporating few-shot examples
    - Leveraging demonstrations via chain-of-thought prompting

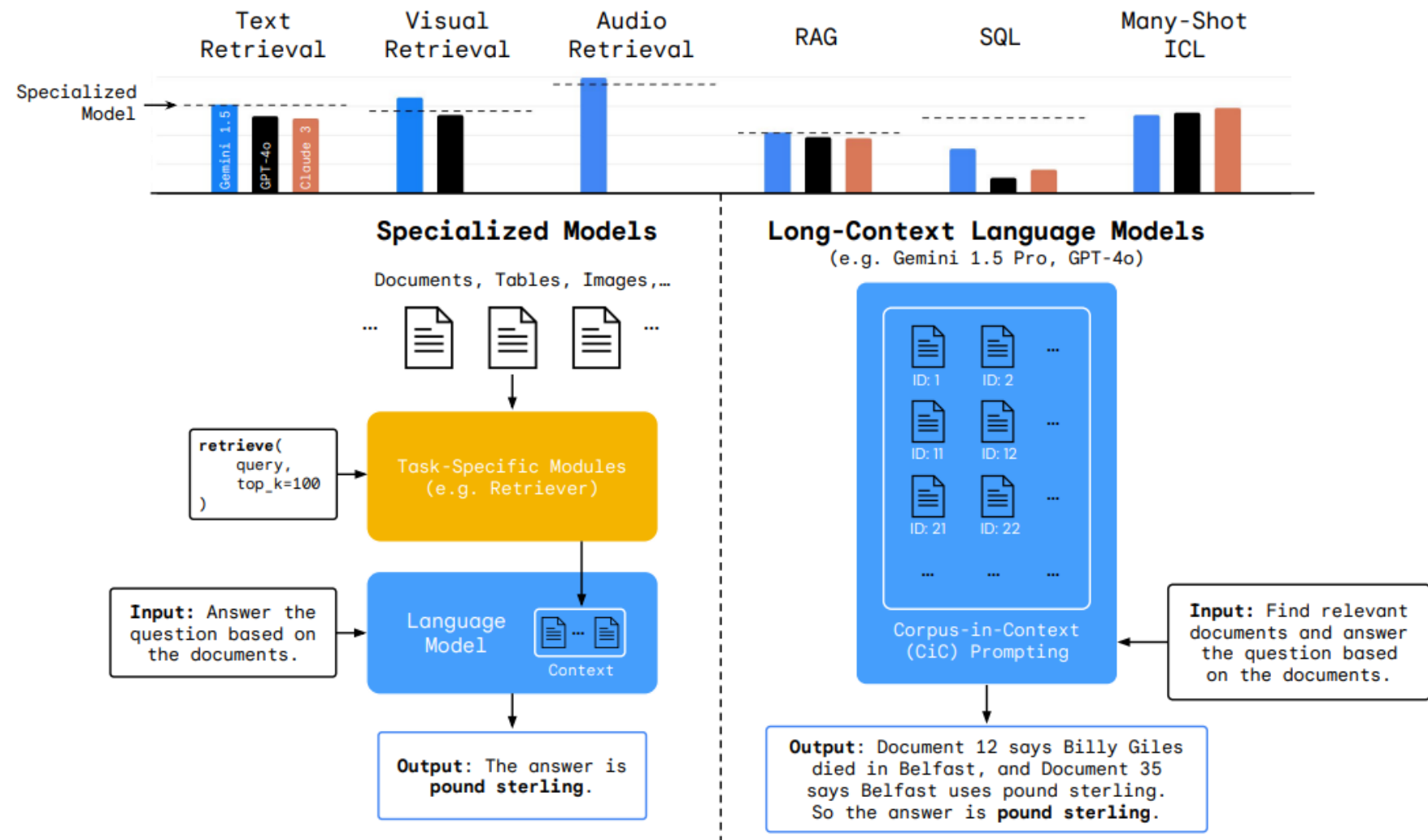
## Part 1. Introduction

---

- **Long-context language models (LCLMs)**
  - Require rigorous evaluation on truly long-context tasks useful in real-world applications
  - Existing benchmarks
    - Rely on synthetic tasks like the popular “needle-in-haystack” or fixed-length datasets
    - Fail to keep pace with the evolving definition of “long-context”
  - Critically, existing evaluations do not adequately stress-test LCLMs on any paradigm-shifting tasks

## Part 1. Introduction

- Long-context language models (LCLMs)



## Part 1. Introduction

---

- **Long-Context Frontiers (LOFT) benchmark**
  - A suite of six tasks consisting of 35 datasets which span text, visual, and audio modalities
  - Allow for automatic creation of increasing context lengths
  - Ensure that rigorous evaluation as LCLMs continue to scale
    - While the current version extends to one million tokens, it can easily be extended further to tens of millions

## Part 1. Introduction

---

- **LOFT focuses on the following areas where LCLMs have the potential for disruption**
  - Retrieval
    - Directly ingest & retrieve information from a corpus, without separate dual-encoder models
    - Retrieval systems such as multi-hop reasoning, instruction following, few-shot task adaptation
  - Retrieval-Augmented Generation (RAG)
    - Simplify RAG pipelines by directly reasoning over a corpus
    - Overcome challenges like query decomposition & mitigating cascading errors due to retrieval misses
  - SQL
    - Process entire databases as text, enabling natural language database querying and bypassing conversion to a formal query language like SQL
  - Many-Shot ICL
    - Scale the number of examples from the tens in the traditional incontext learning setup to hundreds or thousands
    - Remove the need to find the optimal set of few-shot examples to use

## Part 1. Introduction

---

- **LOFT reveals several key insights when comparing state-of-the-art LCLMs**
  - At the 128k token level, the largest size comparable across all models
    - LCLMs rival the performance of Gecko, a leading textual retrieval system
  - LCLMs lag significantly on complex multi-hop compositional reasoning tasks, indicating substantial room for improvement
    - LCLMs rival the performance
  - Reveal large performance variance depending on prompting strategies such as chain-of-thought reasoning
    - Underscore the need for further research to enhance LCLMs robustness and instructability
  - Match the performance of many specialized models
  - Reveal ample headroom for improvement in robust long-context reasoning as context windows continue to scale

## Part 2. LOFT: A 1 Million+ Token Long-Context Benchmark

### • Tasks range from retrieving relevant documents

Task	Dataset	Description	Avg. Cand. Length	# Cand. (128k)	Candidates	Input	Target
<b>Text Retrieval</b>	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			
<b>Visual Retrieval</b>	Flickr30k	Image Retrieval	258	440	Images	Text Query	Image ID
	MS COCO	Image Retrieval	258	440	Images	Text Query	Image ID
	OVEN	Image-text Retrieval	278	448	Images+Texts	Image+Text Query	Wikipedia ID
	MSR-VTT	Video Retrieval	774	140	Videos	Text Query	Video ID
<b>Audio Retrieval</b>	FLEURS-en	Audio Retrieval	249	428	Speech	Text Query	Speech ID
	FLEURS-es		315	343			
	FLEURS-fr		259	412			
	FLEURS-hi		292	369			
	FLEURS-zh		291	370			
<b>RAG</b>	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			
<b>SQL</b>	Spider	Single-turn SQL	111k	1	SQL Database	Question	Answer
	SParC	Multi-turn SQL	111k	1			
<b>Many-Shot ICL</b>	BBH-date	Multiple-choice QA	131	150	Training Examples	Question	Answer
	BBH-salient	Multiple-choice QA	246	104			
	BBH-tracking7	Multiple-choice QA	205	123			
	BBH-web	Multiple-choice QA	43	150			
	LIB-dialogue	Classification	266	274			

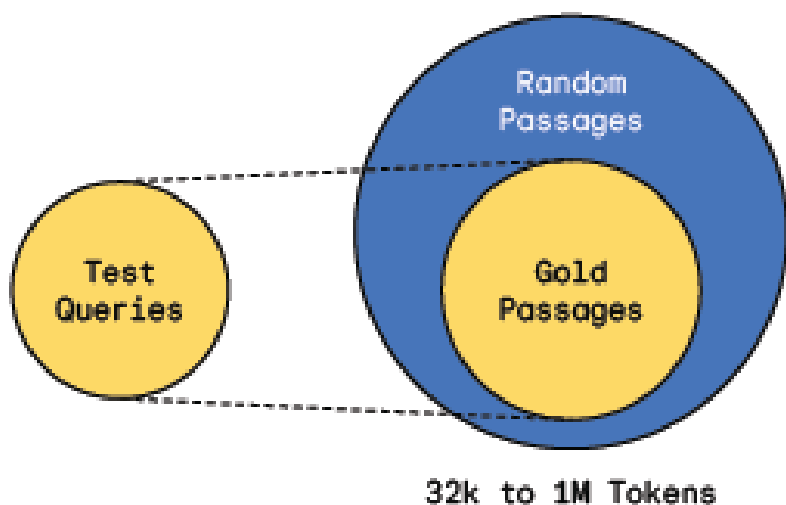
- 6 types of tasks, 4 modalities, and 35 datasets in total
- Sample up to 100 test queries, 5 few-shot queries, and 10 development queries
- Create LOFT with three different context length limits, namely 32k2, 128k, and 1M



# LOFT: A 1 Million+ Token Long-Context Benchmark

## • Retrieval & RAG

- Dataset share a single corpus, mimicking real retrieval applications
- Create this shared corpus
  - Include all gold passages from few-shot, development and the test queries
  - Then sample passages uniformly until reaching the desired context size
  - Ensures smaller corpora (e.g., 128k) are subsets of larger ones (e.g., 1M)
- Gold and random passages are shuffled to avoid positional biases
- Fair comparison:, specialized retriever models use the same corpora for the evaluation



- Corpus creation for retrieval and RAG
  - Given a set of test queries, we use their associated gold passages and other random passages to form the corpus

## Part 2. LOFT: A 1 Million+ Token Long-Context Benchmark

---

- **Many-shot ICL**

- Evaluate many-shot in-context learning (ICL) capabilities
- Adapt datasets from Big-Bench Hard (BBH) & LongICLBench (LIB)
- Construct shared many-shot ICL contexts
- Ensure training examples in smaller contexts are included in larger ones
- All of the many-shot ICL datasets are classification tasks
  - Guarantee that each class is represented at least once

# LOFT: A 1 Million+ Token Long-Context Benchmark

---

- **SQL**

- Spider, a single-turn text-to-SQL dataset, and SparC, its multi-turn variant
- Select the largest databases that will fit into that context
  - The databases for the 1M token setting would not fit into the smaller context length
  - Unlike most of the other tasks that share a corpus, the query sets differ across LOFT sizes
- A maximum context length of  $N \in \{32k, 128k, 1M\}$
- Create a corpus up to a size of  $0.9N$ , to account for differences in tokenizers,

# Corpus-in-Context Prompting (CiC, pronounced "seek")

---

- **Definition**

- Direct ingestion and processing of entire corpora within their context window
- Unlock a novel prompting-based approach for solving new and existing tasks

- **Prompt Design**

- Instructions
  - Provide task-specific instructions to guide the LCLM's behaviors
  - Ask the model to read the corpus carefully and find relevant documents to answer the question
- Corpus Formatting
  - Each candidate (e.g., passage, image, audio) in a corpus is assigned a unique identifier (ID)
  - e.g., Putting document IDs both before and after the passage in text retrieval
  - Mitigate the effects of causal attention in decoder-only LCLMs and enhance retrieval accuracy

# Corpus-in-Context Prompting (CiC, pronounced "seek")

---

- **Prompt Design**

- Few-Shot Examples

- Unlike common approaches where fewshot examples are independent, we ground all examples to the same corpus, aiming to teach the model to also learn more details about the specific corpus it needs to use
    - To facilitate automated evaluation, answers within each few-shot example are formatted as a list (e.g., "Final Answer: [54, 0]" in Figure 3), thus guiding the model to generate responses in a similar structure that can be readily parsed and compared against ground truth labels
    - Each few-shot example is accompanied by a Chain-of-Thought reasoning

- Query Formatting

- The query to be evaluated is formatted similar to the few-shot examples
    - Multi-turn dataset: Prepend previous query turns and model outputs to the current query turn, ensuring that the model's generation is conditioned on its prior responses
    - Based on our query formatting, LCLMs generate tokens that are parsed into the final answer

# Corpus-in-Context Prompting (CiC, pronounced "seek")

## • Example of Corpus-in-Context Prompting for retrieval

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

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

Reference

# Corpus-in-Context Prompting (CiC, pronounced "seek")

---

- **Design Consideration**

- To accommodate this diversity, we allocate ample space for prompt customization
- Strongly recommend that for each maximum context length of LOFT (e.g., 32k or 128k)
- Recommend to evaluate models on the maximum size that can fit into their context length without truncating the corpus or any of the individual examples

- **Discussion on Efficiency**

- Encoding a one million token context can be slow and computationally expensive
- Compatibility with prefix-caching in autoregressive language models as the query appears at the end of the prompt
  - The corpus only needs to be encoded once, similar to the indexing process in traditional information retrieval

# LOFT Tasks and Primary Results

---

- **Specialized model in LCLM benchmark**

- Evaluate the state-of-the-art LCLMs on LOFT
  - Google's Gemini 1.5 Pro, OpenAI's GPT-4o, and Anthropic's Claude 3 Opus
- LCLMs without any task-specific fine-tuning
  - These are benchmarked against specialized models that have undergone extensive fine-tuning or pipelining for the target task
  - These are limited to that specific domain
  - Select each specialized model that exemplifies recent task-specific advancements



# LOFT Tasks and Primary Results

## • Text Retrieval

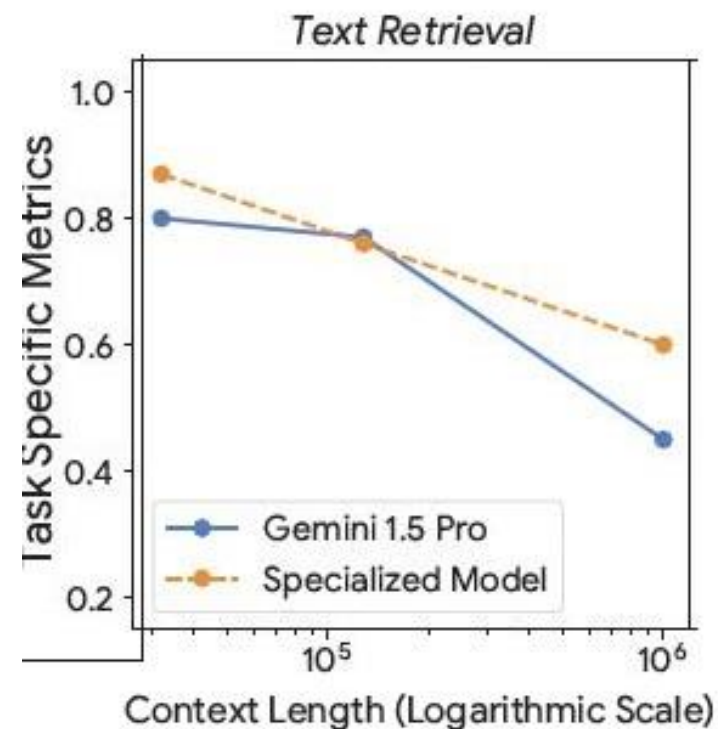
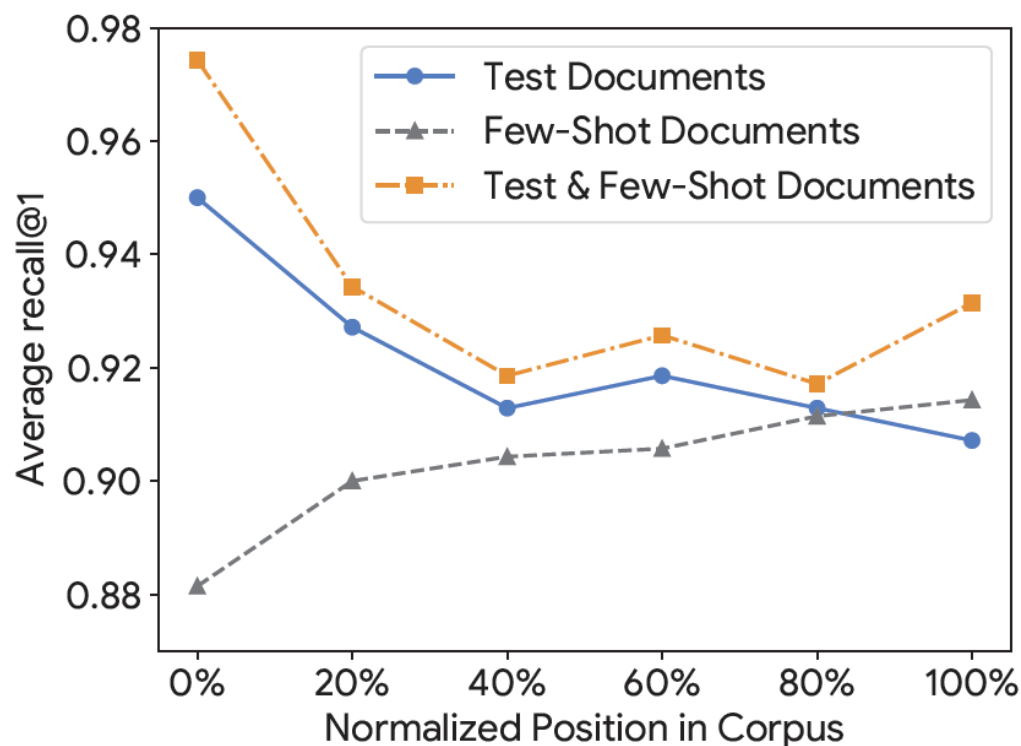
- Adopt Gecko, a state-of-the-art dual encoder as the specialized model for the retrieval task
- Gemini 1.5 Pro performs comparably to Gecko at 128k context length

	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 <sup>†</sup>	0.90	0.82	0.83	0.92
	MuSiQue <sup>†</sup>	0.42	0.10	0.27	0.29
	QAMPARI <sup>†</sup>	0.61	0.18	0.20	0.57
	QUEST <sup>†</sup>	0.30	0.19	0.18	0.54
	Average	<b>0.77</b>	0.67	0.65	0.76

# LOFT Tasks and Primary Results

## • Text Retrieval: Positional Analysis

- Performance drops as the gold documents of the test queries are moved towards the end of the corpus, suggesting reduced attention in later sections of the prompt
- Co-locating gold documents of few-shot and test queries consistently boosts performance

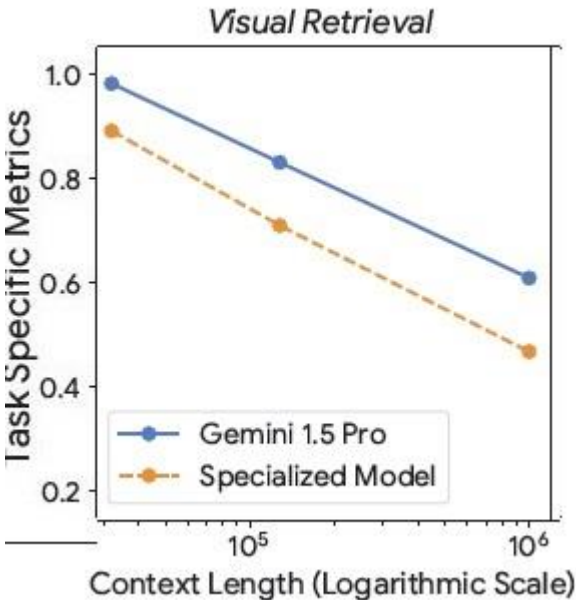


Part 4.

LOFT Tasks and Primary Results

- **Visual Retrieval**
  - Employ CLIP-L/14, a widely used text-to-image retrieval model
  - Flickr30k & MS-COCO: CLIP performs text-to-image retrieval
  - MSR-VTT: Perform text-to-video retrieval by averaging scores across frames
  - OVEN: 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

	Dataset	Gemini 1.5 Pro	GPT-4o	Claude 3 Opus	Specialized
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

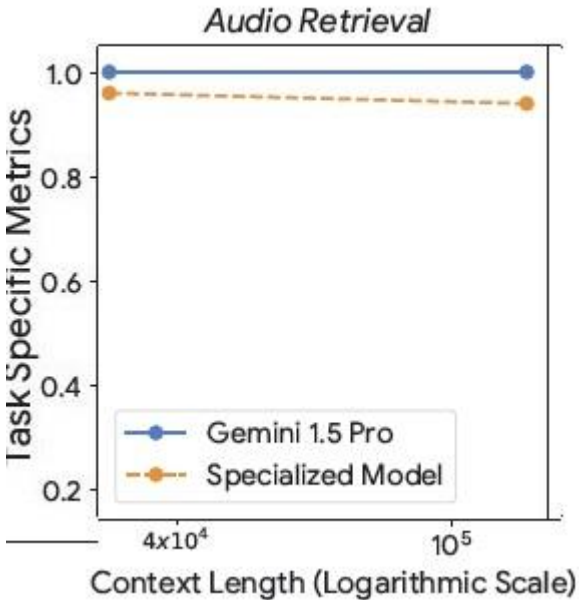


Part 4.

LOFT Tasks and Primary Results

- **Audio Retrieval**
  - PaLM 2 DE as a specialized model
    - A dual-encoder trained to maximize the similarity between audio and their transcription
    - Achieve previous state-of-the-art on the FLEURS datasets
  - Currently, GPT-4o and Claude 3 Opus do not support audio input

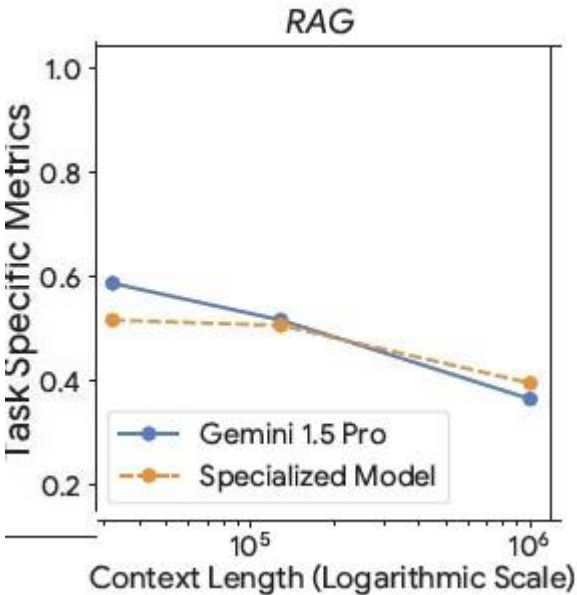
Dataset		Gemini 1.5 Pro	GPT-4o	Claude 3 Opus	Specialized
Audio Retrieval	FLEURS-en	1.00	-	-	0.98
	FLEURS-es	0.99	-	-	0.99
	FLEURS-fr	1.00	-	-	1.00
	FLEURS-hi	1.00	-	-	0.74
	FLEURS-zh	1.00	-	-	1.00
	Average	1.00	-	-	0.94



# LOFT Tasks and Primary Results

- **RAG**
  - A retrieve-and-read RAG pipeline as a specialized model, using Gecko
    - Retrieve the top-40 documents which are then put into the context of Gemini 1.5 Pro and used to generate the answer conditioned on the question and the retrieved documents
  - Reason over multiple passages in the context window using Chain-of-Thought
  - A capability that RAG pipelines typically lack unless they have a separate module for planning and reasoning

	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



# LOFT Tasks and Primary Results

- **RAG**

- Closed-book ablations on Gemini 1.5 Pro
- Remove the corpus from the context to assess LCLM performance based solely on parametric knowledge
- Reveal that the closed-book performance significantly lags behind the long-context and specialized models
- Underscore the tested models' effectiveness in leveraging the external corpus to enhance its reasoning capabilities

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)

## Gemini's closed-book performance on RAG

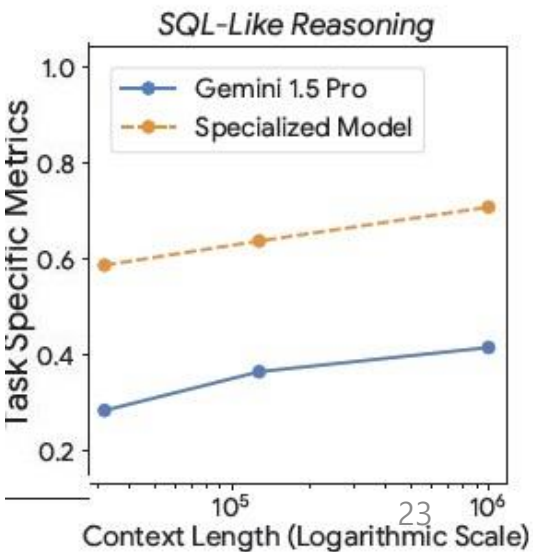
- Red indicates the performance difference compared to the CiC prompting

Part 4.

LOFT Tasks and Primary Results

- **SQL-Like Compositional Reasoning**
  - The traditional SQL pipeline
    - Use a trained semantic parser to translate the natural language input into a SQL query
  - A separate SQL interpreter
    - Execute the SQL query over the database
  - A specialized model
    - Use DAIL-SQL for the semantic parser
    - Prompts an LLM to provide the SQL query. We adapt DAIL-SQL by replacing its LLM with Gemini 1.5 Pro and using a fixed set of few-shot examples

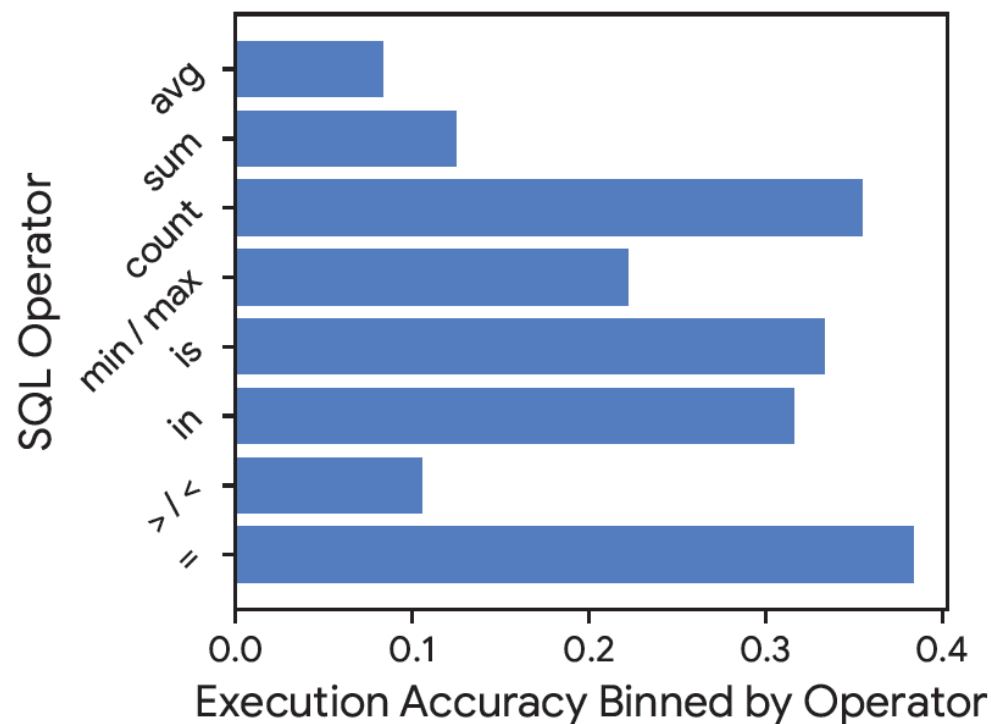
	Dataset	Gemini 1.5 Pro	GPT-4o	Claude 3 Opus	Specialized
SQL	Spider	0.40	0.14	0.19	0.74
	SParC	0.36	0.13	0.21	0.55
	Average	0.38	0.13	0.20	<b>0.65</b>



# LOFT Tasks and Primary Results

- **SQL-Like Compositional Reasoning**

- Categorize queries based on the operators in the gold SQL queries and measure Gemini 1.5 Pro's performance for each operator
- Averaging is the most difficult operation while counting is relatively easy
- Reasoning over equality is considerably easier than reasoning over inequality



## SQL Reasoning Analysis

- Bin Spider queries by operators in their SQL query and report binned Gemini performance
- Group min and max into a bin and > and < into another bin



# LOFT Tasks and Primary Results

## • Many-Shot ICL

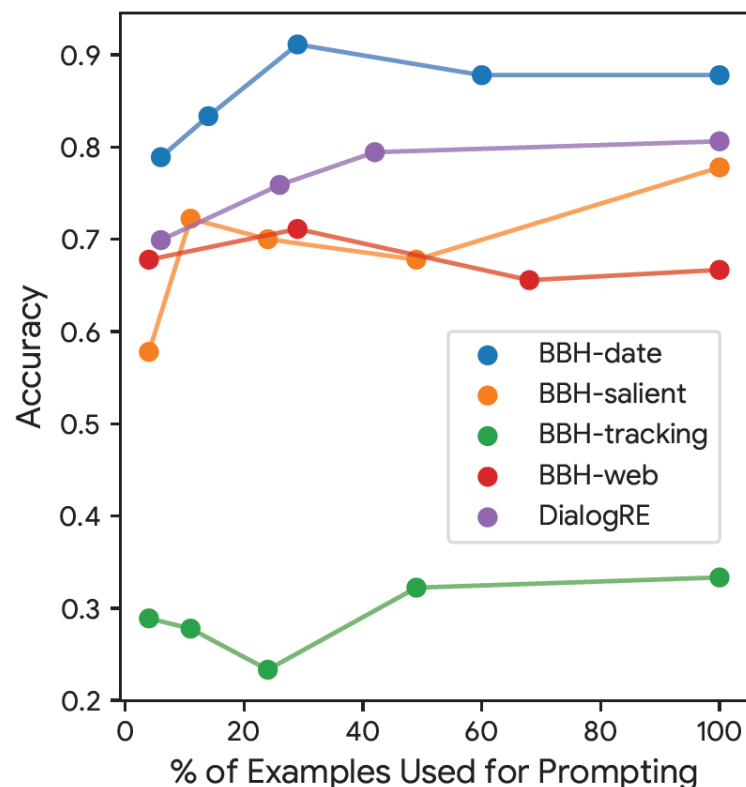
- Gemini 1.5 Pro outperforms GPT-4o on all benchmarks
- Claude 3 Opus achieves the best performance among LCLMs on this task
- BBH: Report the accuracy on 32k, which is the maximum context length available
- BBH-tracking7: Gemini 1.5 Pro performs surprisingly more poorly than GPT-4o

	Dataset	Gemini 1.5 Pro	GPT-4o	Claude 3 Opus	Specialized
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	<b>0.74</b>	-

# LOFT Tasks and Primary Results

## • Many-Shot ICL

- BBH-date & BBH-salient: See monotonic improvements similar to LIB-dialog
- BBH-tracking7 & BBH-web: Reasoning-intensive tasks like do not benefit
- More complicated tasks may see an earlier limit in how much models can learn from scaling the number of in-context examples



### Many-Shot ICL Performance

- Scale the percentage of examples used up to 100%
- The impact of increasing the number of examples in Gemini

## Part 5. CiC Prompt Ablations

- **Original CiC prompt for HotPotQA, a retrieval dataset in LOFT**
  - Evaluate Gemini 1.5 Pro on the 128k version of LOFT
  - The prompt contains an instruction, a corpus, few-shot examples and a query

```
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

```
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

## Part 5. CiC Prompt Ablations

- **Removing task-specific instructions (Generic Instruction)**
  - It leads to worse performance
  - Each few-shot example has its own small corpus consisting of nine random passages and one gold passage

You will be given a list of candidates such as documents, images, videos, audios, etc. You need to check them carefully and understand all of them. Then you will be given a query, and your goal is to find all candidates from the list that can help answer the query. Print out the ID of each candidate.

Generic  
Instruction  
(valid for  
all datasets)

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

## Part 5. CiC Prompt Ablations

- **Removing Chain-of-Thought reasoning (Without CoT)**
  - It leads to worse performance
  - Each few-shot example has its own small corpus consisting of nine random passages and one gold passage

```
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.
```

General  
Instruction

```
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:
Final Answer: [54, 0]
...
```

Few-shot  
Exemples  
(no CoT)

```
===== 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<sub>29</sub>

## Part 5. CiC Prompt Ablations

- **Placing the query at the beginning of the prompt**
  - Query at Beginning led to a significant and consistent performance decrease
  - Prefix-caching actually works better than encoding the corpus conditioned on each query, which would be much more expensive

```
=====You need to answer the following query. =====  
Which document is most relevant to answer the query? Print out the TITLE and ID of the document. Then  
format the IDs into a list in the following format: [id1, id2, ...].  
If there is no perfect answer output the closest one. Do not give an empty final answer.  
query: How many records had the team sold before performing "aint thinkin bout you"?  
====Here are the context you need to read to answer the query. =====
```

Query  
at the  
beginning

```
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

```
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 answer the query! =====
```

Start Answer

## Part 5. CiC Prompt Ablations

- **Replacing monotonic numerical IDs with random**
  - Alphanumeric IDs negatively impacts performance in most datasets
  - Due to way in which numbers are tokenized, with fewer tokens for certain numbers.

```
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

```
ID: D5Y5 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese ... | END ID: D5Y5
...
ID: y2h8 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song ... | END ID: y2h8
ID: E8J2 | TITLE: Best Footballer in Asia 2016 | CONTENT: ... was awarded to Shinji Okazaki ... | END ID: E8J2
...
```

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: E8J2
TITLE: Shinji Okazaki | ID: D5Y5
Final Answer: [E8J2, D5Y5]
...
```

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<sub>1</sub>



## Part 5. CiC Prompt Ablations

- **Only placing the IDs at the front of the document**
  - Without ID Echo resulted in a 5% performance drop
  - Confirm that repeating text can compensate for missing context in autoregressive language models

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

ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese ...  
...  
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song ...  
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: ... was awarded to Shinji Okazaki ...  
...

Corpus  
Formatting  
(END ID  
removed)

==== 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  
Examples

==== 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



## Part 5. CiC Prompt Ablations

- **Remove the content & keep the title & ID in the corpus**
  - Test if the model is simply using parametric knowledge instead of grounding on the context
  - The model is able to perform well because it has already seen all of the datasets we are evaluating on during training

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

ID: 0 | TITLE: Shinji Okazaki | END ID: 0  
...  
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | END ID: 53  
ID: 54 | TITLE: Best Footballer in Asia 2016 | END ID: 54  
...

Corpus  
Formatting  
(CONTENT  
removed)

=====  
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

=====  
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

## Part 5. CiC Prompt Ablations

- **Remove the content & keep the title & ID in the corpus**
  - Test if the model is simply using parametric knowledge instead of grounding on the context
  - The model is able to perform well because it has already seen all of the datasets we are evaluating on during training

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

ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese ...  
...  
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song ...  
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: ... was awarded to Shinji Okazaki ...  
...

Corpus  
Formatting  
(END ID  
removed)

===== 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]**  
...

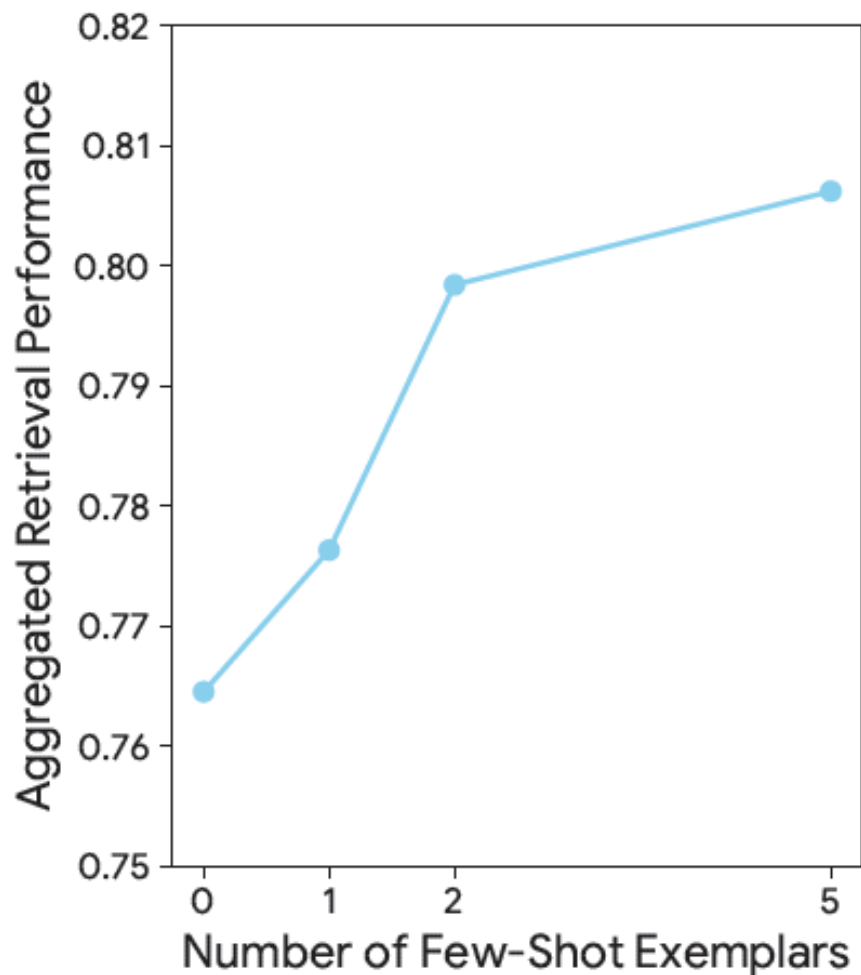
Few-shot  
Exemples

===== 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

## Part 5. CiC Prompt Ablations

- **Most common transfer learning techniques in NLP**



Effect of the number of few-shot examples

- Study how the number of few-shot examples in the prompt affects quality
- Increasing the number of examples improves the quality on the retrieval task, from 0.76 at zero-shot to 0.81 at 5-shots

# Related Work

---

- **Retrieval or Multi-hop QA**

- They do not fully capture the nuances of real-world retrieval or reasoning tasks
- These tasks lack the dynamic scaling capabilities of synthetic benchmarks, which makes them difficult to adapt to very long contexts

- **Context length**

- Longalpaca & LongBench-Chat evaluate instruction-following under long context settings but contain relatively low task diversity and no examples beyond 100k context length
- Similar to LOFT, Ada-LEval proposes a length-adaptable benchmark; however, their tasks are somewhat synthetic and may not resemble real-world applications

- **Long-context QA using the top retrieved documents**

- LCLMs lose recall when relevant information is placed in the middle of the context
- Extend this type of evaluation of LCLMs to context lengths of up to 1M tokens
- Offers an alternative approach where the retrieval corpus is directly provided as context, eliminating task specific training

- **Long Context Frontiers benchmark (LOFT)**
  - A suite of tasks that rigorously assesses LCLMs on tasks ripe for a paradigm shift
    - Retrieval, retrieval-augmented generation, SQL-like reasoning, and in-context learning
  - Provides dynamic scaling of context lengths of up to 1 million tokens
  - Despite having never been trained to do retrieval, LCLMs have retrieval capabilities rivaling task-specific hand-crafted SOTA retrieval systems
  - Nevertheless, there remains considerable room for advancement in long-context reasoning, particularly as models gain access to even longer context windows

# LOFT Dataset Creation

---

- **Limitations**

- The entire LOFT 128k test sets contain around 35 datasets  $\times$  100 prompts  $\times$  128k tokens = 448M input tokens, which cost \$1, 568 for Gemini 1.5 Pro, \$2, 240 for GPT-4o, and \$6, 720 for Claude 3 Opus at the time of writing.
- To reduce costs, we also release dev sets, which are 10x smaller and can be evaluated with around \$200 using Gemini 1.5 Pro or GPT-4o
- Could not measure the efficiency improvements from prefix caching [20] due to API constraints at the time of writing
- Without caching, the Gemini 1.5 Pro API has a median latency of roughly four seconds for 32k input tokens, twelve seconds for 128k input tokens, and 100 seconds for 1 million input tokens
- The speed is likely slower than specialized retrievers or SQL databases
- Our retrieval and RAG tasks was limited to 1 million tokens, which still leaves a large gap from real-world applications that may involve several million or even billions of documents

## Part 9. Detailed Statistics

Task	Dataset	# Queries (Few-shot / Development / Test)	Supported Context Length	# Candidates
<b>Text Retrieval</b>	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
<b>Visual Retrieval</b>	Flickr30k	5 / 10 / 100	32k / 128k	115 / 440
	MS COCO	5 / 10 / 100	32k / 128k / 1M	115 / 440 / 3,448
	OVEN	5 / 10 / 100	32k / 128k / 1M	110 / 448 / 3475
	MSR-VTT	5 / 10 / 100	32k / 128k / 1M	35 / 140 / 1,101

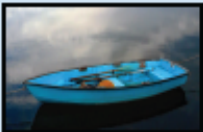

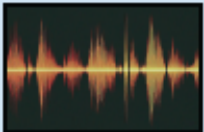
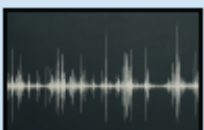




## Part 9. Detailed Statistics

<b>Task</b>	<b>Dataset</b>	<b># Queries</b> (Few-shot / Development / Test)	<b>Supported</b> <b>Context Length</b>	<b># Candidates</b>
<b>RAG</b>	NQ	5 / 10 / 100	32k / 128k / 1M	214 / 883 / 6,999
	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
<b>SQL</b>	Spider	1 / 10 / 100	32k / 128k / 1M	1 / 1 / 1
	SParC	1 / 10 / 100	32k / 128k / 1M	1 / 1 / 1
<b>Many-Shot</b> <b>ICL</b>	BBH-date	- / 10 / 90	32k	150
	BBH-salient	- / 10 / 90	32k	104
	BBH-tracking7	- / 10 / 90	32k	123
	BBH-web	- / 10 / 90	32k	150
	LIB-dialogue	- / 10 / 100	32k / 128k / 1M	61 / 274 / 1,059



# Examples of the task prompts in LOFT

	Text Retrieval	Visual Retrieval	Audio Retrieval	RAG	SQL	Many-Shot ICL
Input	<p>ID: 0   Title: Cheese   Text: Cheese is the...</p> <p>ID: 1   Title: 2016_Olympics   Text: Rio hosted the...</p> <p>...</p> <p>ID: 1000+   Title: Porsche   Text: Porsche is a...</p> <p><b>Find documents about the 2023 NBA Champion.</b></p>	<p>ID: 0 Image: </p> <p>...</p> <p>ID: 1000+ Image: </p> <p><b>Find an image of two people in a driving a blue convertible.</b></p>	<p>ID: 0 Audio: </p> <p>...</p> <p>ID: 1000+ Audio: </p> <p><b>Find audio saying "he documented himself in a 1998 book".</b></p>	<p>ID: 0   Title: Cheese   Text: Cheese is the...</p> <p>ID: 1   Title: 2016_Olympics   Text: Rio hosted the...</p> <p>...</p> <p>ID: 1000+   Title: Porsche   Text: Porsche is a...</p> <p><b>Did Einstein use an iPhone?</b></p>	<p>Table: SINGERS ID   Name   Age 0   John Smith   33 ...</p> <p>Table: CONCERTS ID   Singer_ID   Size 0   0   4322 ...</p> <p><b>Find all singers with concerts greater than the average size.</b></p>	<p>ID: 0   Reverse the word "glue". eulg</p> <p>ID: 1   Reverse the word "bench". hcneb</p> <p>...</p> <p>ID: 1000+   Reverse the word "spider". redips</p> <p><b>Reverse the word "papaya".</b></p>
Output	ID: 425   Title: 2023_NBA_Finals	ID: 12   Image: 	ID: 2344   Audio: 	No, he did not.	The average CONCERTS size is 1,421. Singers with >1,421 attendees are ...	ayapap

# Qualitative Analysis of Long-Context Wins and Losses

- **Use Gemini 1.5 Pro for the analysis**
  - For HotPotQA, we observed that some of the questions are actually answerable from a single document

Task = <b>Retrieval</b>	
Dataset = <b>Natural Questions</b> 1M	
<b>Input</b>	CiC-prompt(query="when does monday night raw come on hulu")
<b>Output</b>	The following documents can help answer the query: TITLE: WWE Raw   ID: 1023 Final Answer: ['1023'] (✓)
Dataset = <b>HotPotQA</b> 128k (multi-hop)	
<b>Input</b>	CiC-prompt(query="Which international football team has had a player endorse the Nike Hypervenom football boot?")
<b>Output</b>	The following documents can help answer the query: reasoning: The Nike Hypervenom is endorsed by players such as Marcus Rashford based on the document **ID:** 413 (title: Nike Hypervenom). reasoning: Marcus Rashford plays for the England national team based on the document **ID:** 558 (title: Marcus Rashford). Final Answer: ['413', '558'] (✓)

- Use Gemini 1.5 Pro for the analysis

Task = RAG	
Dataset = MuSiQue 128k (multi-hop)	
Input	CiC-prompt(query="Where do greyhound buses leave from in the city the performer of <u>Born A Lion</u> formed in?")
Output	reasoning: <u>Born A Lion</u> is the debut album by Canadian rock band <u>Danko Jones</u> . reasoning: <u>Danko Jones</u> is a Canadian rock trio from <u>Toronto, Ontario</u> . reasoning: The <u>Toronto Coach Terminal</u> is the central bus station for inter-city services in <u>Toronto, Ontario, Canada</u> . ID: 427   TITLE: Born a Lion ID: 383   TITLE: Danko Jones ID: 724   TITLE: Toronto Coach Terminal Final Answer: ['Toronto Coach Terminal'] (✓)
Dataset = QUEST 128k (multi-answer)	
Input	CiC-prompt(query="History books about <u>Malaysia</u> or the <u>Qing dynasty</u> or about <u>South Korea</u> ")
Output	reasoning: Chronicle of Malaysia gives an account of <u>Malaysia</u> from January 1957 to 31 August 2007. reasoning: Treason by the Book is a historical account of the Zeng Jing case which took place during the reign of the Yongzheng Emperor of <u>Qing China</u> around 1730. reasoning: Korea: A Walk Through the Land of Miracles is a book by Simon Winchester. He recounts his experience walking across <u>South Korea</u> , from Jeju in the south to the DMZ in the north. Final Answer: ['Chronicle of Malaysia', 'Korea: A Walk Through the Land of Miracles', 'Treason by the Book'] (✓)

# Qualitative Analysis of Long-Context Wins and Losses

- Use Gemini 1.5 Pro for the analysis

Task = SQL																							
Dataset = Spider 1M																							
Input	CiC-prompt(query="Find the year which offers the largest number of courses.")																						
Output	<p>Here's a step-by-step approach using the provided tables:</p> <p><b>**1. Identify relevant data:**</b> We need to look at the "year" column in the "section" table, which lists the year each course section was offered.</p> <p><b>**2. Count courses per year:**</b> Group the data by "year" and count the number of unique "course_id" values for each year. This will give us the number of distinct courses offered in each year.</p> <p><b>**3. Find the year with the maximum count:**</b> Identify the year with the highest count of distinct courses.</p> <p><b>**Based on the data provided, here's the breakdown of courses per year:**</b></p> <table><thead><tr><th>Year</th><th>Number of Courses</th></tr></thead><tbody><tr><td>2001</td><td>29</td></tr><tr><td>2002</td><td>36</td></tr><tr><td>2003</td><td>38</td></tr><tr><td>2004</td><td>31</td></tr><tr><td>2005</td><td>31</td></tr><tr><td>2006</td><td>41</td></tr><tr><td>2007</td><td>40</td></tr><tr><td>2008</td><td>37</td></tr><tr><td>2009</td><td>35</td></tr><tr><td>2010</td><td>39</td></tr></tbody></table> <p><b>**The year 2006 has the largest number of courses offered (41).**</b></p> <p>Final Answer: ['2006'] (✓)</p>	Year	Number of Courses	2001	29	2002	36	2003	38	2004	31	2005	31	2006	41	2007	40	2008	37	2009	35	2010	39
Year	Number of Courses																						
2001	29																						
2002	36																						
2003	38																						
2004	31																						
2005	31																						
2006	41																						
2007	40																						
2008	37																						
2009	35																						
2010	39																						

# Qualitative Analysis of Long-Context Wins and Losses

- Use Gemini 1.5 Pro for the analysis

Task = SQL																							
Dataset = Spider 1M																							
Input	CiC-prompt(query="Find the year which offers the largest number of courses.")																						
Output	<p>Here's a step-by-step approach using the provided tables:</p> <p><b>**1. Identify relevant data:**</b> We need to look at the "year" column in the "section" table, which lists the year each course section was offered.</p> <p><b>**2. Count courses per year:**</b> Group the data by "year" and count the number of unique "course_id" values for each year. This will give us the number of distinct courses offered in each year.</p> <p><b>**3. Find the year with the maximum count:**</b> Identify the year with the highest count of distinct courses.</p> <p><b>**Based on the data provided, here's the breakdown of courses per year:**</b></p> <table><thead><tr><th>Year</th><th>Number of Courses</th></tr></thead><tbody><tr><td>2001</td><td>29</td></tr><tr><td>2002</td><td>36</td></tr><tr><td>2003</td><td>38</td></tr><tr><td>2004</td><td>31</td></tr><tr><td>2005</td><td>31</td></tr><tr><td>2006</td><td>41</td></tr><tr><td>2007</td><td>40</td></tr><tr><td>2008</td><td>37</td></tr><tr><td>2009</td><td>35</td></tr><tr><td>2010</td><td>39</td></tr></tbody></table> <p><b>**The year 2006 has the largest number of courses offered (41).**</b></p> <p>Final Answer: ['2006'] (✓)</p>	Year	Number of Courses	2001	29	2002	36	2003	38	2004	31	2005	31	2006	41	2007	40	2008	37	2009	35	2010	39
Year	Number of Courses																						
2001	29																						
2002	36																						
2003	38																						
2004	31																						
2005	31																						
2006	41																						
2007	40																						
2008	37																						
2009	35																						
2010	39																						