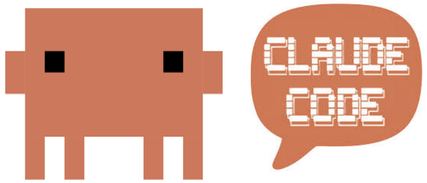
# Reasoning Over Long Context

University of Alberta


Princeton Language & Intelligence

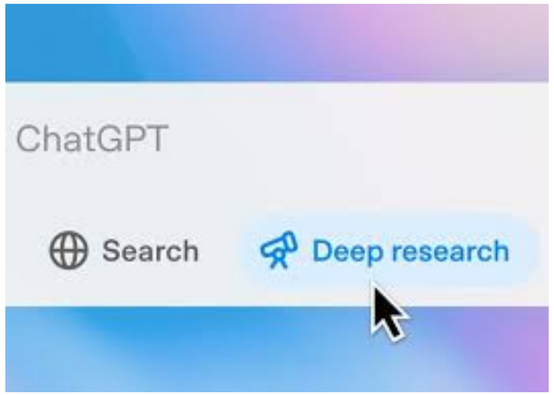
Xi Ye

## Coding





## Research

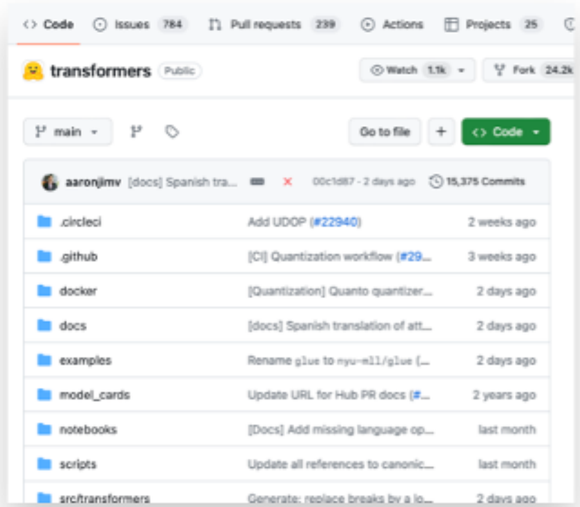



## Automation



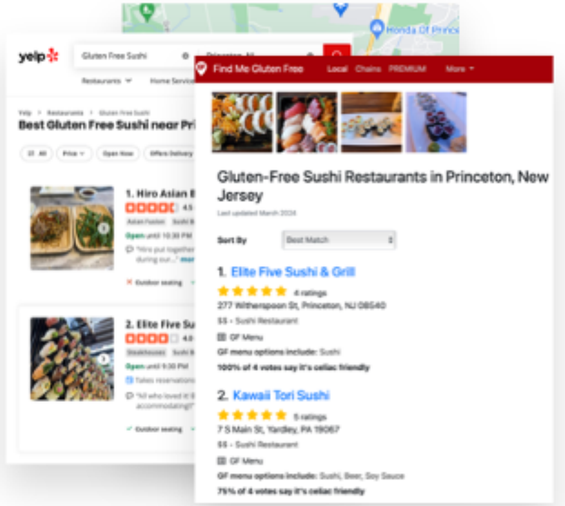
The Dune series

(~**2M** tokens)




The Transformers package

(~**10M** tokens)




100 web pages

(~**100K** tokens)

LLMs need to process really really … really long context!

# This lecture

Enabling LLMs to process long context

Long-context mid-training (in LLM tech reports)

A full-stack overview of long context training

Long context and long chain-of-thought reasoning

Scaffolding for long context

# Recap: model training pipeline

Pre-training

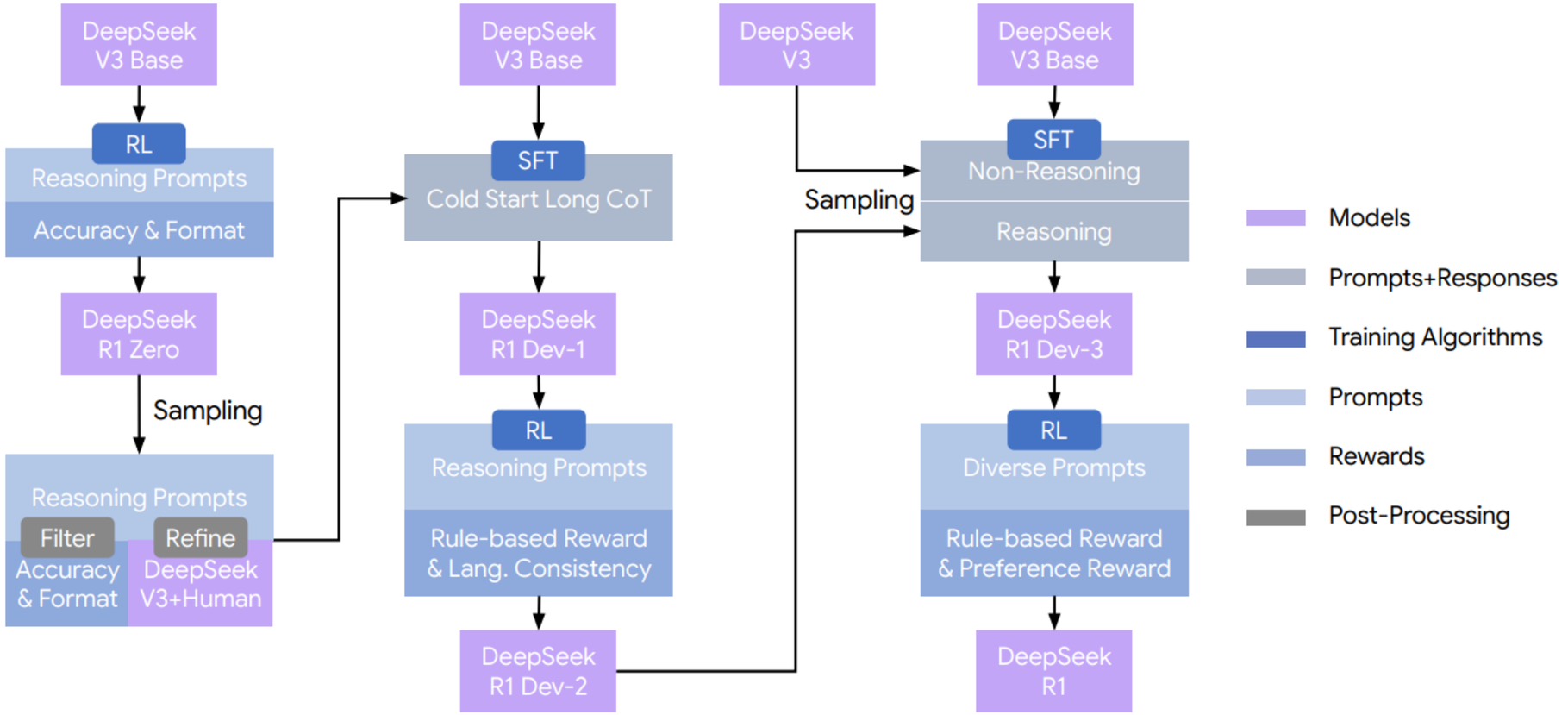
Post-Training

Scaling laws; data mixture; …

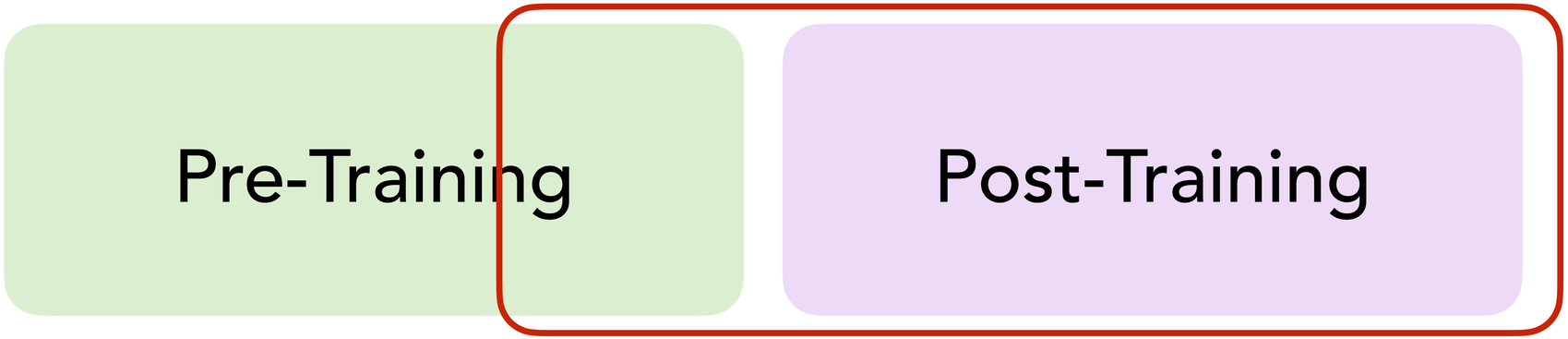
SFT; DPO; GRPO; …

What if we further unpack …

# How real training pipeline looks like



# When to inject long-context capabilities?




The end of pre-training and some parts of post-training

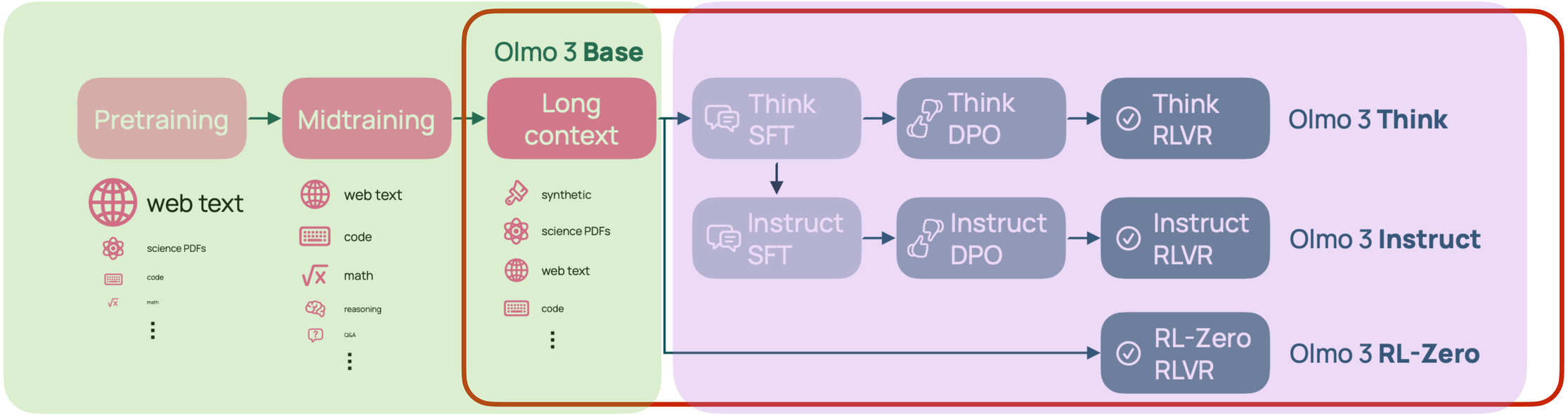
# When to inject long-context capabilities?

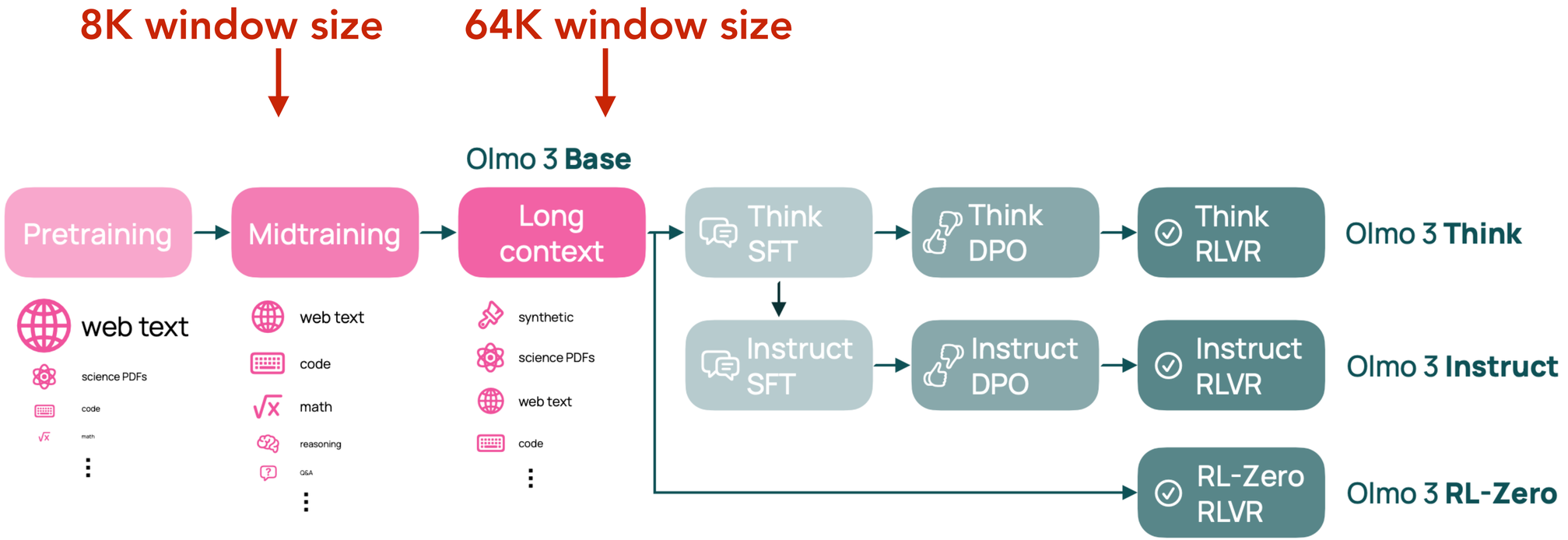
Pre-Training

Post-Training

The end of pre-training and some parts of post-training

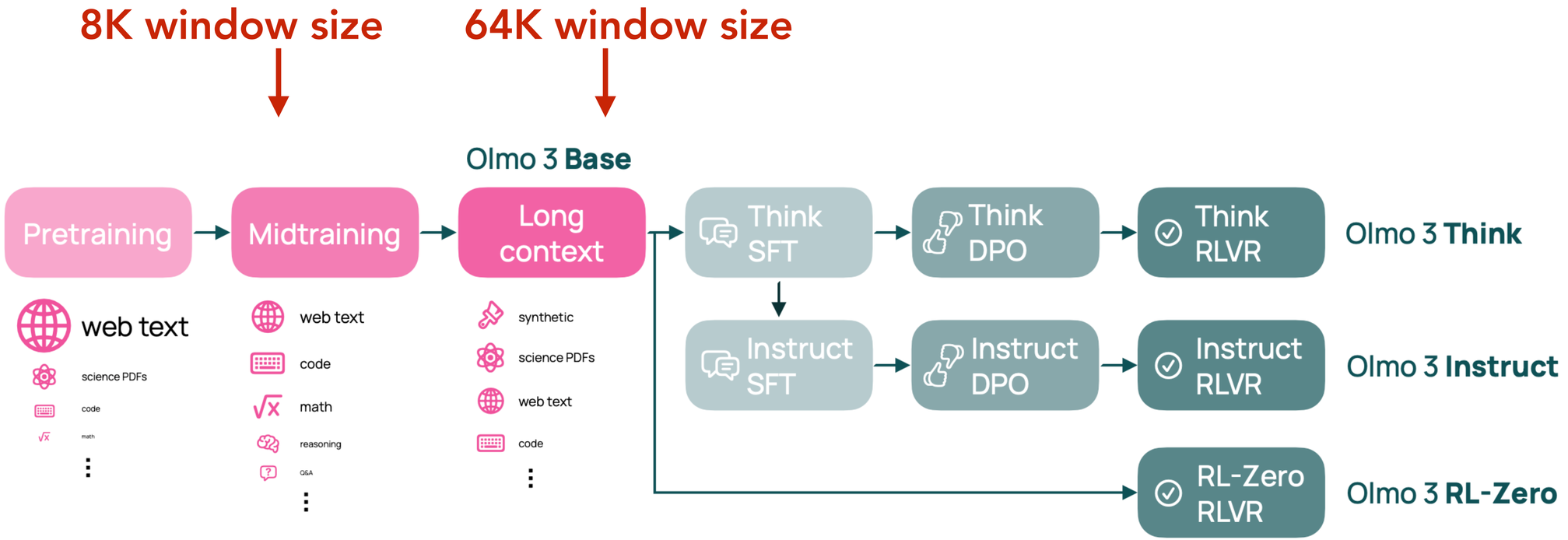
## OLmo3 Training Pipeline






Train on short context size first:

- majority of the pre-training data is short-context anyway
- Natural curriculum (longer context data is harder to learn
- Efficiency consideration




**Common Recipe:** OLmo3 as an example (but it is similar in Qwen3, DeepSeek-V3)

- Continual **pre-training** on long documents
- **Synthetic** long context data
- Further context extension (YaRN)

# Continual Pre-training

First adjust hyper-parameters of positional embeddings (to support longer context)

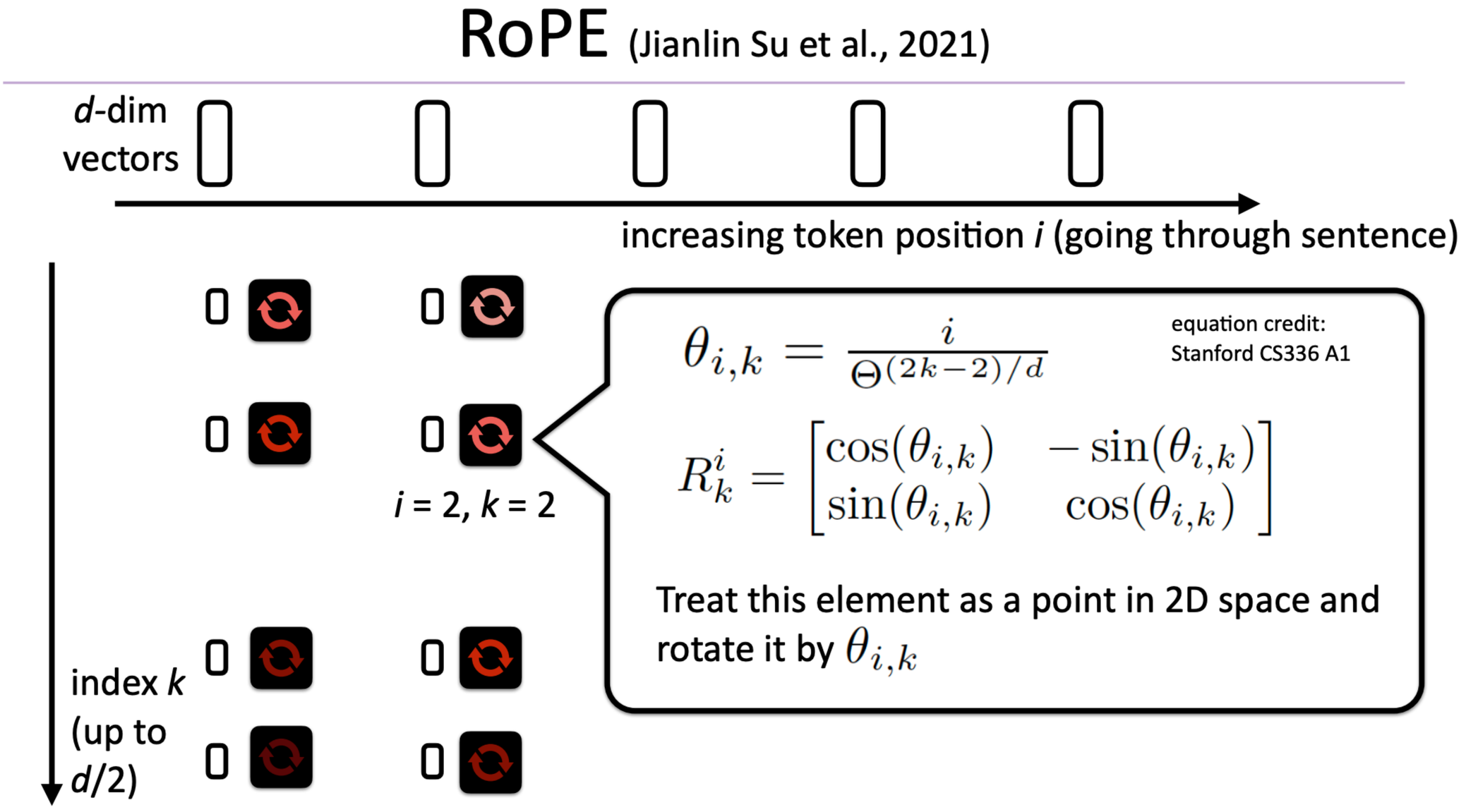
## OLmo3

**RoPE extension** OLMO 3 uses RoPE (Su et al., 2024) to encode positional information within the transformer architecture. We experiment with several methods for extending RoPE beyond the original pre-training context length, including adjusted base frequency scaling (Xiong et al., 2023; Rozière et al., 2024), position interpolation (Chen et al., 2023), and YaRN (Peng et al., 2023). Each approach is applied either to all RoPE instances or is restricted to RoPE used in full attention layers. We find that applying YaRN only to full attention layers yields the best overall performance.

## Qwen3

Following Qwen2.5 (Yang et al., 2024b), we increase the base frequency of RoPE from 10,000 to 1,000,000 using the ABF technique (Xiong et al., 2023). Meanwhile, we introduce YARN (Peng et al., 2023) and Dual Chunk Attention (DCA, An et al., 2024) to achieve a four-fold increase in sequence length capacity during inference.

## RoPE (previous slides)




69

RoPE (previous slides)

Max i increases as we extend L to 4L or 8L

## RoPE (Jianlin Su et al., 2021)

$$\theta_{i,k} = \frac{i}{\Theta^{(2k-2)/d}}$$

What happens as *i* increases?

What happens as *k* increases?

Need to increase theta

# Continual Pre-training

First adjust hyper-parameters of positional embeddings (to support longer context)

Continual pre-train (language modeling loss) on longer documents (e.g., arXiv papers, code repository

**Typical source of long documents**

(ProLong, Gao et al., 25)

| Data | #Long tokens |
|---|---|
| **Code Repos** | **98.8B** |
| **SP/Books** | **33.2B** |
| SP/CC | 15.3B |
| SP/Arxiv | 5.2B |
| SP/GitHub | 2.8B |
| SP/Wiki | 0.1B |
| SP/StackEx | <0.1B |
| SP/C4 | <0.1B |

# Continual Pre-training

Typically train on **both short-context** data and **long-context data** to preserve short-context performance

**Long Context Extension data used for OLMo3**

| Source | Length bucket | 600B Pool | | 50B Mix | |
|---|---|---|---|---|---|
| | | Tokens | Docs | Tokens | Docs |
| Synthetic—CWE | 32k-64k | 8.77B (1.37%) | 189K | 1.94B (3.88%) | 71.3K |
| Synthetic—REX | 32k-64k | 24.1B (3.77%) | 492K | 6.08B (12.2%) | 217K |
| olmOCR PDFs | 8k-16k | 144B (22.5%) | 12.7M | 2.27B (4.55%) | 235K |
| olmOCR PDFs | 16k-32k | 115B (18.0%) | 5.06M | 1.85B (3.70%) | 110K |
| olmOCR PDFs | 32k-64k | 106B (16.6%) | 2.30M | 4.81B (9.63%) | 177K |
| olmOCR PDFs | 64k-128k | 96.0B (15.0%) | 1.05M | – | – |
| olmOCR PDFs | 128k-256k | 60.8B (9.5%) | 342K | – | – |
| olmOCR PDFs | 256k-512k | 35.1B (5.49%) | 97.1K | – | – |
| olmOCR PDFs | 512k-1M | 21.5B (3.36%) | 30.2K | – | – |
| olmOCR PDFs | 1M+ | 26.9B (4.21%) | 12.2K | – | – |
| Midtraining data mix | Variable | – | – | 33.0B (66.1%) | 79.2M |
| **Total** | | **639B** | **22.3M** | **50.0B (100%)** | **80.0M** |

50B tokens in total
66% short context

# Continual Pre-training

Typically train on **both short-context** data and **long-context data** to preserve short-context performance

**Interleaving long and short context data.** Rather than training on just long-context data, we mix high quality, short-context data from midtraining (stage two) to ensure that performance on short context tasks is not meaningfully degraded. Early experiment on 10 billion token extension show that a 66% / 34% mix of long-context to short-context data drops performance on a subset of our evaluation suite by 2.5 points; in comparison, a 34% long-context, 66% short-context only drops by 0.8 points.

34% short + 66% long -> 2.5% degradation on short-context tasks

66% short + 34% long -> 0.8% degradation on short-context tasks

# Synthetic Long Context Data

Mostly in an **instruction** style (as opposed to narrative style in natural text)

OLMo3 uses synthetic data for continual pre-training; they can also be used for post-training

**Common Synthetic Long-Context Tasks**

- Fill-in-the-Middle (commonly used)
- Question answering
- Aggregation (e.g., summarization
- (More, like paragraph reordering ….

**Fill-in-the-Middle Task**

Context …
<missing part>
Context …

Q: Fill in the missing part

# Synthetic Long Context Data

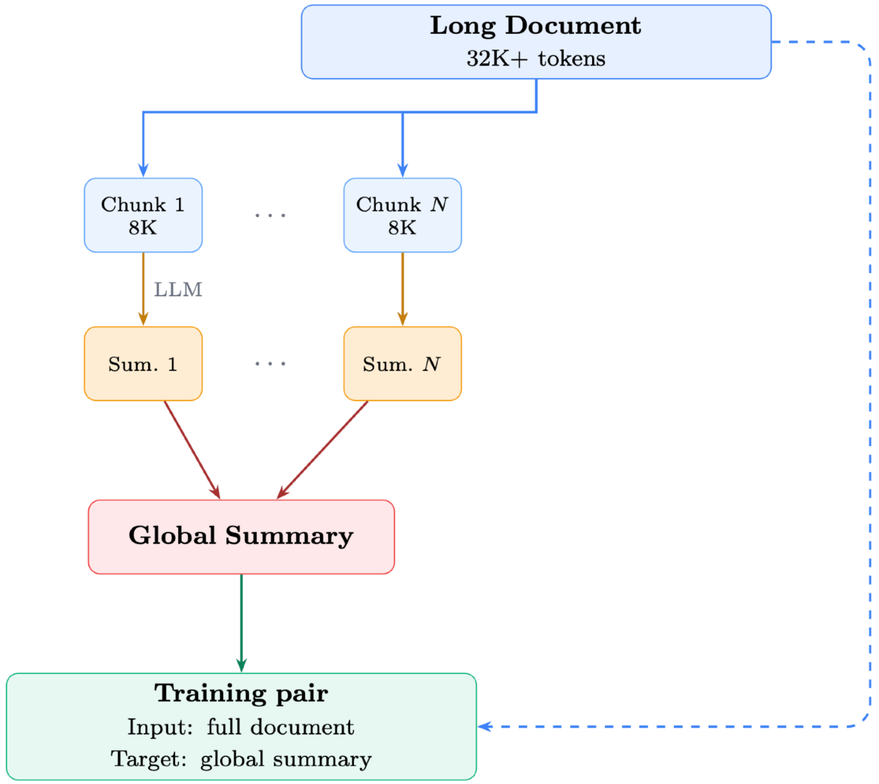
Mostly in an **instruction** style (as opposed to narrative style in natural text)

OLMo3 uses synthetic data for continual pre-training; they can also be used for post-training

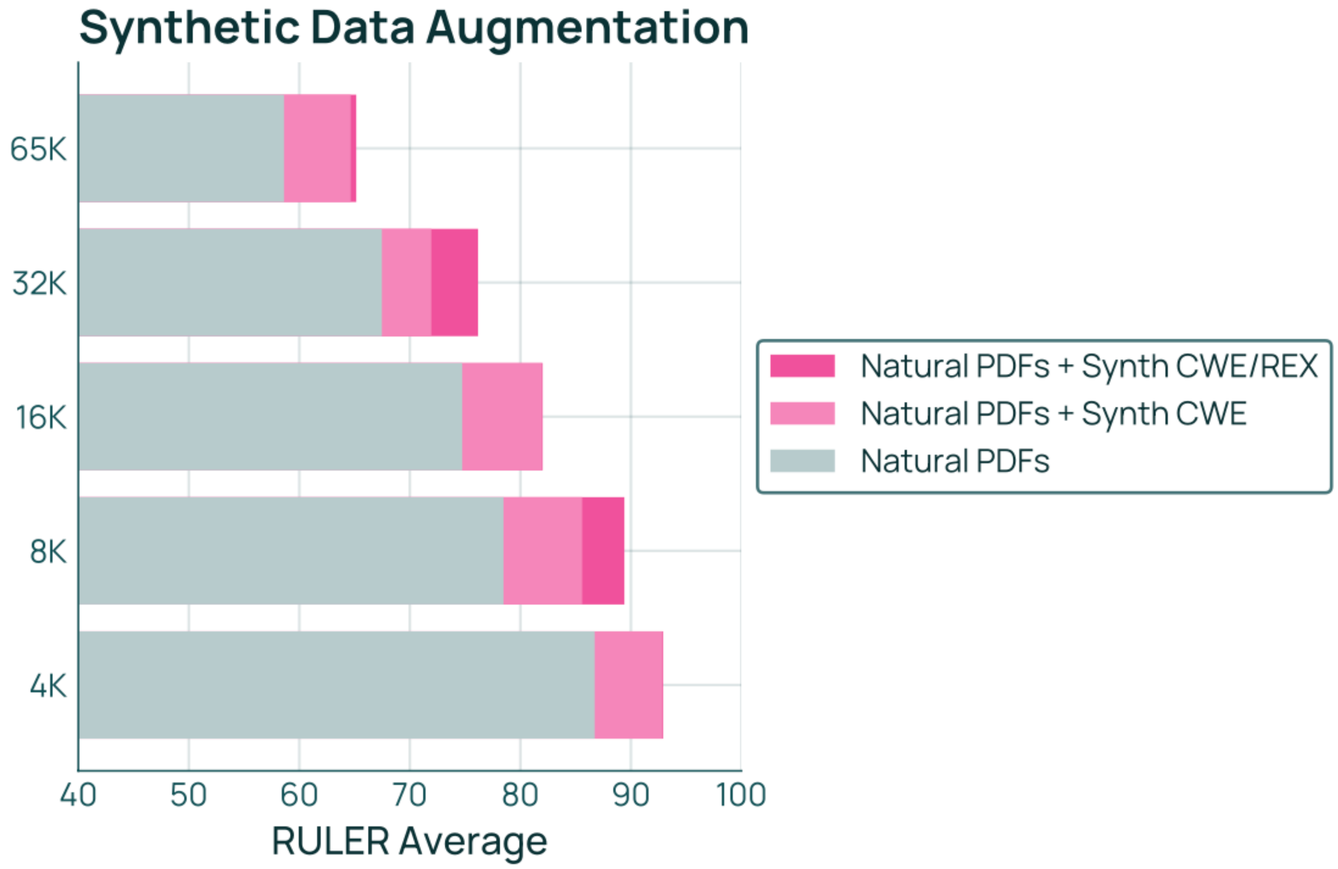
**Common Synthetic Long-Context Tasks**

- Fill-in-the-Middle (commonly used)
- Question answering
- Aggregation (e.g., summarization
- (More, like paragraph reordering ….

**Synthetic Summarization Data**

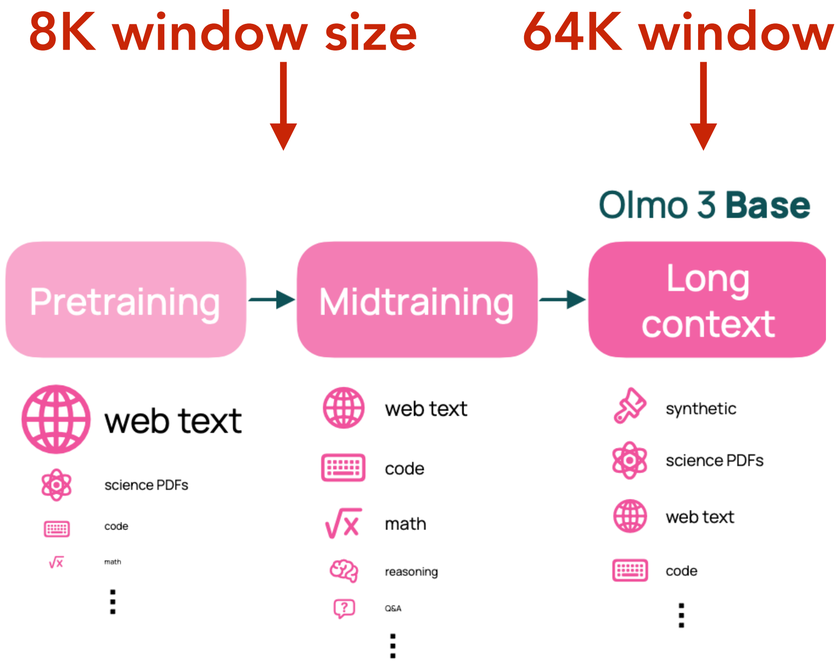


# Synthetic Long Context Data




OLmo3 shows substantial improvements from **synthetic data**

# Further Context Length Extension (After Training)




What if I want to use this LM on 256K tasks?

Use **YaRN** (Bowen Peng et al., 2023) at inference time

This is how Qwen supports 128K or 1M context size

YaRN Slides (previous lecture)

# YaRN (Bowen Peng et al., 2023)

Position interpolation: if RoPE is trained with encodings up to token *L*, you can expand it to *L'* by

$$\theta_{i,k} = \frac{i\,L/L'}{\Theta^{(2k-2)/d}}$$

Notion of wavelength at position *k*: the number of tokens needed to do a full rotation

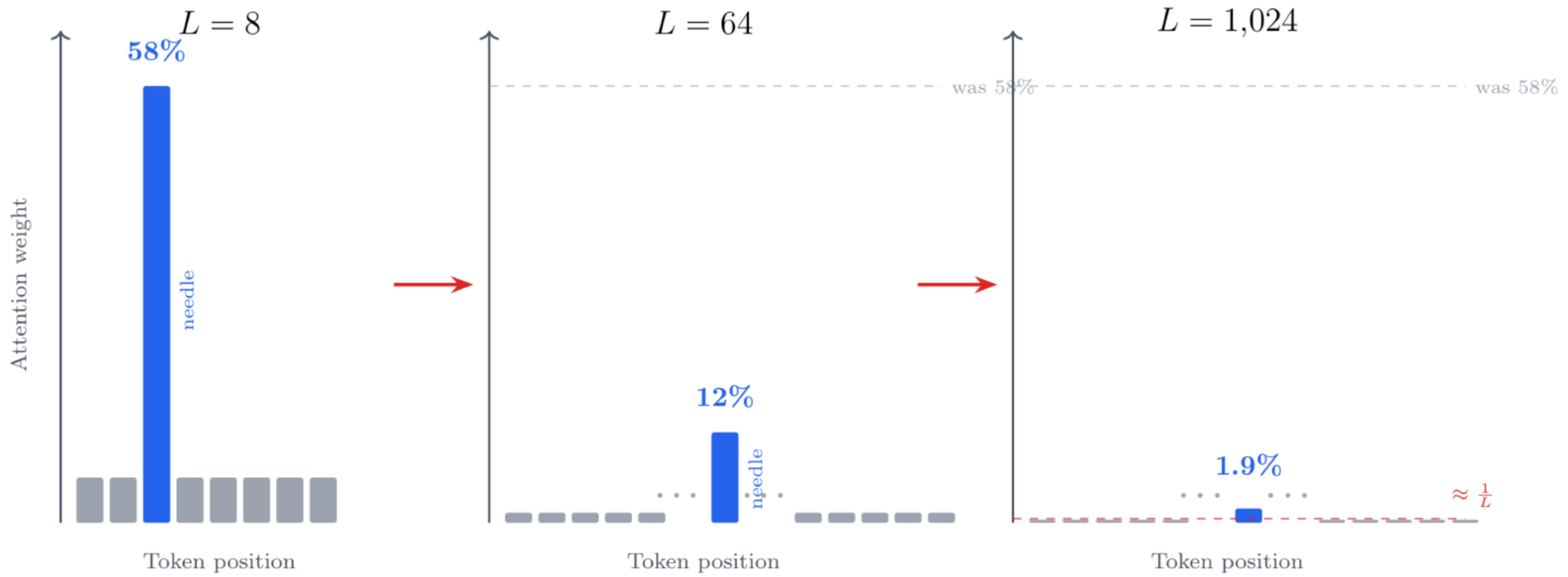
$$\lambda_k = \frac{2\pi}{\theta_k} = 2\pi\,\Theta^{\frac{2k}{|D|}}$$

Two ideas in YaRN:

1. If wavelength is small (*r* is large), we do not use position interp. If wavelength is large, then we use position interp.
2. Introducing a temperature *t* on the attention computation to further rescale it

# Attention Scaling in YaRN

**Problem:** Longer sequences ⇒ more keys ⇒ softmax entropy increases ⇒ attention becomes diffuse.




$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{L} e^{z_j}}$$

Even if the **logit gap** stays constant, the denominator grows with $L$,

This is **inevitable** as softmax never produces 0 or negative values

**Fix:** Scale attention logits by temperature $\sqrt{1/t}$:

$$\text{Attention} = \text{softmax}\left(\frac{QK^\top}{\sqrt{d} \cdot \sqrt{t}}\right) V, \qquad t = 0.1 \ln(s) + 1$$

# Summary: Long-Context Extension Training

**Common Recipe:** OLmo3 as an example (but it is similar in Qwen3, DeepSeek-V3)

- Continual **pre-training** on long documents
- **Synthetic** long context data
- Further context extension (YaRN)

For context extension, the recipe is still dominated by data decisions

# This lecture

Enabling LLMs to process long context

Long-context mid-training (in LLM tech reports)

A full-stack overview of long context training

Long context and long chain-of-thought reasoning

Scaffolding for long context

# Long-Context is Expensive: Memory Costs

**KV cache at inference** (per sequence, bf16):

$$\underbrace{2}_{\text{K,V}} \times n_{\text{layers}} \times n_{\text{kv}} \times d_{\text{head}} \times \underbrace{L}_{\text{seq}} \times 2\text{B}$$

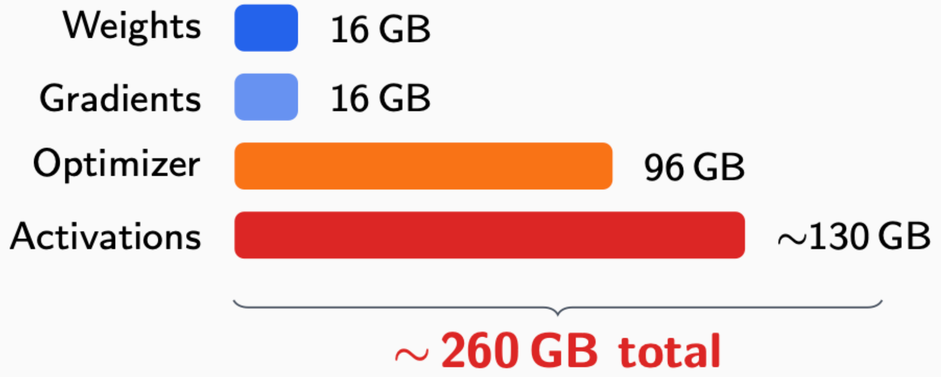
**Ex:** 8B model, GQA (8 KV heads), 32 layers, $d_{\text{head}}=128$:

| Context | KV Cache | vs. Weights |
|---|---|---|
| 4K | 0.5 GB | 3% |
| 32K | 4 GB | 25% |
| 128K | **16 GB** | 100% |
| 512K | **64 GB** | 400% |

Weights $\approx$ 16 GB in bf16.

At 128K the KV cache **equals** the model.

**Training memory** (8B, 128K, bf16 + Adam):



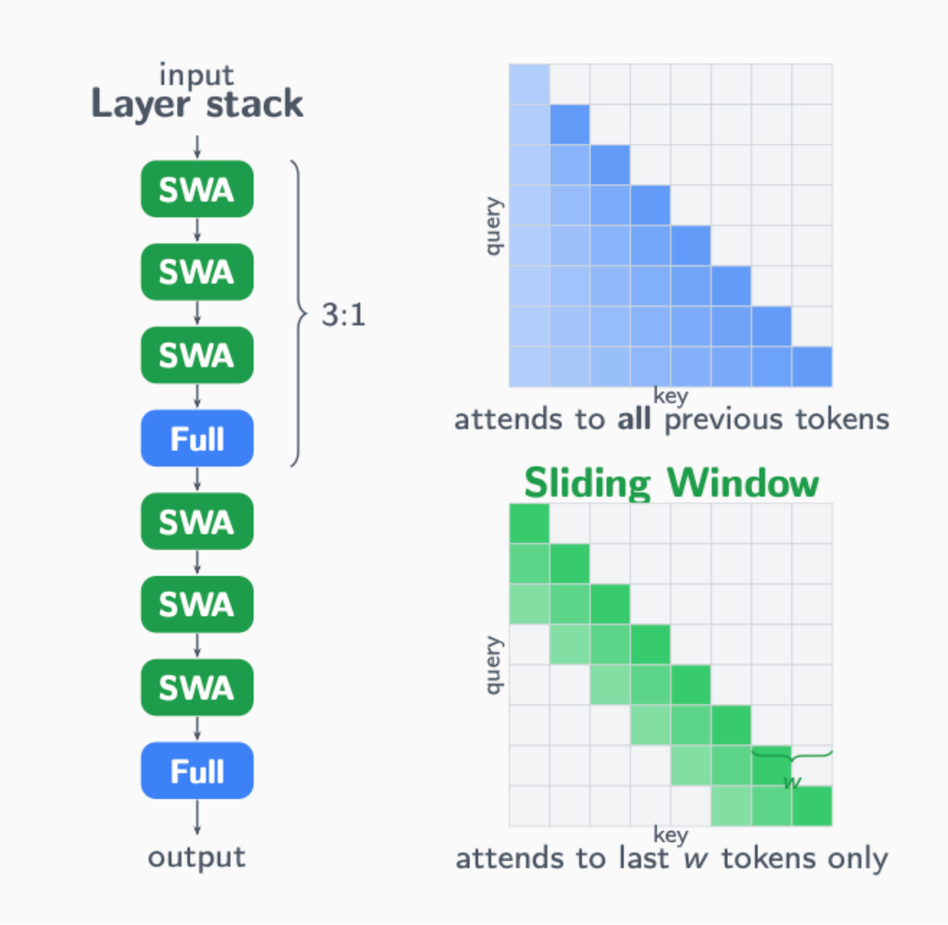

*Activations with FlashAttention + checkpointing.

Without FlashAttn: attention alone is $O(L^2)$ per layer.

Activations (basically gradients on KV caches) takes a huge chunk of memory

# Efficient Architecture

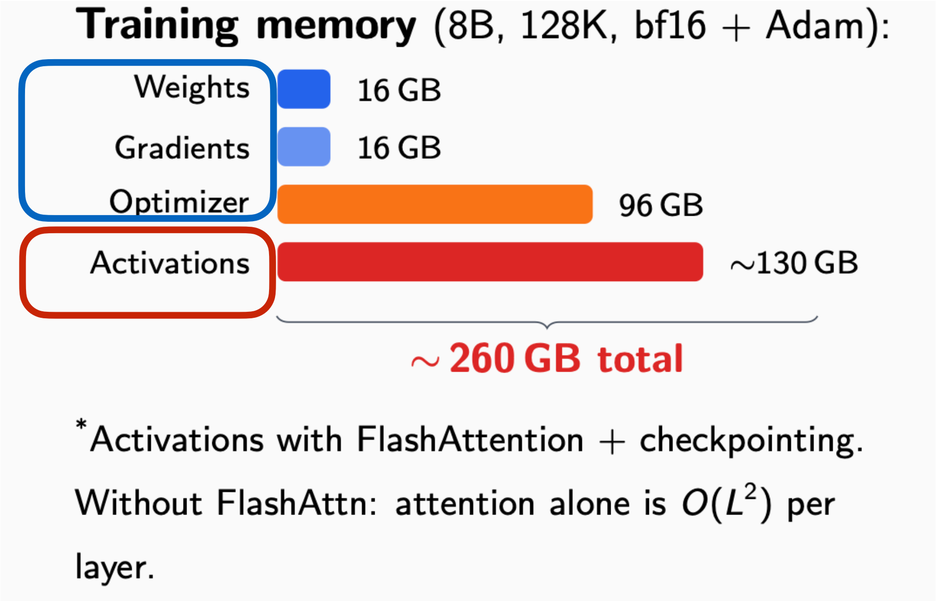
Interleaving **sliding window attention** and full attention layer (e.g. OLMo3, Gemma 4, GPT-OSS)
- Efficient for training and inference




Interleaving linear attention and full attention layer (e.g. Qwen3.5, Kimi-Linear, MinMax-01)

# Memory Efficient Training: Sequence Parallelism

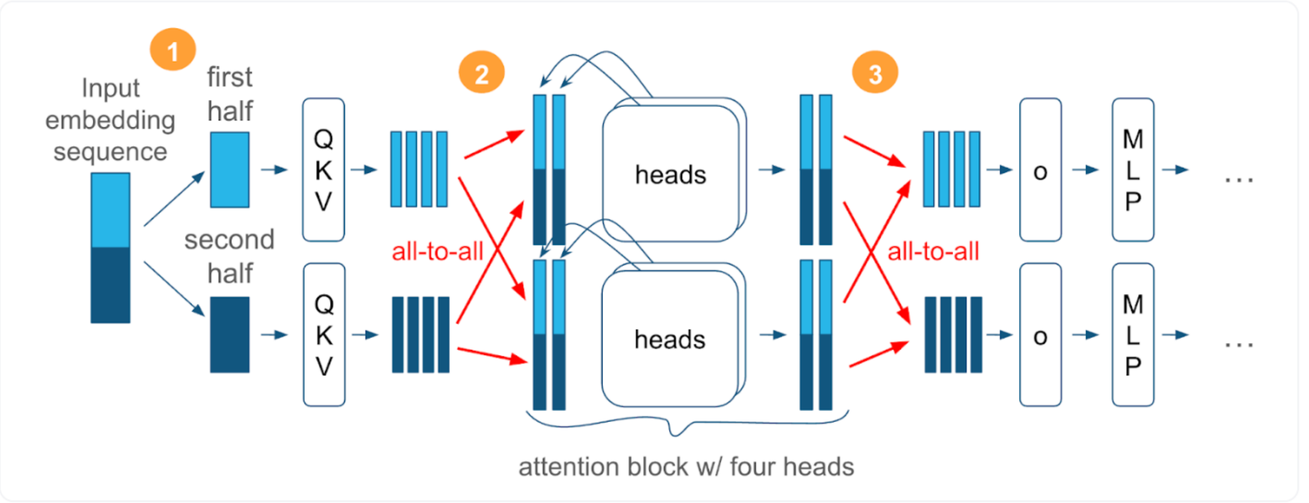
Model and optimizer shardded with FSDP




Activations are shardded with **sequence parallelism**

DeepSpeed Ulysses Sequence Parallelism (Sam Ade Jacobs et al, 2023)

**Core idea:** each GPU holds L/N tokens. Queries are local, but keys & values are streamed across GPUs so every query can attend to the full sequence.




Ulysses splits input sequences along the sequence dimension and uses all-to-all communication to exchange key-value pairs, enabling each GPU to compute a subset of attention heads. (***Source:*** *Snowflake Engineering Blog*)

# Long Context Post Training (RLVR)

We been mostly talking about mid-training but not post-training 🤔 ❓

- Agentic RL post-training naturally trains LMs to process long-context better

- Few (academic) papers study long-context post-training, as it is very compute-intensive

Please read the following text.
Document 0:
…
Document 3:
Who's Who? is a studio album by American jazz musician John Scofield. It features two different bands, one acoustic and one electric. The acoustic group, featuring Scofield's then-employer Dave Liebman on saxophones, Eddie G\u00f3mez on bass, and Billy Hart on drums, recorded “The Beatles” and “How the West Was Won”. …
{“bdd640fb-0667-4ad1-9c80-317fa3b1799d”: “23b8c1e9-3924-46de-beb1-3b9046685257”}.
…
Document 10:
…
The university is one of the smallest of the 23 CSU campuses in California. Sonoma State offers 92 Bachelor's degrees, 19 Master's degrees, one Doctoral degree (Doctor of Education), and 11 teaching credentials. {“972a8469-1641-4f82-8b9d-2434e465e150”: “Musician and satirist Allie Goertz wrote a song about the “The Simpsons” character Milhouse, who Matt Groening named after who?”}.
…
Document 47:
Neil Affleck
{“bd9c66b3-ad3c-4d6d-9a3d-1fa7bc8960a9”: “972a8469-1641-4f82-8b9d-2434e465e150”}.
Neil Affleck (born 1953) is a Canadian animator, director, and former actor. He has worked as an animator on “The Simpsons” and “Family Guy”, and as an actor appeared in a leading role in the 1981 film “My Bloody Valentine”. {“9a1de644-815e-46d1-bb8f-aa1837f8a88b”: “b74d0fb1-32e7-4629-8fad-c1a606cb0fb3”}.
…
In the context above, there is one correct question to answer. The correct question can only be found by following the correct consecutive chain of key:value pairs encoded with UUID strings (e.g., f81d4fae-7dec-11d0-a765-00a0c91e6bf6), starting from "bdd640fb-0667-4ad1-9c80-317fa3b1799d".
Find the correct question first, then answer it.

Figure 6: A skeleton of KeyChain-augmented training data used in LoongRL.

Still most papers on long-context post-training are about synthetic data

# This lecture

Enabling LLMs to process long context

Long-context mid-training (in LLM tech reports)

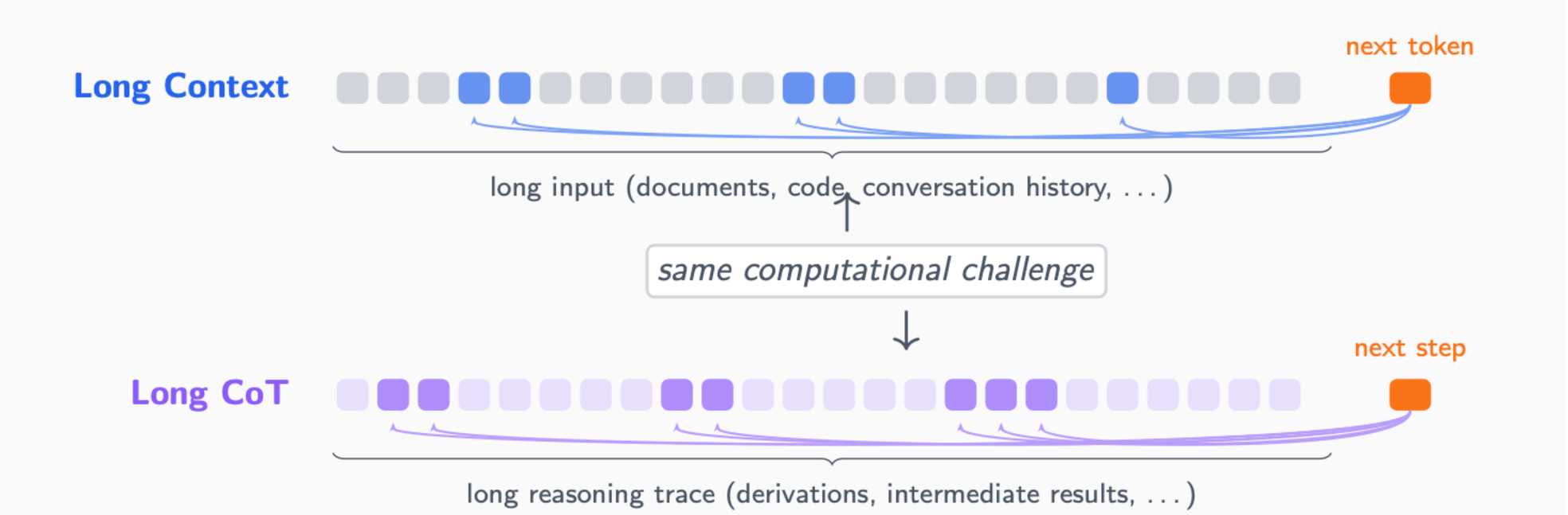
A full-stack overview of long context training

Long context and long chain-of-thought reasoning

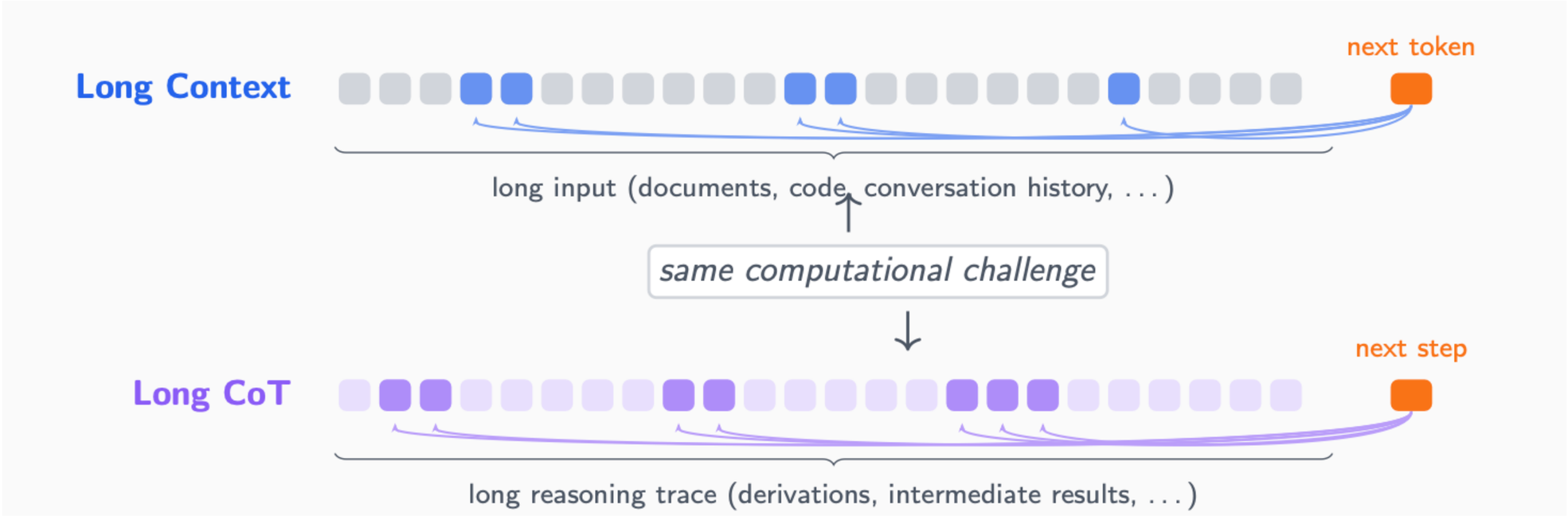
Scaffolding for long context

# Connection between Long-Context and Long CoT

- Naturally, LLMs need long context to support long CoT
- Shared challenge: each generation step must selectively attend to critical positions in a long preceding sequence (with semantical-wise and embedding-wise similar spans)



# Evaluating Long-Context by Long-CoT Generation




We evaluate long-context LMs by requiring models to generate long structured CoT (e.g. search trees)

**LongProc: Benchmarking Long-Context Language Models on Long Procedural Generation**

Xi Ye[♠] Fangcong Yin[◇*] Yinghui He[♠*] Joie Zhang[♠*] Howard Yen[♠*]
Tianyu Gao[♠] Greg Durrett[◇] Danqi Chen[♠]

# Evaluating Long-Context by Long-CoT Generation

## Input to LMs:

**Example Search Procedure for Countdown.**

[INSTRUCTION]
We will follow this search process:
- At each state, first choose two numbers from the number set.
- Next, try the four operations (+, −, ×, and /) to obtain the new number and add the new number to the number set.
- Continue this process until we reach the target number.

[EXAMPLE PROBLEM]
Numbers: [40, 19, 23, 7]
Target: 29

[EXAMPLE PROCEDURE]
Current number set: [40, 19, 23, 7]
|- Pick two numbers (40, 19) (numbers left: [23, 7])
|- Try 40+19=59. Current number set: [59, 23, 7]
|- Pick two numbers (59, 23) (numbers left: [7])
|- Try 59+23=82. Current number set: [82, 7]
|- Try 82+7=89. Evaluate 89!=29. Drop this branch.
|- Try 82-7=75. Evaluate 75!=29. Drop this branch.
|- Try 82*7=574. Evaluate 574!=29. Drop this branch.
|- Try 82/7=11.7. Evaluate 11.7!=29. Drop this branch
|- Try 59-23=36. Current number set: [36, 7].
|- Try 36+7=43. Evaluate 43!=29. Drop this branch.
|- Try 36-7=29. Evaluate 29==29. Target found!

[SOLUTION]
40+19=59, 59-23=36, 36-7=29

Instructions that describes a procedure

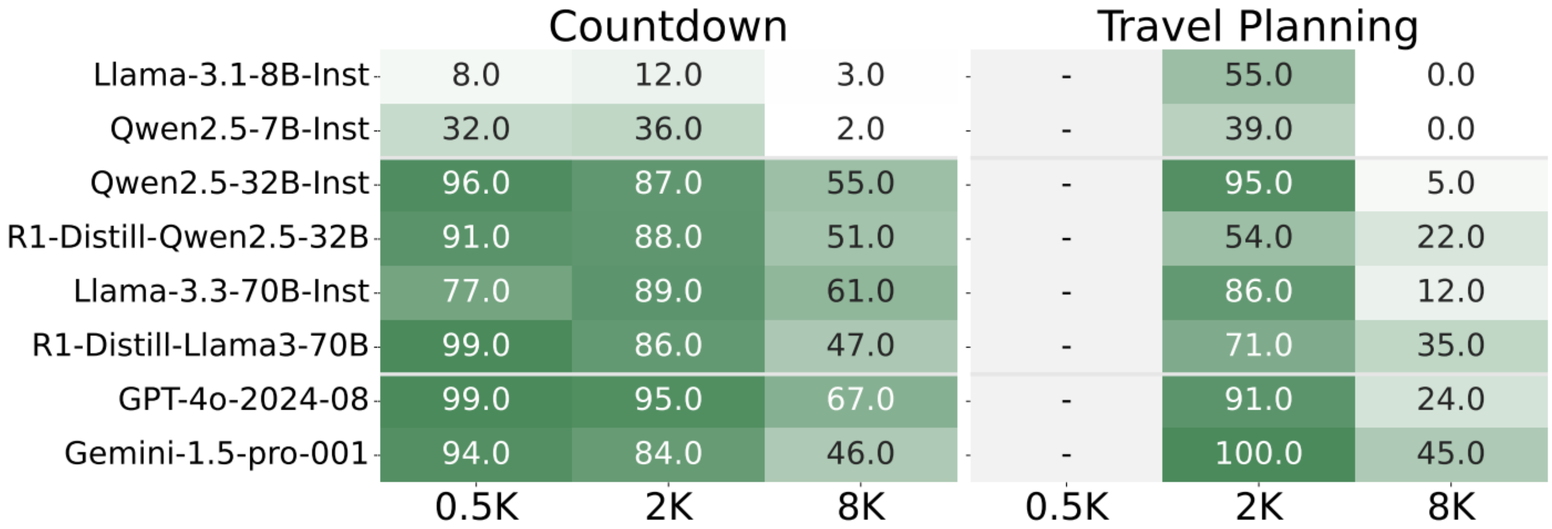
In-context examples

## Example Output

CoT for the given question

**Solving Procedure**
Initial number set: [44, 48, 35, 6], target: 25. Options for choosing two numbers: [(44, 48), (44, 35), (44, 6), (48, 35), (48, 6), (35, 6)].
|- Pick two numbers (44, 48) (numbers left: [35, 6]). Try possible operations.
|- Try 48 + 44 = 92. Add 92 to the number set. Current number set: [92, 35, 6], target: 25. Options for choosing two numbers: [(92, 35), (92, 6), (35, 6)].
|- Pick two numbers (92, 35) (numbers left: [6]). Try possible operations.
|- Try 92 + 35 = 127. Add 127 to the number set. Current number set: [127, 6], target: 25, just two numbers left.
|- Try 127 + 6 = 133. Evaluate 133 != 25, drop this branch.
|- Try 127 - 6 = 121. Evaluate 121 != 25, drop this branch.
|- Try 127 * 6 = 762. Evaluate 762 != 25, drop this branch.
|- Try 127 / 6 = 21.2. 21.2 is a decimal, drop this branch.
|- Try 92 - 35 = 57. Add 57 to the number set. Current number set: [57, 6], target: 25, just two numbers left.
|- Try 57 + 6 = 63. Evaluate 63 != 25, drop this branch.
|- Try 57 - 6 = 51. Evaluate 51 != 25, drop this branch.
|- Try 57 * 6 = 342. Evaluate 342 != 25, drop this branch.
|- Try 57 / 6 = 9.5. 9.5 is a decimal, drop this branch.
|- Try 92 * 35 = 3220. 3220 exceeds the maximum intermediate result, drop this branch.
|- Try 92 / 35 = 2.6. 2.6 is a decimal, drop this branch.
|- Pick two numbers (92, 6) (numbers left: [35]). Try possible operations.
|- Try 92 + 6 = 98. Add 98 to the number set. Current number set: [98, 35], target: 25, just two numbers left.
|- Try 98 + 35 = 133. Evaluate 133 != 25, drop this branch.
|- Try 98 - 35 = 63. Evaluate 63 != 25, drop this branch.
|- Try 98 * 35 = 3430. 3430 exceeds the maximum intermediate result, drop this branch.
|- Try 98 / 35 = 2.8. 2.8 is a decimal, drop this branch.
|- Try 92 - 6 = 86. Add 86 to the number set. Current number set: [86, 35], target: 25, just two numbers left.
|- Try 86 + 35 = 121. Evaluate 121 != 25, drop this branch.
|- Try 86 - 35 = 51. Evaluate 51 != 25, drop this branch.
|- Try 86 * 35 = 3010. 3010 exceeds the maximum intermediate result, drop this branch.
|- Try 86 / 35 = 2.5. 2.5 is a decimal, drop this branch.
|- Try 92 * 6 = 552. Add 552 to the number set. Current number set: [552, 35], target: 25, just two numbers left.
|- Try 552 + 35 = 587. Evaluate 587 != 25, drop this branch.

# Evaluating Long-Context by Long-CoT Generation




SoTA models (by the time we ran our evaluation) fails at long CoT generation even when the overall task length is not that long; (# input tokens: 8K, # output tokens: 0.5 - 8K)

Long CoT generation provides reliable signals for evaluating long context models

# Reasoning Models vs Non-Reasoning Models

| | Countdown | | | Travel Planning | | |
|---|---|---|---|---|---|---|
| | 0.5K | 2K | 8K | 0.5K | 2K | 8K |
| Llama-3.1-8B-Inst | 8.0 | 12.0 | 3.0 | - | 55.0 | 0.0 |
| R1-Distill-Llama3-8B | 83.0 | 69.0 | 26.0 | - | 44.0 | 3.0 |
| Qwen2.5-32B-Inst | 96.0 | 87.0 | 55.0 | - | 95.0 | 5.0 |
| R1-Distill-Qwen2.5-32B | 91.0 | 88.0 | 51.0 | - | 54.0 | 22.0 |
| Llama-3.3-70B-Inst | 77.0 | 89.0 | 61.0 | - | 86.0 | 12.0 |
| R1-Distill-Llama3-70B | 99.0 | 86.0 | 47.0 | - | 71.0 | 35.0 |

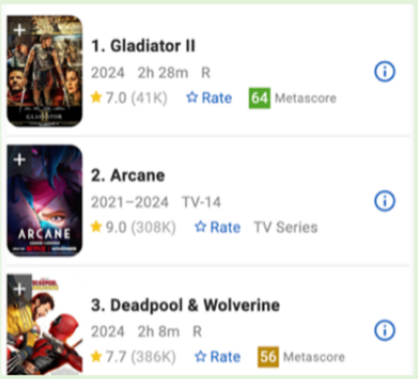
| | HTML to TSV | | |
|---|---|---|---|
| | 0.5K | 2K | 8K |
| Llama-3.1-8B-Inst | 43.4 | 29.0 | 23.4 |
| R1-Distill-Llama3-8B | 30.0 | 19.4 | 8.4 |
| Qwen2.5-32B-Inst | 74.5 | 46.9 | 25.4 |
| R1-Distill-Qwen2.5-32B | 78.2 | 53.9 | 38.8 |
| Llama-3.3-70B-Inst | 78.0 | 60.2 | 51.7 |
| R1-Distill-Llama3-70B | 74.8 | 60.2 | 46.4 |

Reasoning models outperform non-reasoning models (even on HTML to TSV, a non reasoning task)

**HTML to TSV**

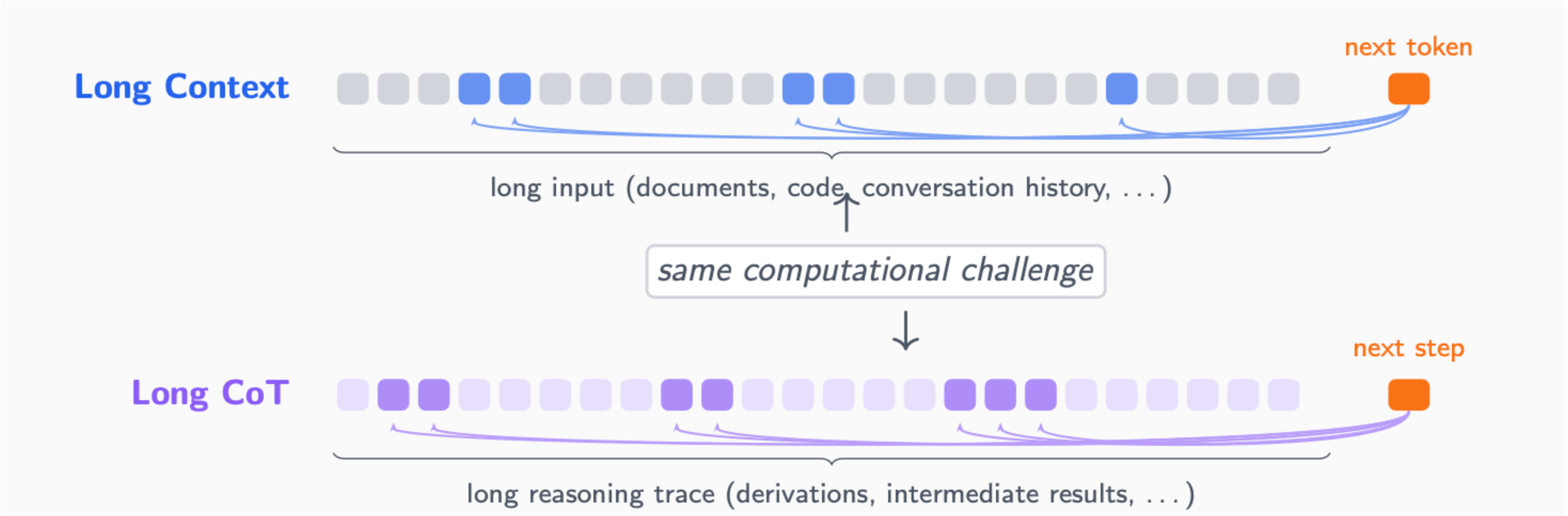
Extract specified information from HTML pages and structure it into a table format

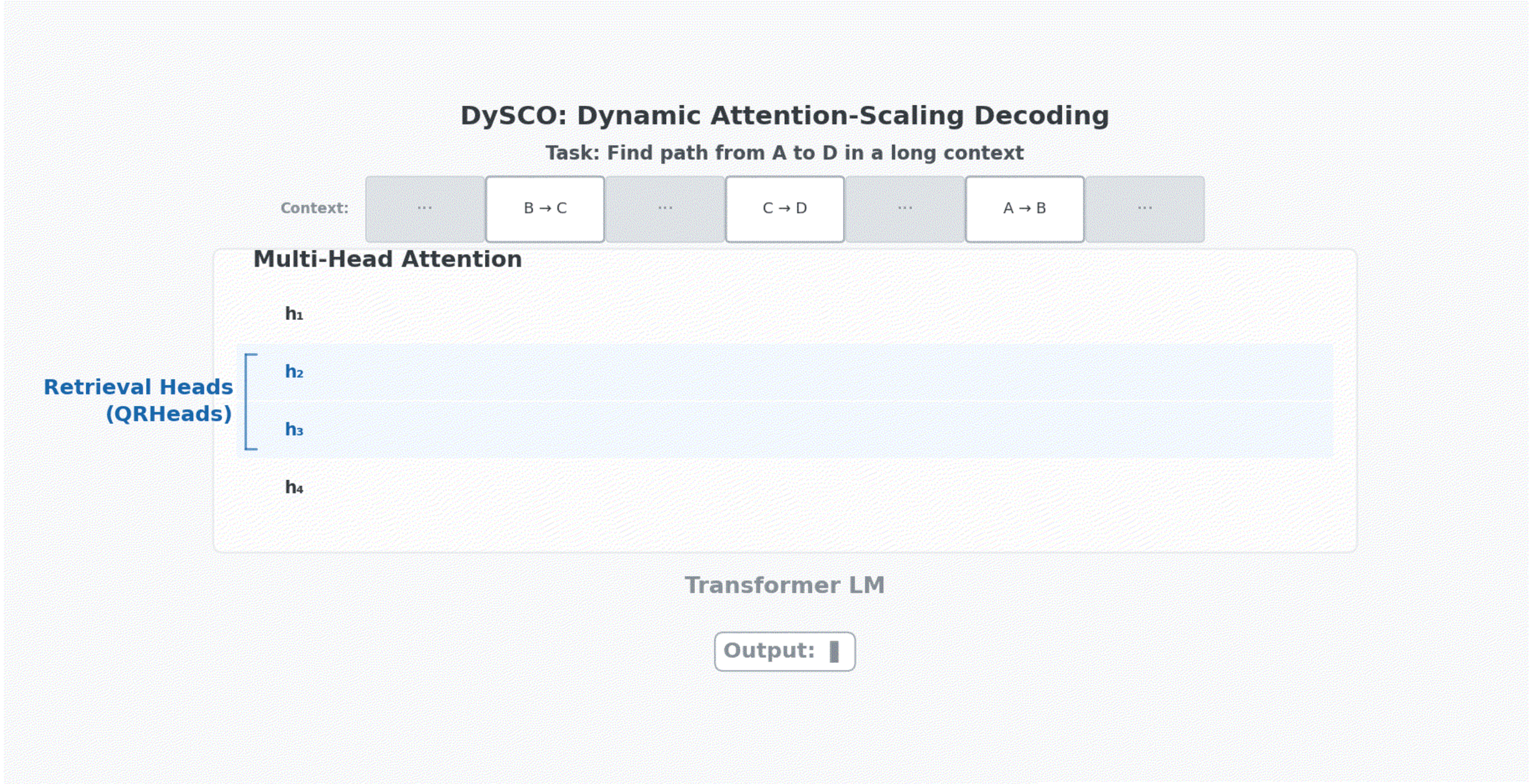
**[TASK]** Extract the following properties from the items listed on the webpages: Title, Year, Genre, Rating




| Title | Year | Genre | Rating |
|---|---|---|---|
| Gladiator II | 2024 | Action, Adventure | 7.0 |
| Arcane | 2021-2024 | Animation, Action | 9.0 |
| Deadpool & | 2024 | Action, Adventure | 7.7 |
| Red One | 2024 | Adventure, Comedy | 6.9 |
| Lioness | 2023 | Action, Thriller | 7.7 |
| ..... | | | |

Can we directly tackle the challenges in long-context reasoning (attending to key information at each generation step)?






Fangcong Yin @NYU

DySCO: Dynamic Attention-Scaling Decoding for Long-Context LMs

Xi Ye [*1] Wuwei Zhang [*1] Fangcong Yin [2] Howard Yen [1] Danqi Chen [1]

Do we have to process all the context at once? 🤔

We can build agentic scaffolding that allows LMs to process context piece by piece.

# This lecture

Enabling LLMs to process long context

Long-context mid-training (in LLM tech reports)

A full-stack overview of long context training

Long context and long chain-of-thought reasoning

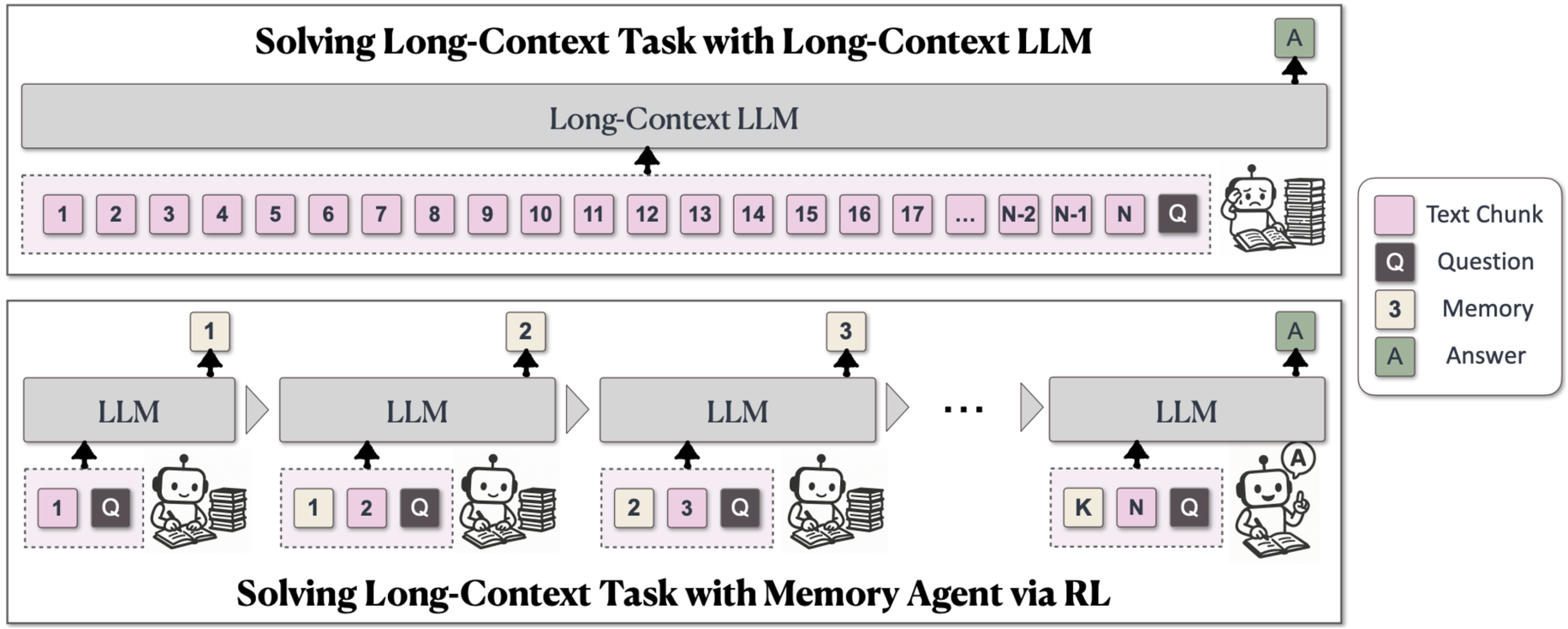
Scaffolding for long context

# Scaffolding for Long Context

Designing agentic systems that allows LMs to interact with the context, e.g.

- Slicing context and viewing a slice
- Retrieving from context
- Taking summary notes
- .....

# Scaffolding: Memory Systems




Divide the document into chunks; process one chunk at a time and update the memory

**MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-based Memory Agent**


Hongli Yu[1,2,3], Tinghong Chen[2], Jiangtao Feng[2], Jiangjie Chen[1,3], Weinan Dai[1,2,3], Qiying Yu[1,2,3], Ya-Qin Zhang[2,3], Wei-Ying Ma[2,3], Jingjing Liu[2,3], Mingxuan Wang[1,3], Hao Zhou[2,3]

[1]ByteDance Seed [2]Institute for AI Industry Research (AIR), Tsinghua University
[3]SIA-Lab of Tsinghua AIR and ByteDance Seed

# Scaffolding: Memory Systems

Question: Which part of New York City is the director of *Big Stone Gap* based in?

**Chunk3**

...

Document 81: Big Stone Gap (film)

**Big Stone Gap** is a 2014 American drama romantic comedy film written and directed by **Adriana Trigiani** and produced by Donna Gigliotti for Altar Identity Studios, a subsidiary of Media Society.

...

Document 96: Adriana Trigiani

**Adriana Trigiani** is an Italian American best-selling author of sixteen books, television writer, film director, and entrepreneur based in **Greenwich Village, New York City**.

...

**Memory3**

Updated memory:

- Ghost is a Swedish record producing and songwriting team, composed of Ulf Lindström and Johan Ekhé, based in New York City. They are perhaps best known for writing and producing Swedish singer Robyn's three first studio albums.
- **Adriana Trigiani** is an Italian American best-selling author of sixteen books, television writer, film director, and entrepreneur based in Greenwich Village, New York City. Trigiani has published a novel a year since 2000.
- **Big Stone Gap** is a 2014 American drama romantic comedy film written and directed by Adriana Trigiani and produced by Donna Gigliotti for Altar Identity Studios, a subsidiary of Media Society. Based on Trigiani's 2000 best-selling novel of the same name, the story is set in the actual Virginia town of Big Stone Gap circa 1970s. The film had its world premiere at the Virginia Film Festival on November 6, 2014.
  The director of the romantic comedy "Big Stone Gap" is Adriana Trigiani, who is based in **Greenwich Village, New York City**.

Input chunk

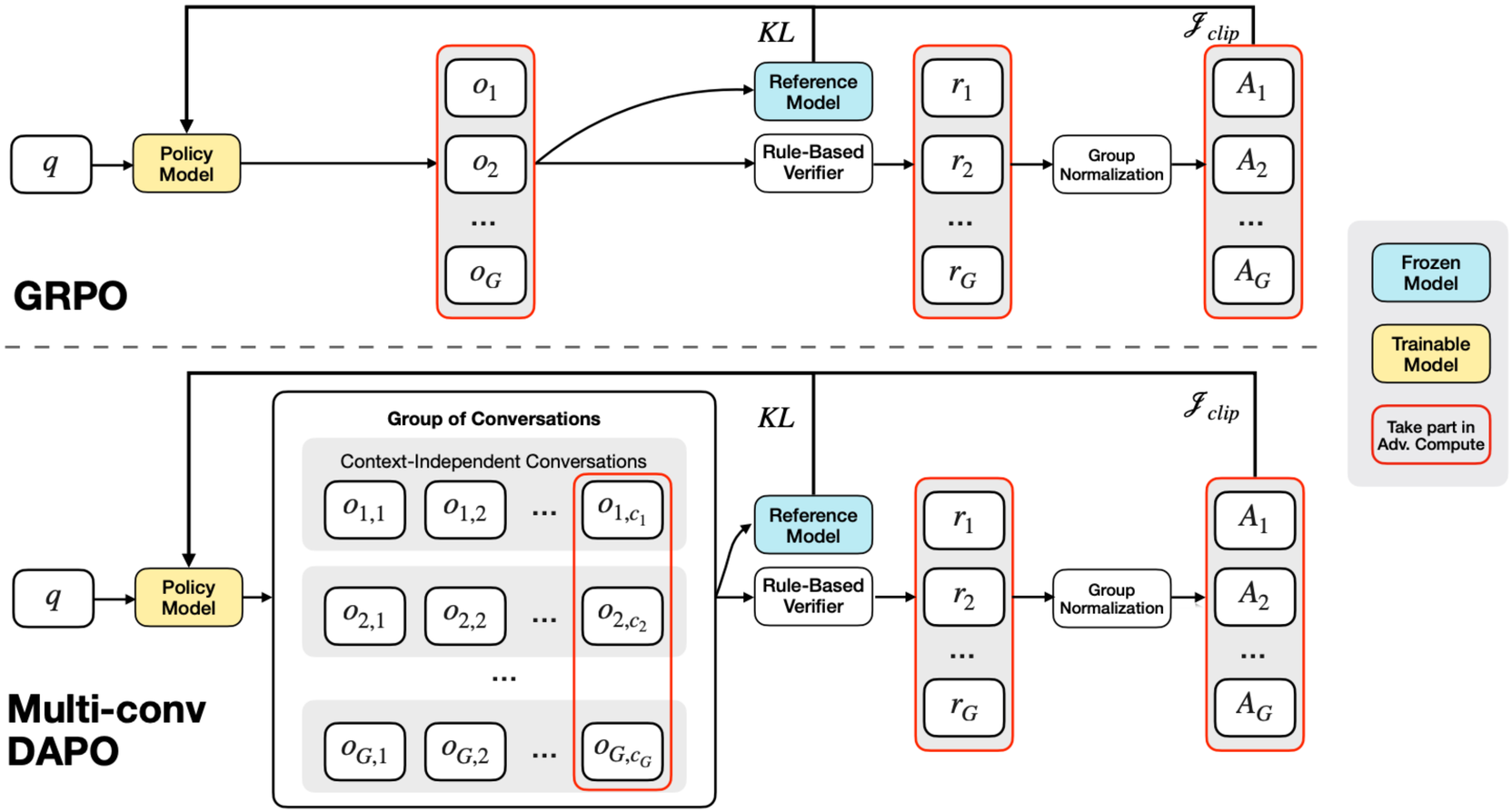
Updated memory (notes)

**MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-based Memory Agent**

Hongli Yu[1,2,3], Tinghong Chen[2], Jiangtao Feng[2], Jiangjie Chen[1,3], Weinan Dai[1,2,3], Qiying Yu[1,2,3], Ya-Qin Zhang[2,3], Wei-Ying Ma[2,3], Jingjing Liu[2,3], Mingxuan Wang[1,3], Hao Zhou[2,3]

[1]ByteDance Seed [2]Institute for AI Industry Research (AIR), Tsinghua University
[3]SIA-Lab of Tsinghua AIR and ByteDance Seed

# Scaffolding: Memory Systems



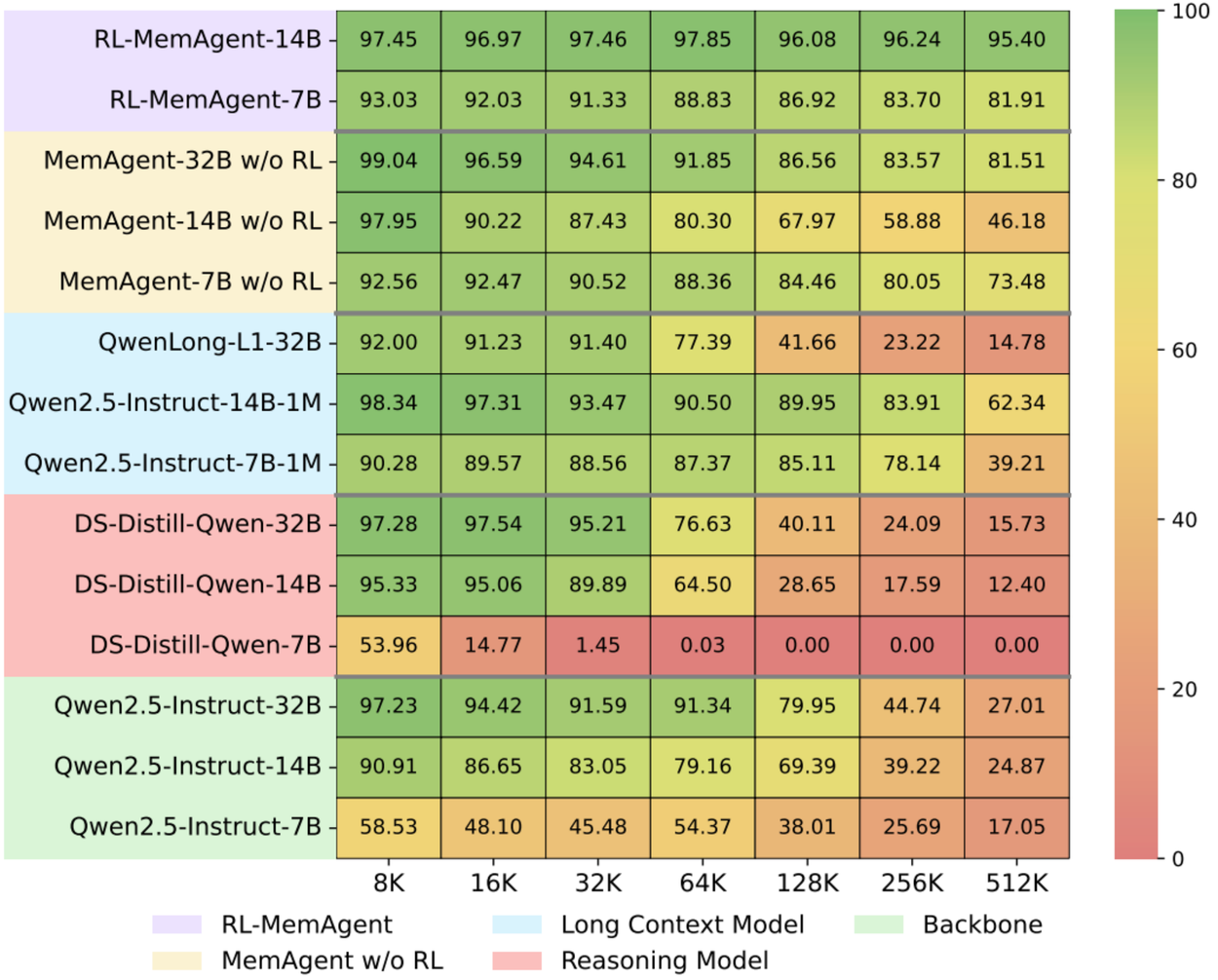

Train **end-to-end with RL**; propagate the reward at the end of the multi-turn conversations to all preceding conversations

**MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-based Memory Agent**


Hongli Yu[1,2,3], Tinghong Chen[2], Jiangtao Feng[2], Jiangjie Chen[1,3], Weinan Dai[1,2,3], Qiying Yu[1,2,3], Ya-Qin Zhang[2,3], Wei-Ying Ma[2,3], Jingjing Liu[2,3], Mingxuan Wang[1,3], Hao Zhou[2,3]

[1]ByteDance Seed [2]Institute for AI Industry Research (AIR), Tsinghua University
[3]SIA-Lab of Tsinghua AIR and ByteDance Seed

# Scaffolding: Memory Systems



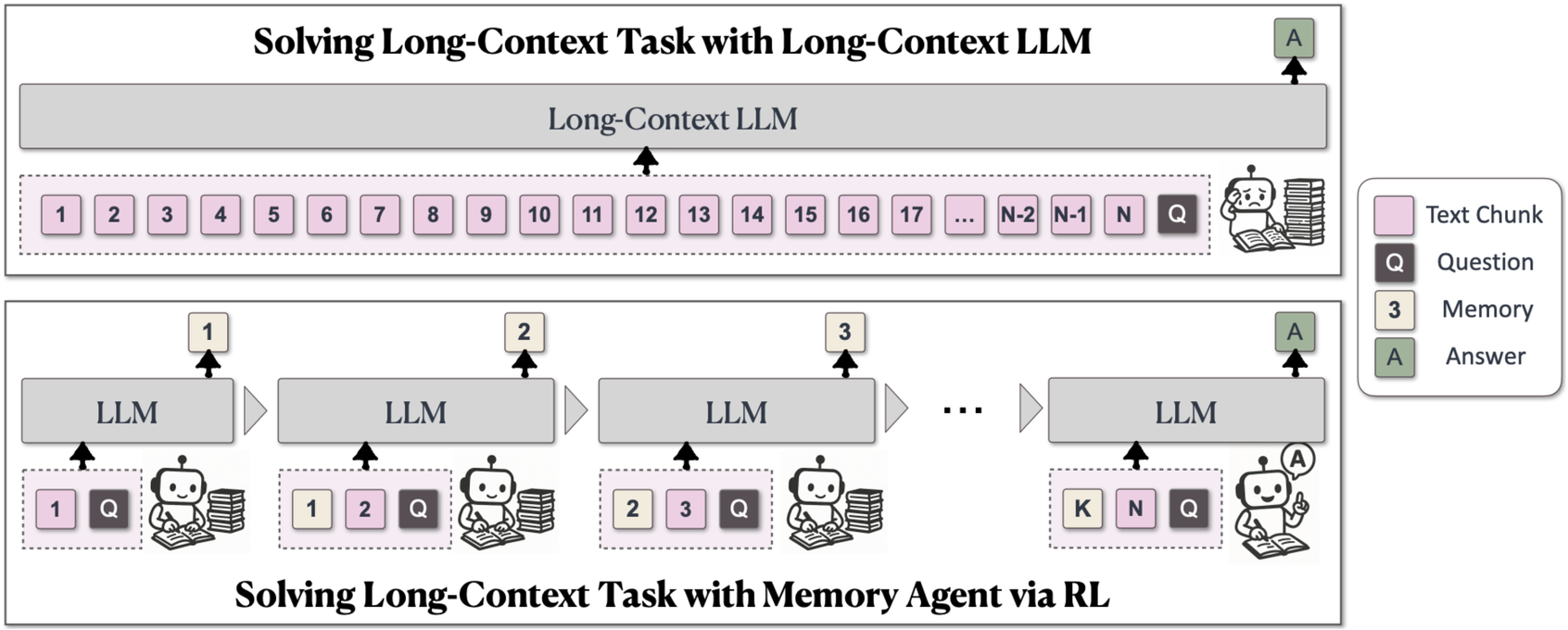

Preserve performance with increasing context length as LMs always only process one chunk at each time

**MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-based Memory Agent**


Hongli Yu[1,2,3], Tinghong Chen[2], Jiangtao Feng[2], Jiangjie Chen[1,3], Weinan Dai[1,2,3], Qiying Yu[1,2,3], Ya-Qin Zhang[2,3], Wei-Ying Ma[2,3], Jingjing Liu[2,3], Mingxuan Wang[1,3], Hao Zhou[2,3]

[1]ByteDance Seed [2]Institute for AI Industry Research (AIR), Tsinghua University
[3]SIA-Lab of Tsinghua AIR and ByteDance Seed

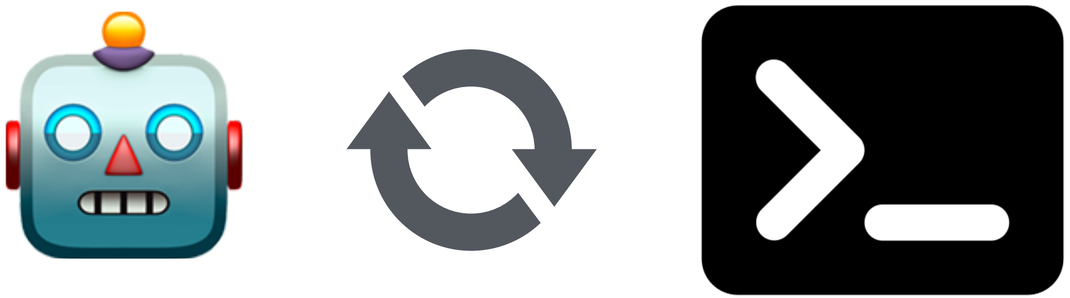
# Scaffolding: Memory Systems




Issues with MemAgent: LMs can **never revisit** a chunk

We need smarter, more autonomous exploration mechanism for agents

# Scaffolding: Recursive Language Models



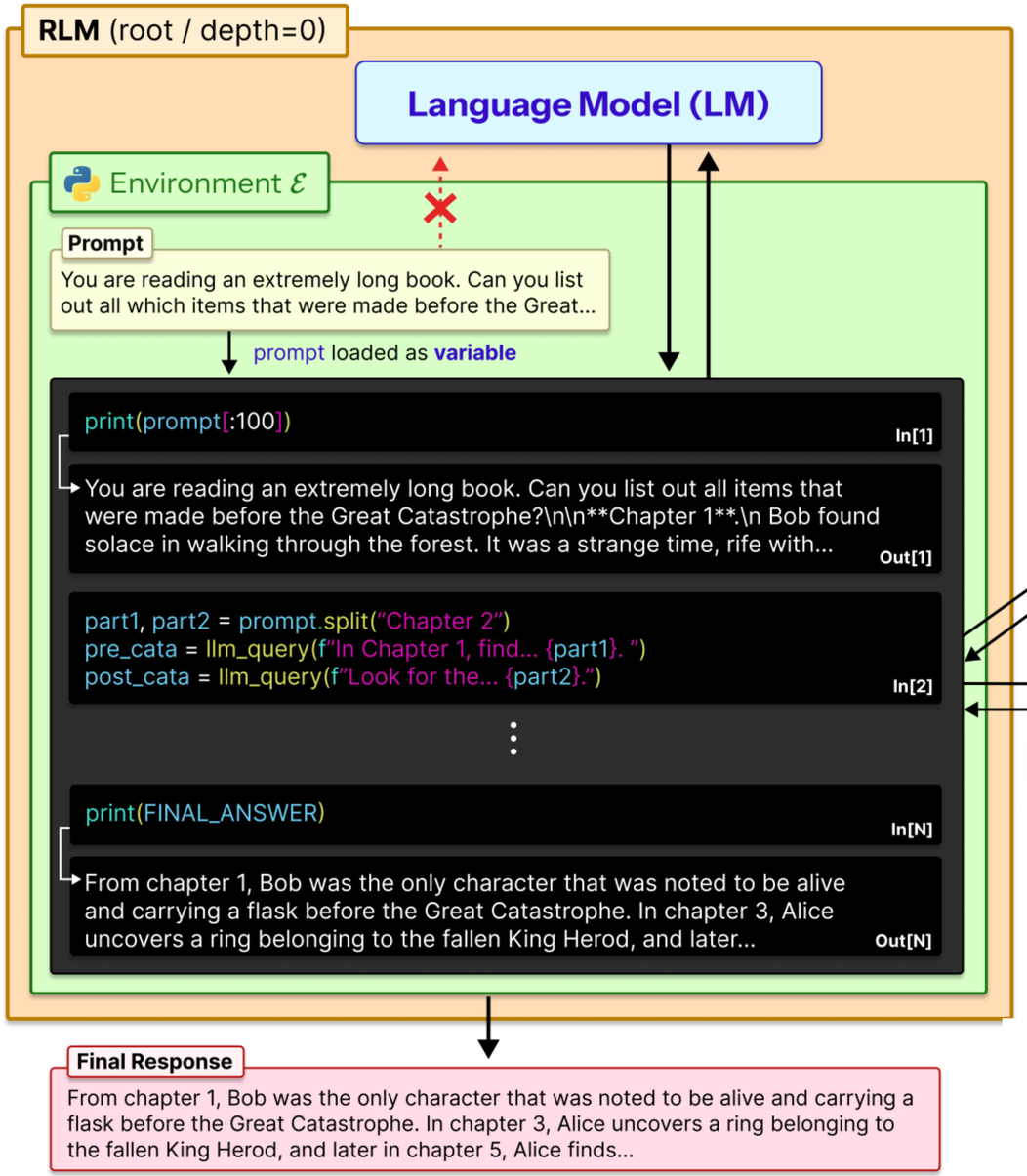
Wrap context in a executable coding environment (REPL)

**Recursive Language Models**

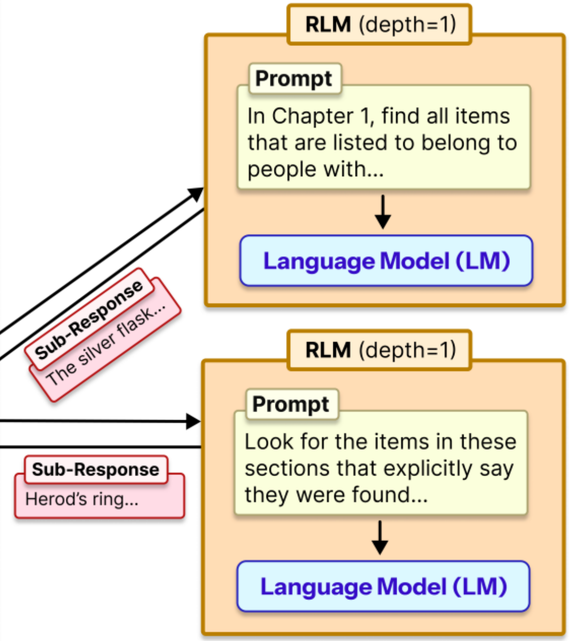
Alex L. Zhang [1] Tim Kraska [1] Omar Khattab [1]

# Scaffolding: Recursive Language Models

Flexible interaction with contexts via code



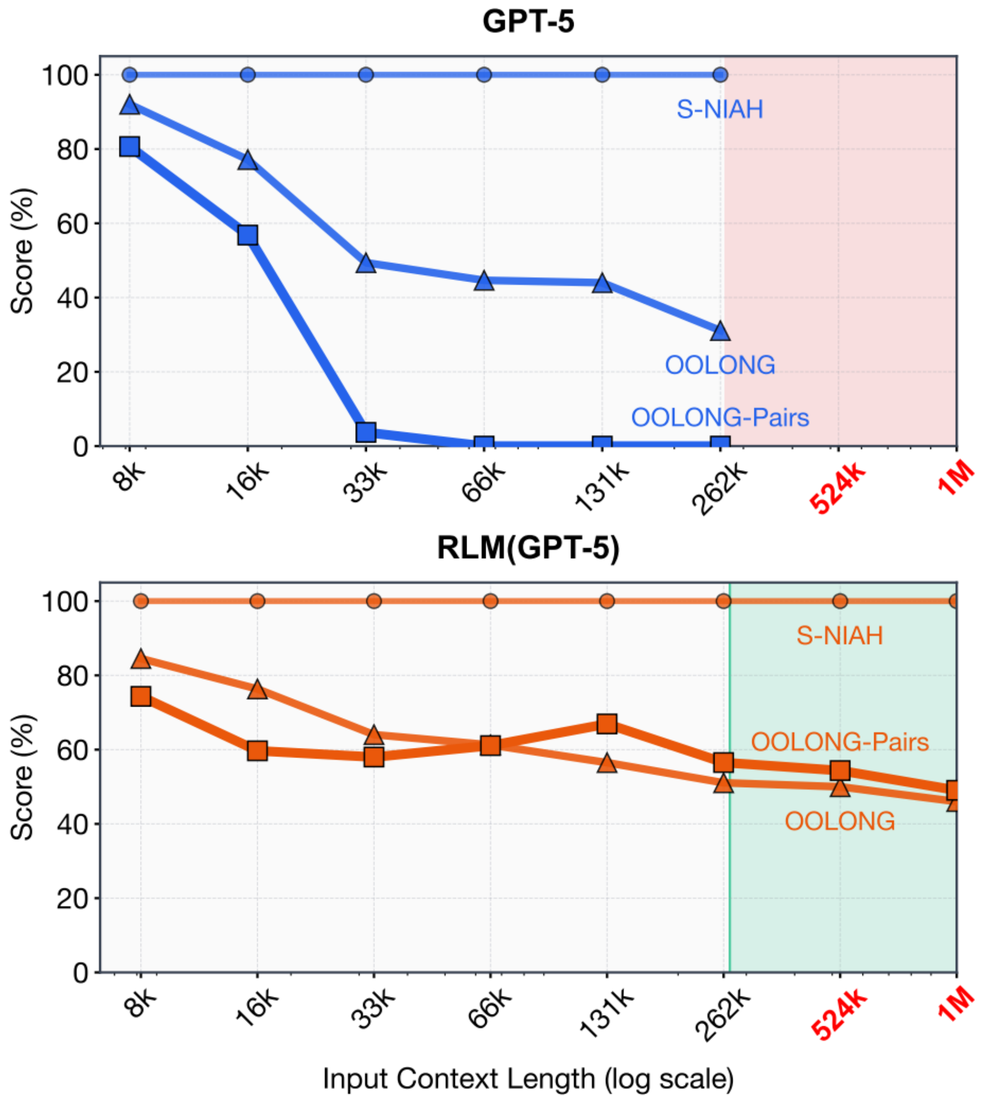

Allow calling sub-LMs (with customized prompts assembled with code




**Recursive Language Models**

Alex L. Zhang [1] Tim Kraska [1] Omar Khattab [1]

# Scaffolding: Recursive Language Models



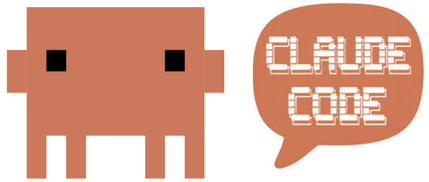

Preserve performance with increasing context length

**Recursive Language Models**

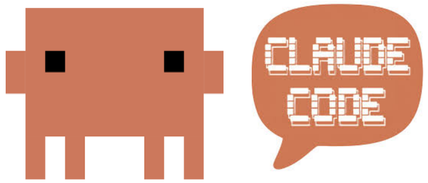
Alex L. Zhang [1] Tim Kraska [1] Omar Khattab [1]

# Scaffolding: General Coding Agents

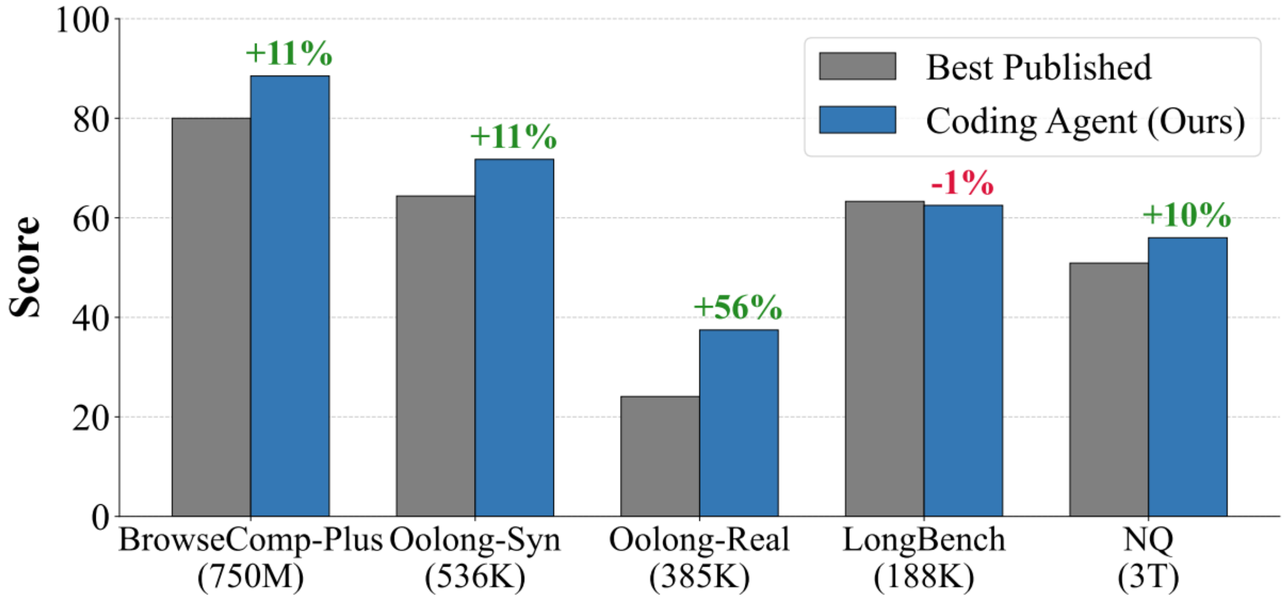
Can it be even more flexible?




# Scaffolding: General Coding Agents



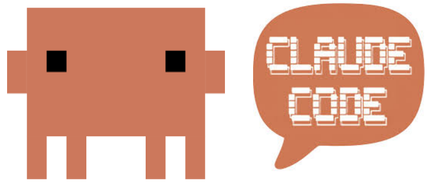




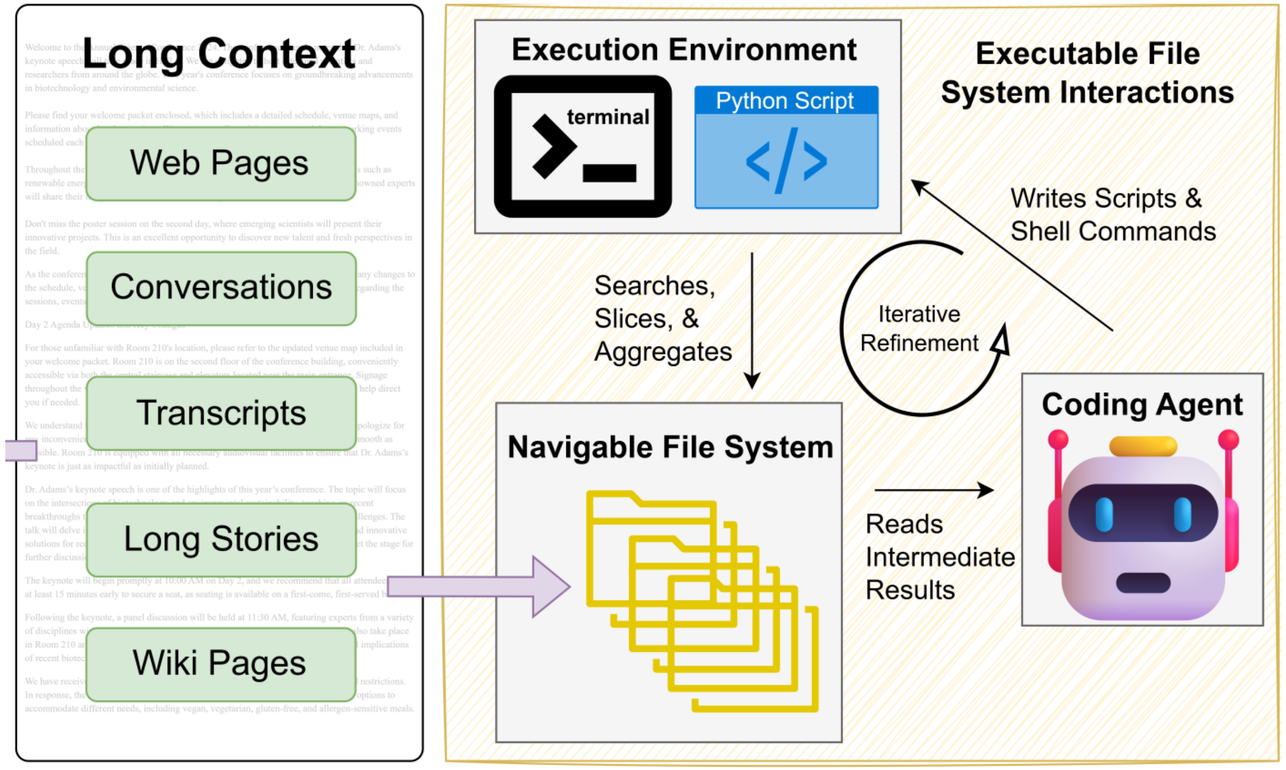

**Coding Agents are Effective Long-Context Processors**

Weili Cao [1] Xunjian Yin [1] Bhuwan Dhingra [*1] Shuyan Zhou [*1]

# Scaffolding: General Coding Agents





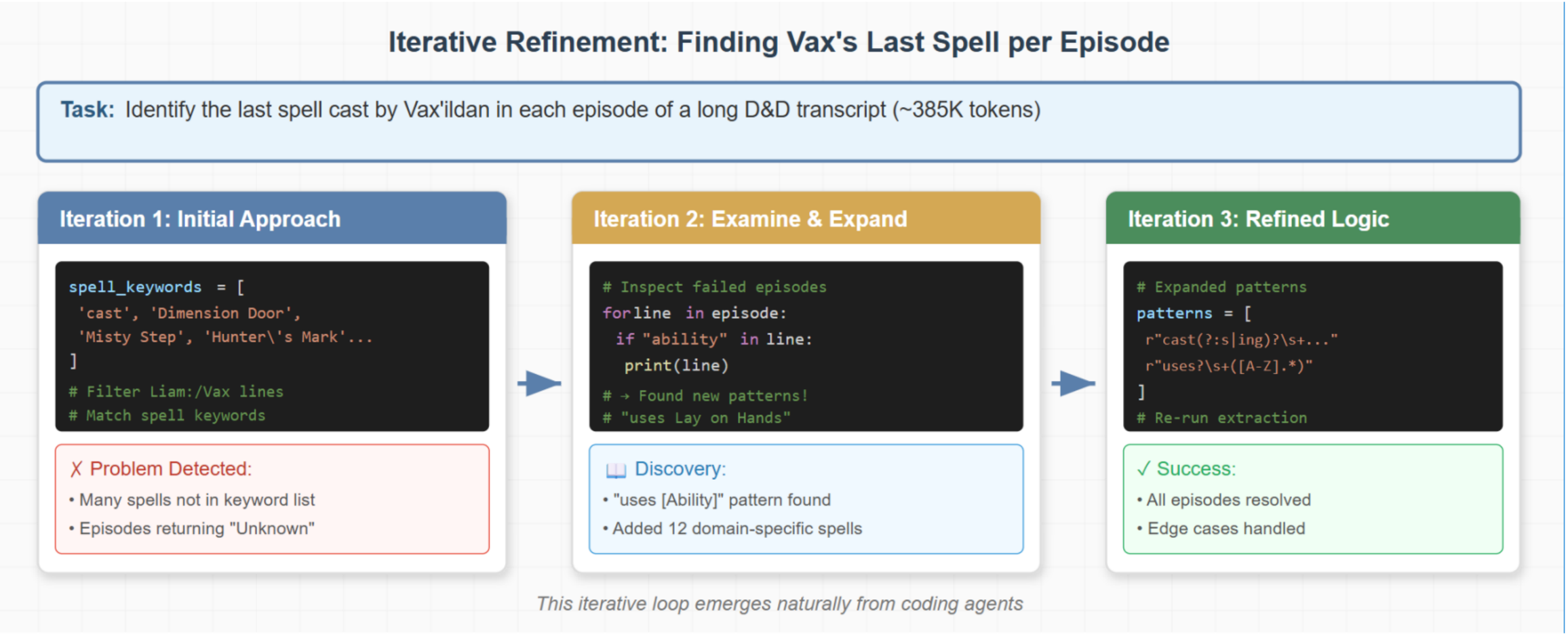



Present context to coding agents as file systems

**Coding Agents are Effective Long-Context Processors**

Weili Cao [1] Xunjian Yin [1] Bhuwan Dhingra [*1] Shuyan Zhou [*1]

# Scaffolding: General Coding Agents




Emerging strategy from coding agent (Claude Code)

# Summary: Scaffolding

For LMs with strong agentic capabilities, we can present context as an interactive environment

# What's Next

**Better synthetic data for long-context training**

**RL training recipe for long-context scaffolding**

- Better scaffolding design
- We need better open-source data (or data gyms)

# This lecture

Enabling LLMs to process long context

Long-context mid-training (in LLM tech reports)

A full-stack overview of long context training

Long context and long chain-of-thought reasoning

Scaffolding for long context