

Bets on scaling long-running agents: the long and short of it

Daniel Khashabi



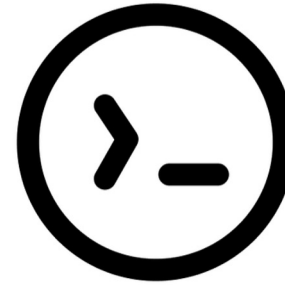
State of today: Agents are all the rage



Gemini Code



Claude Code



Codex

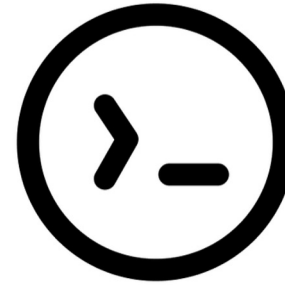
State of today: Agents are all the rage



Gemini Code



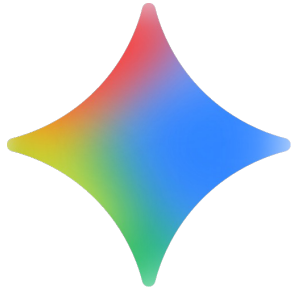
Claude Code



Codex

Coding agents:
Implement a feature, run
tests, and revise until it works.

State of today: Agents are all the rage



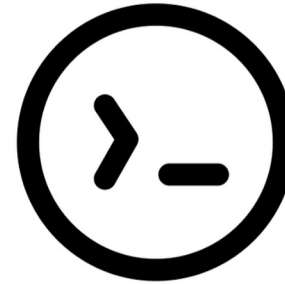
Copilots



Gemini Code

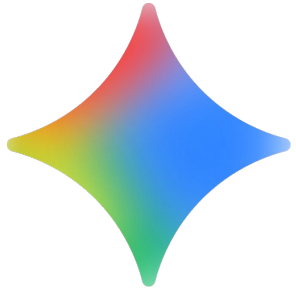


Claude Code



Codex

State of today: Agents are all the rage



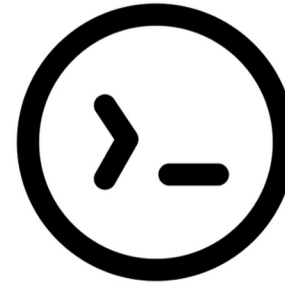
Copilots



Gemini Code



Claude Code

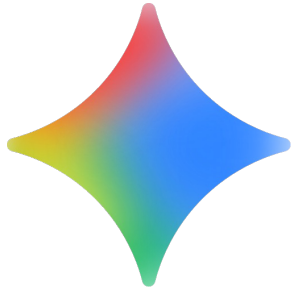


Codex

Day-to-day copilots:

Coordinate work across users, emails, documents, and past interactions.

State of today: Agents are all the rage



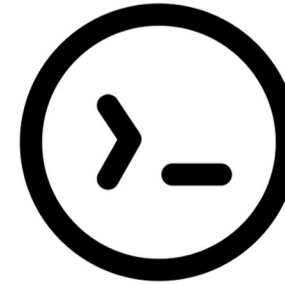
Copilots



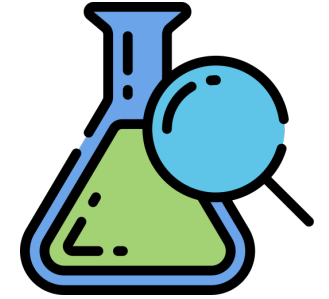
Gemini Code



Claude Code

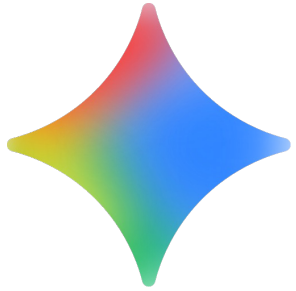


Codex



Discovery agents

State of today: Agents are all the rage



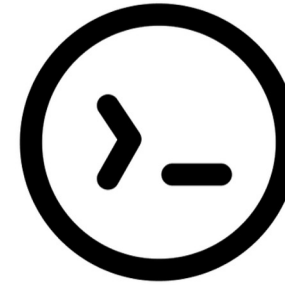
Copilots



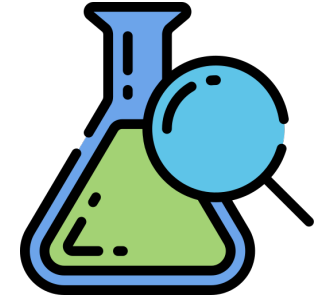
Gemini Code



Claude Code



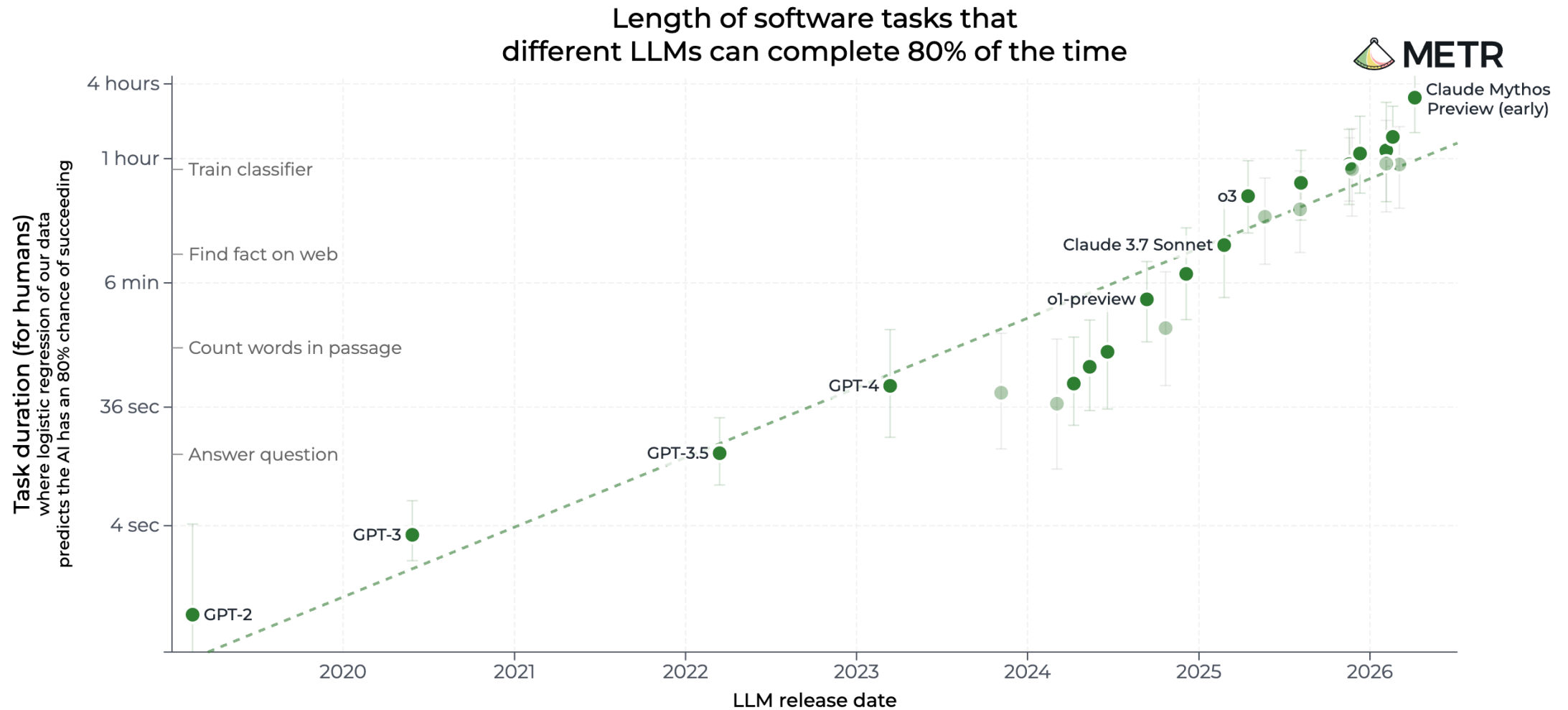
Codex



Discovery agents

Research and discovery agents:
Develop hypotheses, run experiments,
and revise plans as evidence arrives.

State of today: Agents progress in solving longer-horizon tasks



The myth of “long-running” agents

- Most individuals or organizations maintain ongoing relationships and commitments for years.
 - **Following a patient:** remembering their history and adapting as their condition changes, *over years*.
 - **Managing a farm:** learning from harvests and responding to weather and soil changes, *across seasons*.

The myth of “long-running” agents

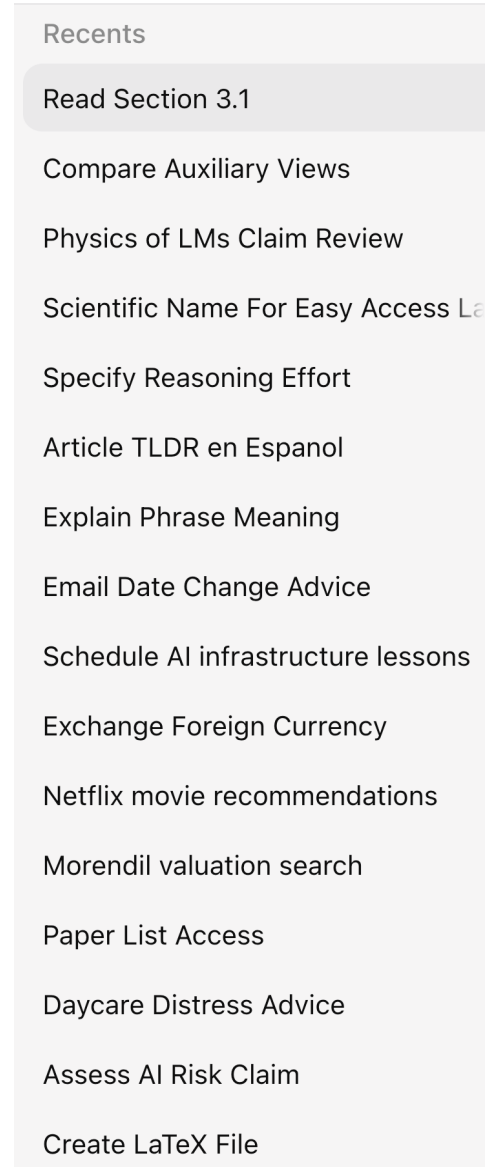
- Most individuals or organizations maintain ongoing relationships and commitments for years.
 - **Following a patient:** remembering their history and adapting as their condition changes, *over years*.
 - **Managing a farm:** learning from harvests and responding to weather and soil changes, *across seasons*.

The myth of “long-running” agents

- Most individuals or organizations maintain ongoing relationships and commitments for years.
 - **Following a patient:** remembering their history and adapting as their condition changes, *over years*.
 - **Managing a farm:** learning from harvests and responding to weather and soil changes, *across seasons*.

Also today: work fragmented across sessions

Also today: work fragmented across sessions



Also today: work fragmented across sessions

Recents

Read Section 3.1

Compare Auxiliary Views

Physics of LMs Claim Review

Scientific Name For Easy Access La

Specify Reasoning Effort

Article TLDR en Espanol

Explain Phrase Meaning

Email Date Change Advice

Schedule AI infrastructure lessons

Exchange Foreign Currency

Netflix movie recommendations

Morendil valuation search

Paper List Access

Daycare Distress Advice

Assess AI Risk Claim

Create LaTeX File

Chats and tasks

- Scientific name for cognitive laziness with u
- Red region mass proportion under curve
- Converting table to bar plot
- Symbols for multimodal data affiliations
- LLM serving caching strategies
- Converting y-axis to logarithmic scale
- Continual learning research at JHU
- MacBook Pro pricing and specs
- Major discovery benchmarks
- PyTorch negative infinity and tilde operator
- Interpretation request
- Work summary in bullet points
- Converting plots to inverse frequency
- Text message composition
- Zipfian distribution frequency basis
- LaTeX command definitions for document sy
- Understanding paged attention

Also today: work fragmented across sessions

Today's agents organize work into **isolated** chats, projects, and environments.

Recents

Read Section 3.1

Compare Auxiliary Views

Physics of LMs Claim Review

Scientific Name For Easy Access La

Specify Reasoning Effort

Article TLDR en Espanol

Explain Phrase Meaning

Email Date Change Advice

Schedule AI infrastructure lessons

Exchange Foreign Currency

Netflix movie recommendations

Morendil valuation search

Paper List Access

Daycare Distress Advice

Assess AI Risk Claim

Create LaTeX File

Chats and tasks

- Scientific name for cognitive laziness with u
- Red region mass proportion under curve
- Converting table to bar plot
- Symbols for multimodal data affiliations
- LLM serving caching strategies
- Converting y-axis to logarithmic scale
- Continual learning research at JHU
- MacBook Pro pricing and specs
- Major discovery benchmarks
- PyTorch negative infinity and tilde operator
- Interpretation request
- Work summary in bullet points
- Converting plots to inverse frequency
- Text message composition
- Zipfian distribution frequency basis
- LaTeX command definitions for document sy
- Understanding paged attention

Also today: work fragmented across sessions

Today's agents organize work into **isolated** chats, projects, and environments.

What is learned or decided in one session doesn't reliably carry into others as the work evolves.

Recents

Read Section 3.1

Compare Auxiliary Views

Physics of LMs Claim Review

Scientific Name For Easy Access La

Specify Reasoning Effort

Article TLDR en Espanol

Explain Phrase Meaning

Email Date Change Advice

Schedule AI infrastructure lessons

Exchange Foreign Currency

Netflix movie recommendations

Morendil valuation search

Paper List Access

Daycare Distress Advice

Assess AI Risk Claim

Create LaTeX File

Chats and tasks

- Scientific name for cognitive laziness with u
- Red region mass proportion under curve
- Converting table to bar plot
- Symbols for multimodal data affiliations
- LLM serving caching strategies
- Converting y-axis to logarithmic scale
- Continual learning research at JHU
- MacBook Pro pricing and specs
- Major discovery benchmarks
- PyTorch negative infinity and tilde operator
- Interpretation request
- Work summary in bullet points
- Converting plots to inverse frequency
- Text message composition
- Zipfian distribution frequency basis
- LaTeX command definitions for document sy
- Understanding paged attention

The future is *ever-longer-running* agents

- Agents that reason and adapt, 24/7, over weeks and months.
- **Example:** The agent that coordinated last month's event remembers decisions, tracks commitments, and adapts plans for the next phase.
- **One** agent throughout; not many isolated environments.

The future is *ever-longer-running* agents

- Agents that reason and adapt, 24/7, over weeks and months.
- **Example:** The agent that coordinated last month's event remembers decisions, tracks commitments, and adapts plans for the next phase.
- **One** agent throughout; not many isolated environments.

The future is *ever-longer-running* agents

- Agents that reason and adapt, 24/7, over weeks and months.
- **Example:** The agent that coordinated last month's event remembers decisions, tracks commitments, and adapts plans for the next phase.
- **One** agent throughout; not many isolated environments.

The future is *ever-longer-running* agents

- Agents that reason and adapt, 24/7, over weeks and months.
- **Example:** The agent that coordinated last month's event remembers decisions, tracks commitments, and adapts plans for the next phase.
- **One** agent throughout; not many isolated environments.

How do we scale **long-running** agents?

- Bet 1: **Expand context:**
 - Keep more of the history *directly available*.

Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters:**
 - Update model parameters through *continued* training
- Bet 3: **Manage context effectively:**
 - Selectively preserve what matters through *compression*.

How do we scale **long-running** agents?

- Bet 1: **Expand context:**
 - Keep more of the history *directly available*.

Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters:**
 - Update model parameters through *continued* training
- Bet 3: **Manage context effectively:**
 - Selectively preserve what matters through *compression*.

How do we scale **long-running** agents?

- Bet 1: **Expand context**:
 - Keep more of the history *directly available*.

Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters**:
 - Update model parameters through *continued* training
- Bet 3: **Manage context effectively**:
 - Selectively preserve what matters through *compression*.

How do we scale **long-running** agents?

- Bet 1: **Expand context**:
 - Keep more of the history *directly available*.

Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters**:
 - Update model parameters through *continued* training
- Bet 3: Manage **context effectively**:
 - Selectively preserve what matters through *compression*.

How do we scale **long-running** agents?

- Bet 1: **Expand context**:
 - Keep more of the history *directly available*.

Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters**:
 - Update model parameters through *continued* training
- Bet 3: **Manage context effectively**:
 - Selectively preserve what matters through *compression*.

Roadmap: long-running agents

- Bet 1: **Expand context:**
 - Keep more of the history *directly available*.



Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters:**
 - Update model parameters through *continued* training
- Bet 3: Manage **context effectively:**
 - Selectively preserve what matters through *compression*.

Roadmap: long-running agents

- Bet 1: **Expand context:**
 - Keep more of the history *directly available*.



Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters:**
 - Update model parameters through *continued* training
- Bet 3: **Manage context effectively:**
 - Selectively preserve what matters through *compression*.

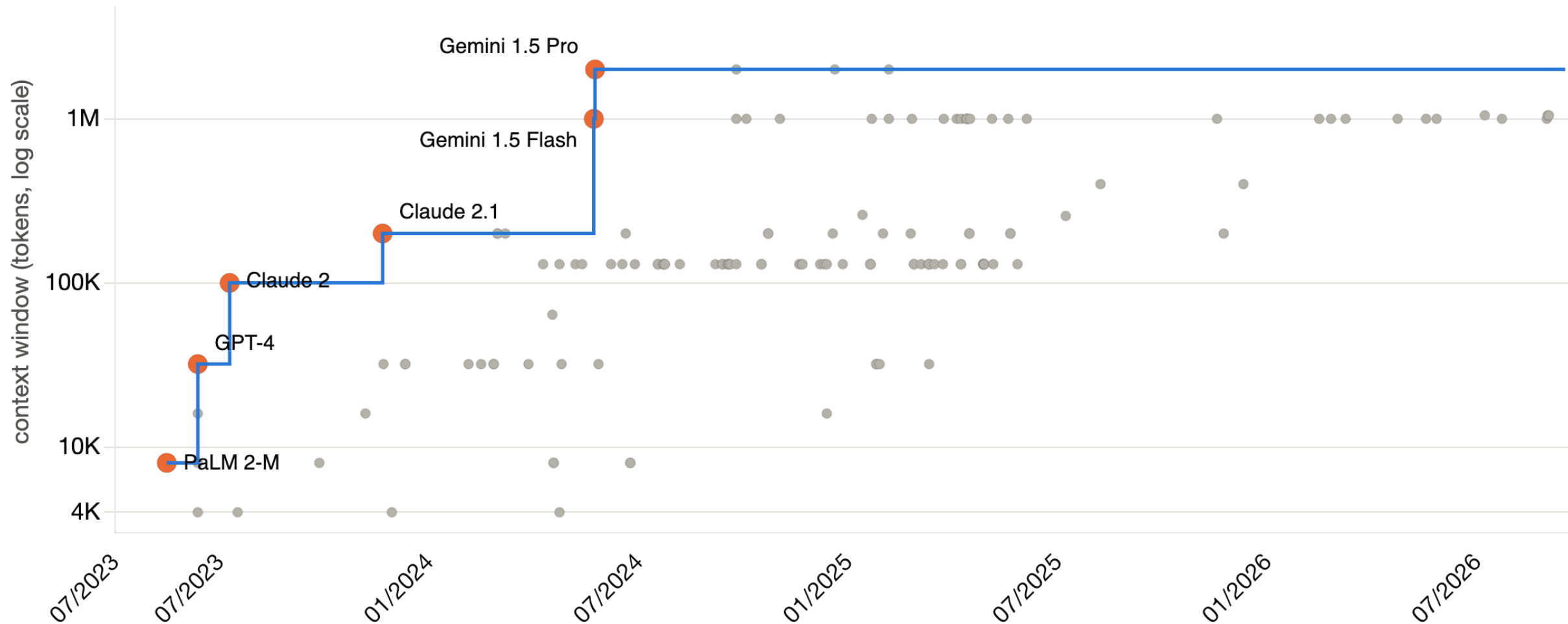
The bet on ever-scaling context windows

- Keep more of an agent's history directly available.
- Most SOTA LLMs* have context window of >1M tokens.
 - GPT6-Astra, Opus 5, Gemini 3.8

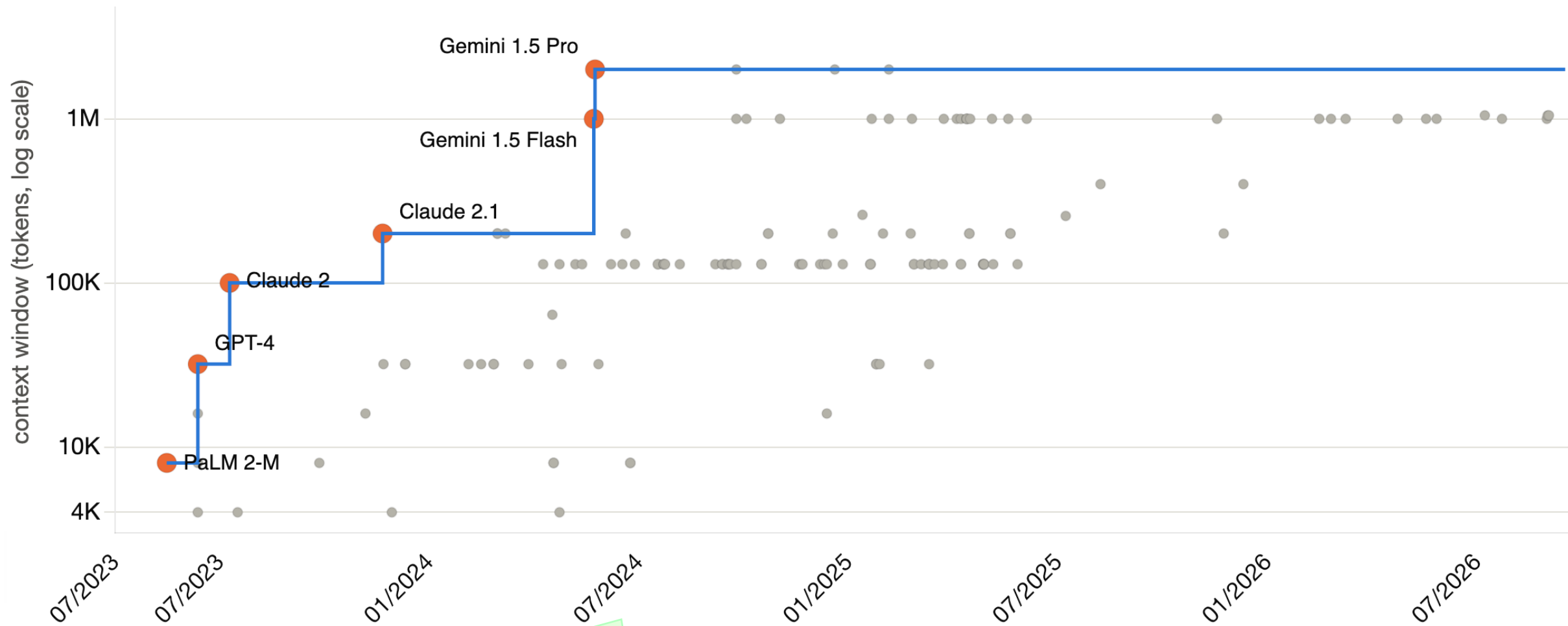
The bet on ever-scaling context windows

- Keep more of an agent's history directly available.
- Most SOTA LLMs* have context window of >1M tokens.
 - GPT6-Astra, Opus 5, Gemini 3.8

The bet on ever-scaling context windows

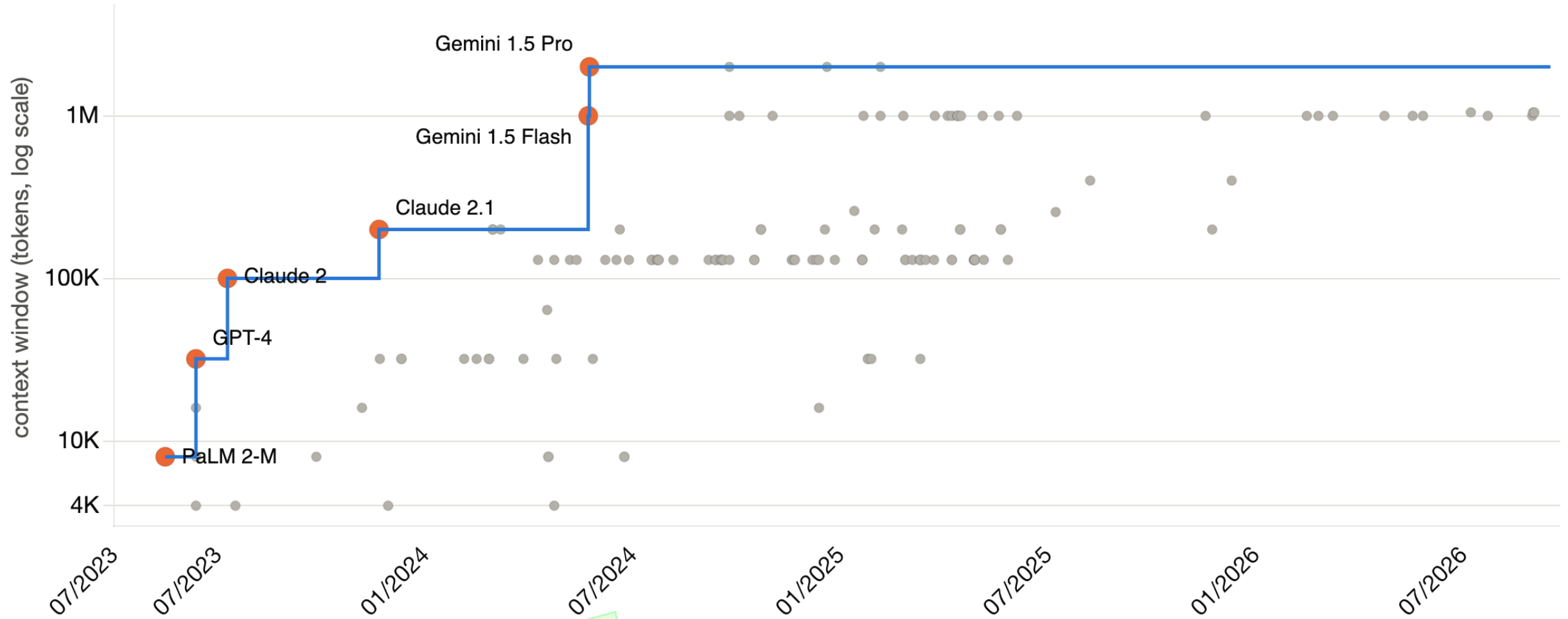


The bet on ever-scaling context windows



Consistent growth over time

The bet on ever-scaling context windows



Consistent growth over time

But dwindling progress? 🤔

The bet on ever-scaling context windows

- What needs to be unlocked to continue to scale it?
 1. **Efficiency barrier:** going beyond the *quadratic* cost.
 2. **Data barrier:** quality long-context data is *scarce*.

The bet on ever-scaling context windows

- What needs to be unlocked to continue to scale it?
1. **Efficiency barrier:** going beyond the *quadratic* cost.
 2. **Data barrier:** quality long-context data is *scarce*.

The bet on ever-scaling context windows

- What needs to be unlocked to continue to scale it?
 1. **Efficiency barrier:** going beyond the *quadratic* cost.
 2. **Data barrier:** quality long-context data is *scarce*.

The bet on ever-scaling context windows

- What needs to be unlocked to continue to scale it?
1. **Efficiency barrier:** going beyond the *quadratic* cost.
 2. **Data barrier:** quality long-context data is *scarce*.

What if long data is abundant (one day)?

A thought experiment on context scaling

- *A hypothetical future:*
 - Assume we have endless long-context data.
 - Also, assume sufficient compute.

A thought experiment on context scaling

- A *hypothetical* future:
 - Assume we have endless long-context data.
 - Also, assume sufficient compute.

A thought experiment on context scaling

- A *hypothetical* future:
 - Assume we have endless long-context data.
 - Also, assume sufficient compute.

A thought experiment on context scaling

- A *hypothetical* future:
 - Assume we have **endless long-context data**.
 - Also, assume sufficient **compute**.

Would simply increasing the training window reliably improve what the model learns?

A skeptical take

- **Cognitive offloading** := the tendency (of humans) to use external resources (notes, phones, search engines) to reduce mental effort.
 - **Example:** The “Google effect” (or "digital amnesia")
 - People who expect info to remain accessible, remember less.

Sparrow et al. Google effects on memory: cognitive consequences of having information at our fingertips, 2011.

- **Reaction to the context scaling thought experiment:**
 - Could more available context reduce what a model internalizes?

A skeptical take

- **Cognitive offloading** := the tendency (of humans) to use external resources (notes, phones, search engines) to reduce mental effort.
 - **Example:** The “Google effect” (or "digital amnesia")
 - People who expect info to remain accessible, remember less.

Sparrow et al. Google effects on memory: cognitive consequences of having information at our fingertips, 2011.

- **Reaction to the context scaling thought experiment:**
 - Could more available context reduce what a model internalizes?

A skeptical take

- **Cognitive offloading** := the tendency (of humans) to use external resources (notes, phones, search engines) to reduce mental effort.
 - **Example:** The “Google effect” (or "digital amnesia")
 - People who expect info to remain accessible, remember less.

Sparrow et al. Google effects on memory: cognitive consequences of having information at our fingertips, 2011.

- **Reaction to the context scaling thought experiment:**
 - Could more available context reduce what a model internalizes?

A skeptical take

- **Cognitive offloading** := the tendency (of humans) to use external resources (notes, phones, search engines) to reduce mental effort.
 - **Example:** The “Google effect” (or "digital amnesia")
 - People who expect info to remain accessible, remember less.

Sparrow et al. Google effects on memory: cognitive consequences of having information at our fingertips, 2011.

- **Reaction to the context scaling thought experiment:**
 - Could more available context reduce what a model internalizes?

A controlled experiment for scaling context

- **Fixed Data:** Long documents from Project Gutenberg
- **Fixed token budget** (10B tokens) and **tokens per batch** (1M tokens)
- **Vary the training window size:**
 $W_{\text{train}} \in \{512, 1\text{K}, 2\text{K}, 4\text{K}, 8\text{K}, 16\text{K}, 32\text{K}\}$
- **Evaluation:** existing (not-very-long) tasks.
- **Q:** How does this control manifest itself in downstream evals?

(*another variable is compute-per-model; longer models require more compute, despite being token-matched.)

A controlled experiment for scaling context

- **Fixed Data:** Long documents from Project Gutenberg
- **Fixed token budget** (10B tokens) and **tokens per batch** (1M tokens)
- **Vary the training window size:**
 $W_{\text{train}} \in \{512, 1\text{K}, 2\text{K}, 4\text{K}, 8\text{K}, 16\text{K}, 32\text{K}\}$
- **Evaluation:** existing (not-very-long) tasks.
- **Q:** How does this control manifest itself in downstream evals?

(*another variable is compute-per-model; longer models require more compute, despite being token-matched.)

A controlled experiment for scaling context

- **Fixed Data:** Long documents from Project Gutenberg
- **Fixed token budget** (10B tokens) and **tokens per batch** (1M tokens)
- **Vary the training window size:**
 $W_{\text{train}} \in \{512, 1\text{K}, 2\text{K}, 4\text{K}, 8\text{K}, 16\text{K}, 32\text{K}\}$
- **Evaluation:** existing (not-very-long) tasks.
- **Q:** How does this control manifest itself in downstream evals?

(*another variable is compute-per-model; longer models require more compute, despite being token-matched.)

A controlled experiment for scaling context

- **Fixed Data:** Long documents from Project Gutenberg
- **Fixed token budget** (10B tokens) and **tokens per batch** (1M tokens)
- **Vary the training window size:**

$$W_{\text{train}} \in \{512, 1\text{K}, 2\text{K}, 4\text{K}, 8\text{K}, 16\text{K}, 32\text{K}\}$$

- **Evaluation:** existing (not-very-long) tasks.
- **Q:** How does this control manifest itself in downstream evals?

(*another variable is compute-per-model; longer models require more compute, despite being token-matched.)

A controlled experiment for scaling context

- **Fixed Data:** Long documents from Project Gutenberg
- **Fixed token budget** (10B tokens) and **tokens per batch** (1M tokens)
- **Vary the training window size:**
 $W_{\text{train}} \in \{512, 1\text{K}, 2\text{K}, 4\text{K}, 8\text{K}, 16\text{K}, 32\text{K}\}$
- **Evaluation:** existing (not-very-long) tasks.
- **Q:** How does this control manifest itself in downstream evals?

(*another variable is compute-per-model; longer models require more compute, despite being token-matched.)

A controlled experiment for scaling context

- **Fixed Data:** Long documents from Project Gutenberg
- **Fixed token budget** (10B tokens) and **tokens per batch** (1M tokens)
- **Vary the training window size:**
 $W_{\text{train}} \in \{512, 1\text{K}, 2\text{K}, 4\text{K}, 8\text{K}, 16\text{K}, 32\text{K}\}$
- **Evaluation:** existing (not-very-long) tasks.
- **Q:** How does this control manifest itself in downstream evals?

(*another variable is compute-per-model; longer models require more compute, despite being token-matched.)

Context length: the gory details

Training time

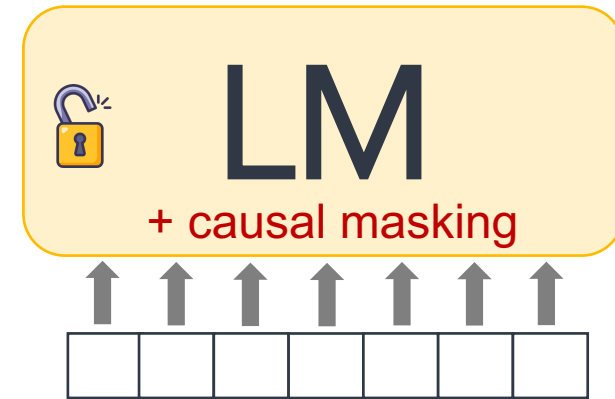
Context length: the gory details

Training time



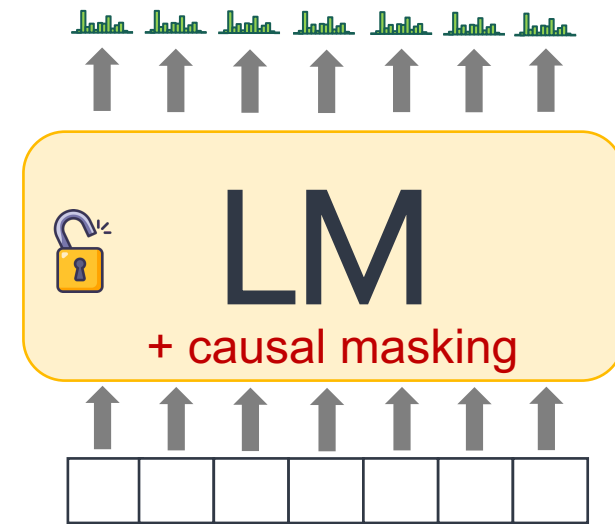
Context length: the gory details

Training time



Context length: the gory details

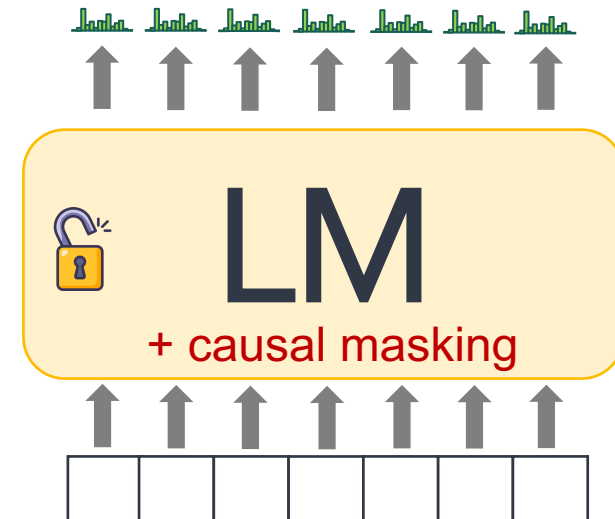
Training time



Context length: the gory details

Training time

CE loss: $\ell_t = -\log P(y_t | y_{<t}, X)$

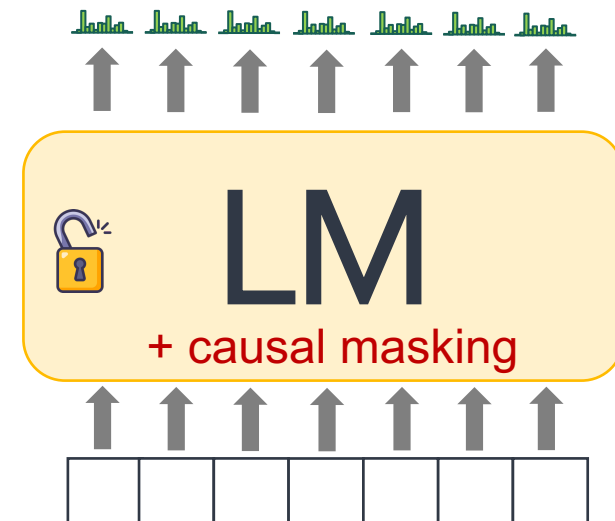


Context length: the gory details

Training time

CE loss: $\ell_t = -\log P(y_t | y_{<t}, X)$

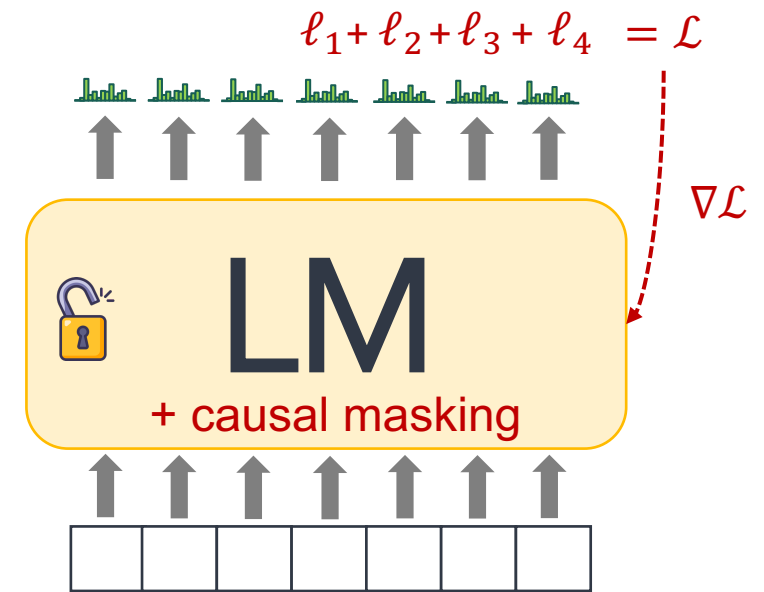
$$\ell_1 + \ell_2 + \ell_3 + \ell_4 = \mathcal{L}$$



Context length: the gory details

Training time

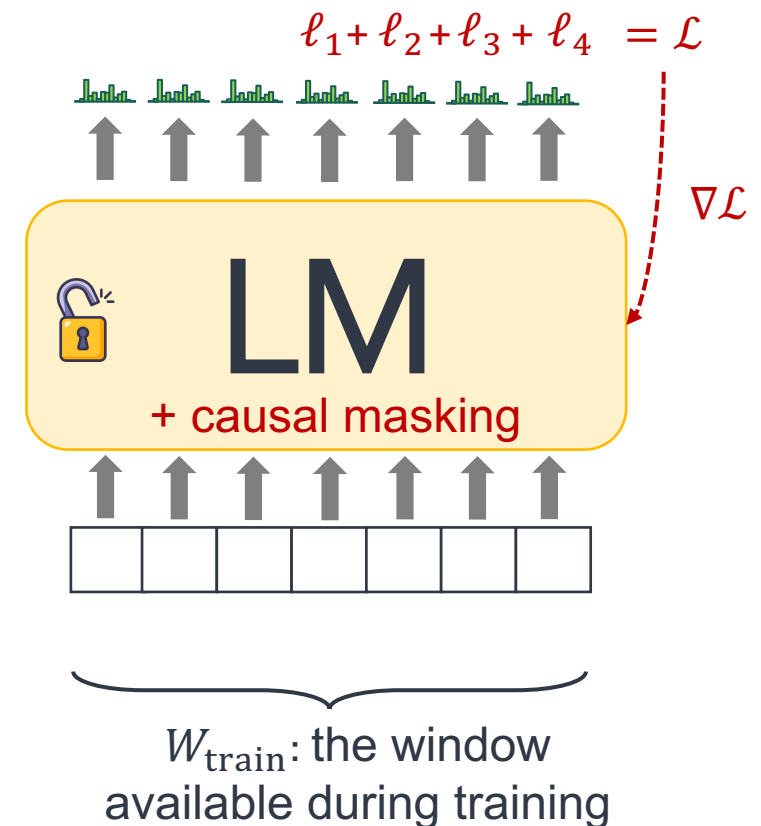
CE loss: $\ell_t = -\log P(y_t | y_{<t}, X)$



Context length: the gory details

Training time

CE loss: $\ell_t = -\log P(y_t | y_{<t}, X)$

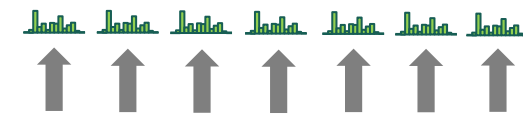


Context length: the gory details

Training time

CE loss: $\ell_t = -\log P(y_t | y_{<t}, X)$

$$\ell_1 + \ell_2 + \ell_3 + \ell_4 = \mathcal{L}$$



W_{train} : the window
available during training



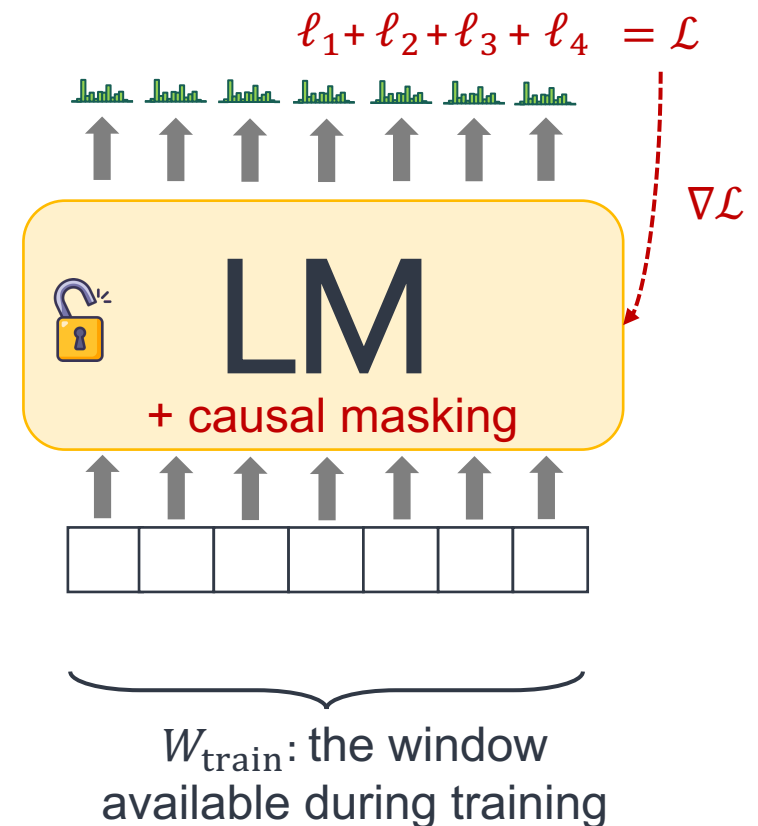
Context length: the gory details

Test time



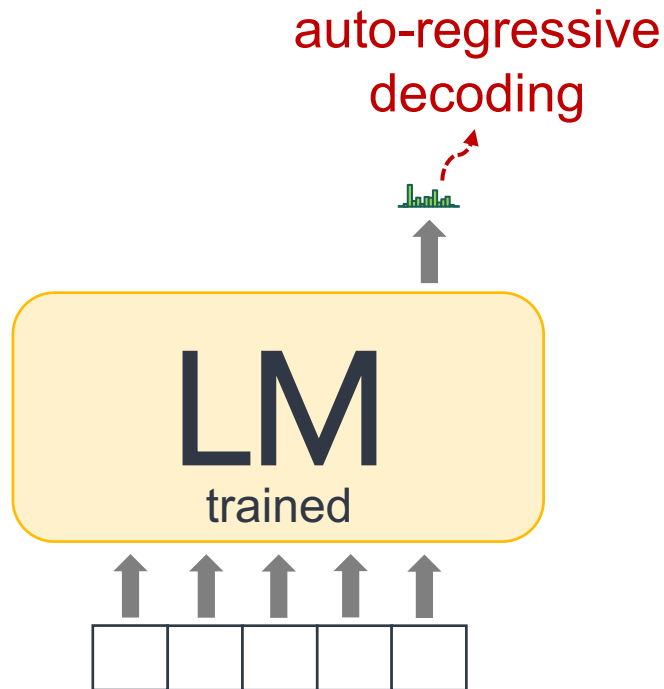
Training time

CE loss: $\ell_t = -\log P(y_t | y_{<t}, X)$



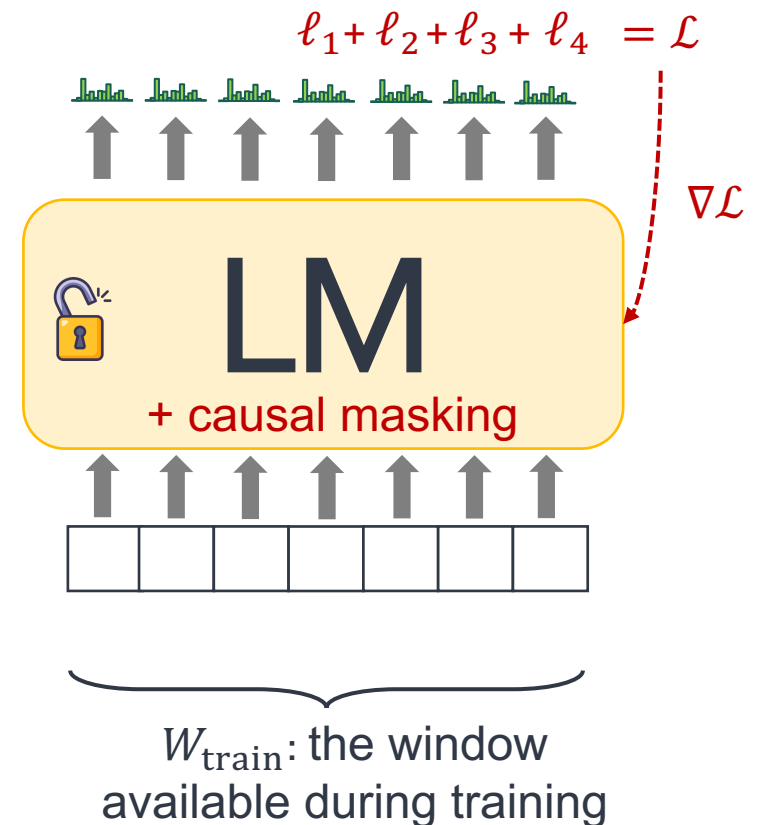
Context length: the gory details

Test time



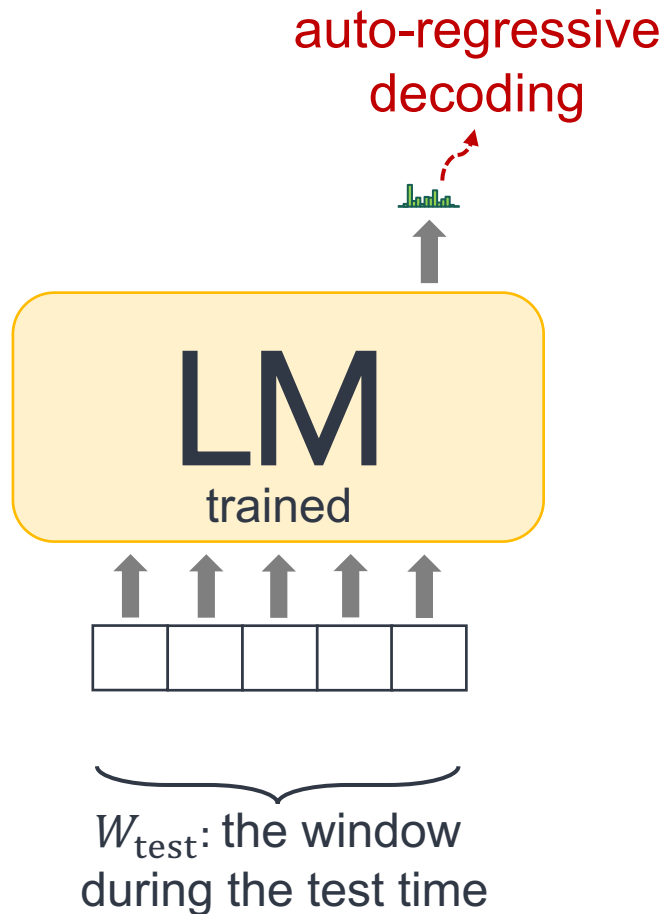
Training time

CE loss: $\ell_t = -\log P(y_t | y_{<t}, X)$

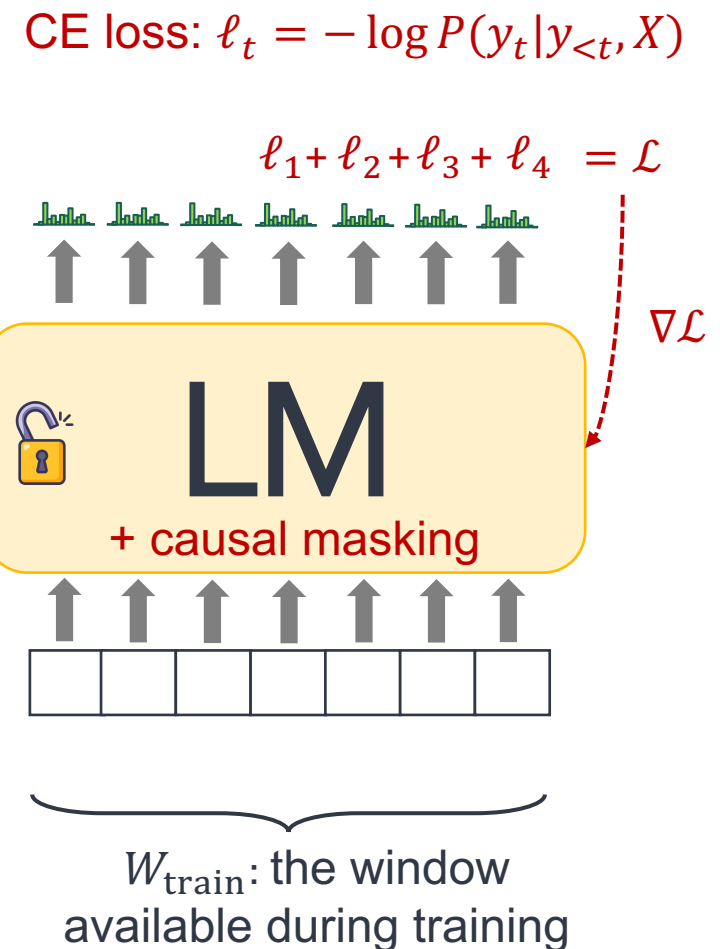


Context length: the gory details

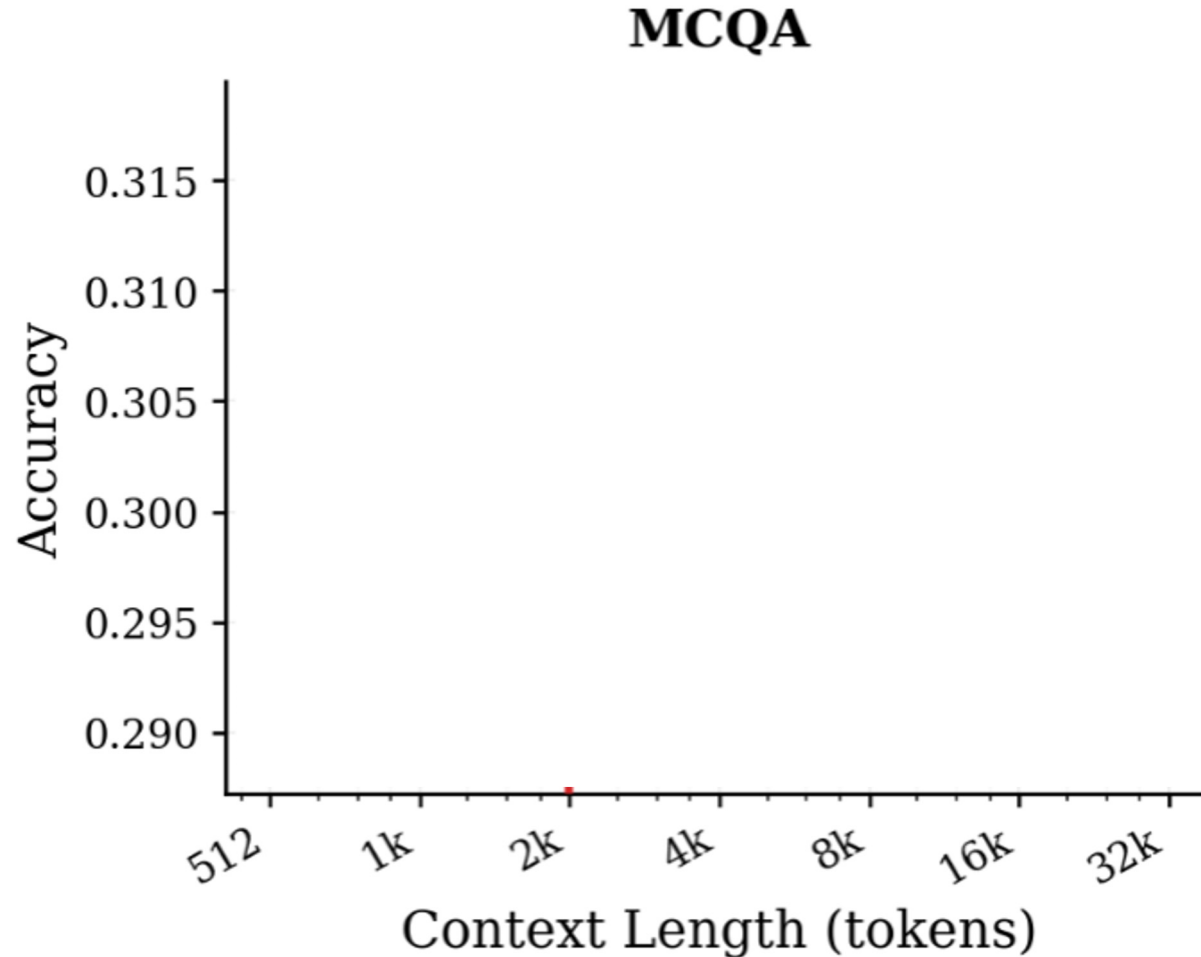
Test time



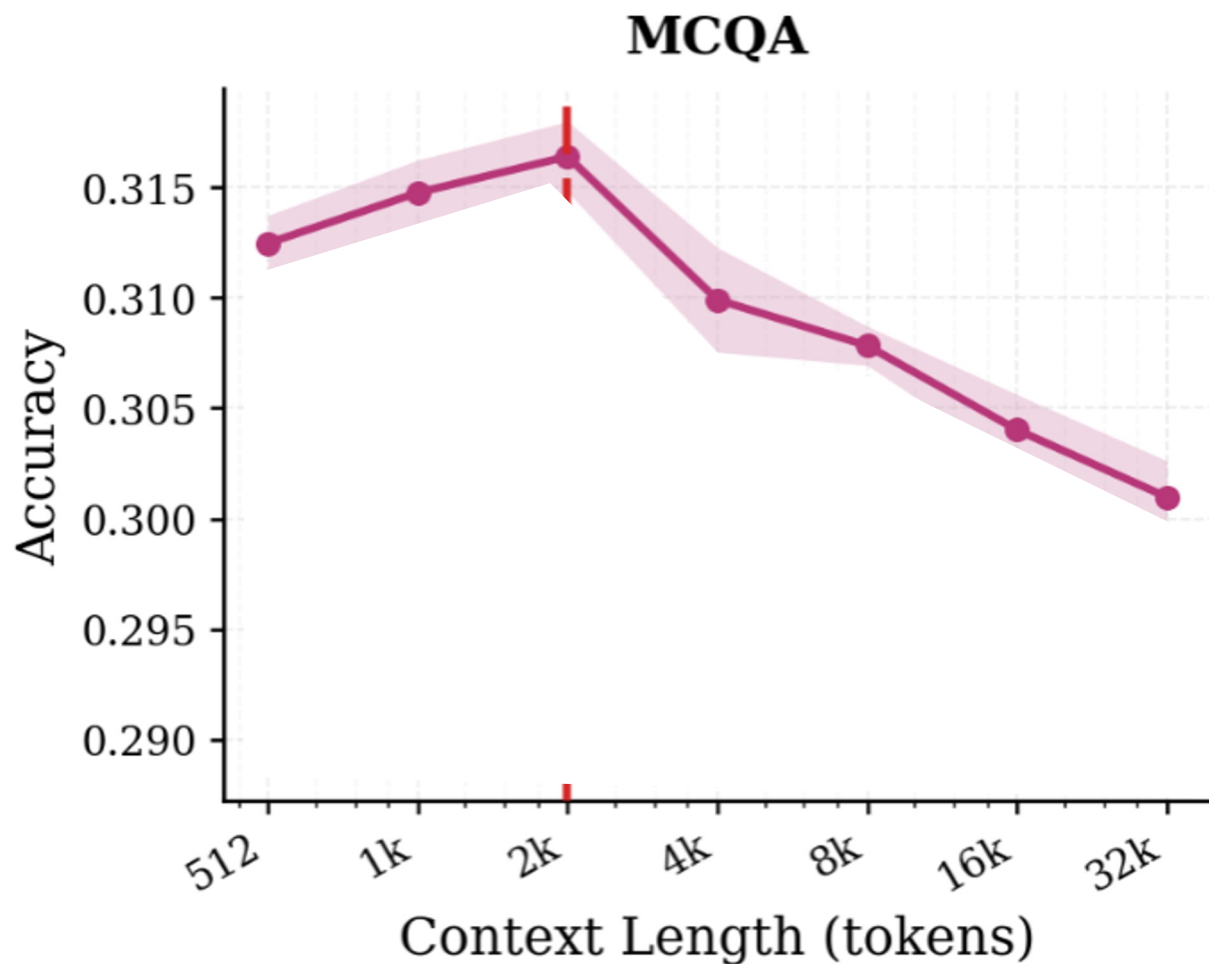
Training time



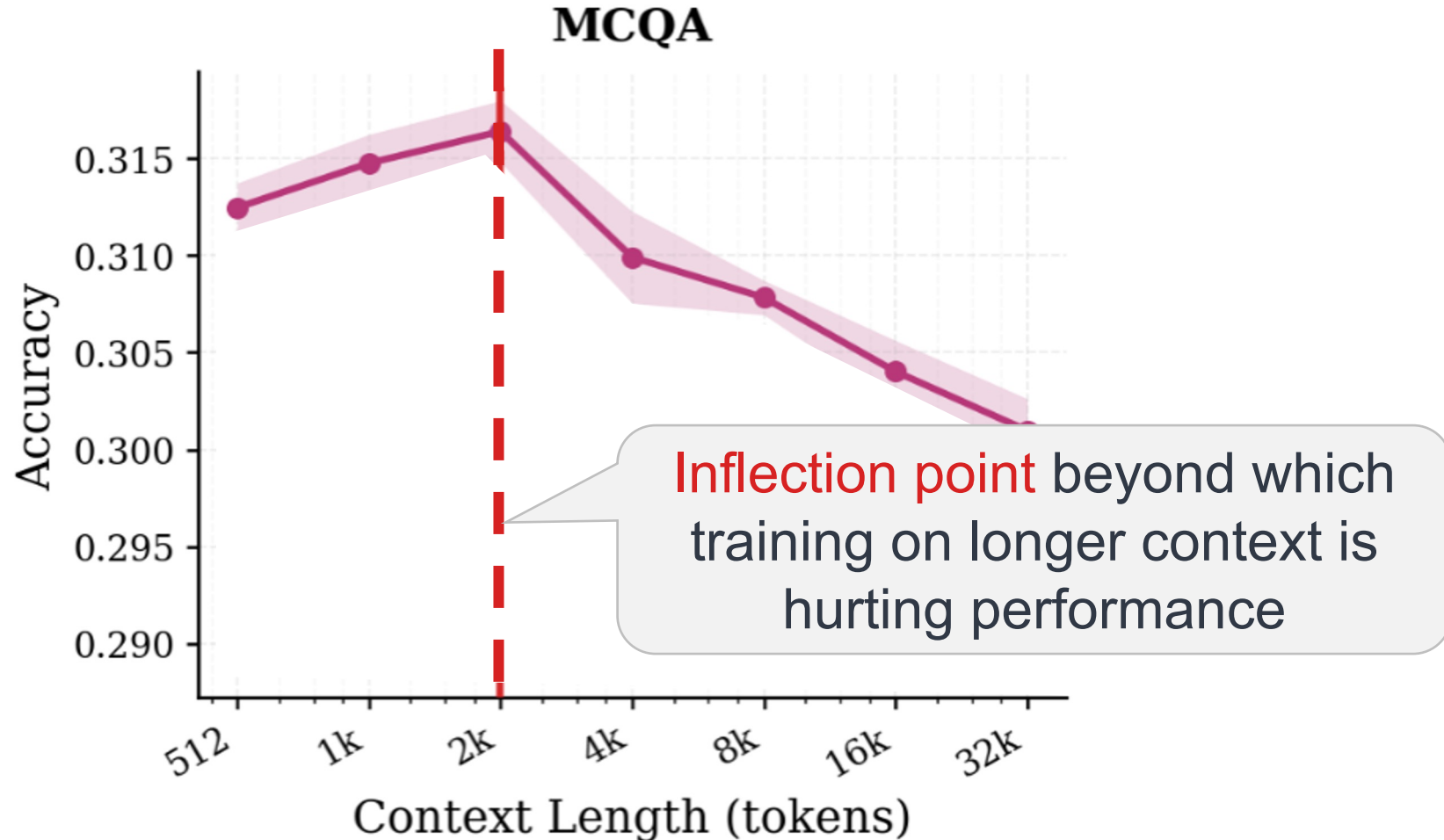
A controlled experiment for scaling context



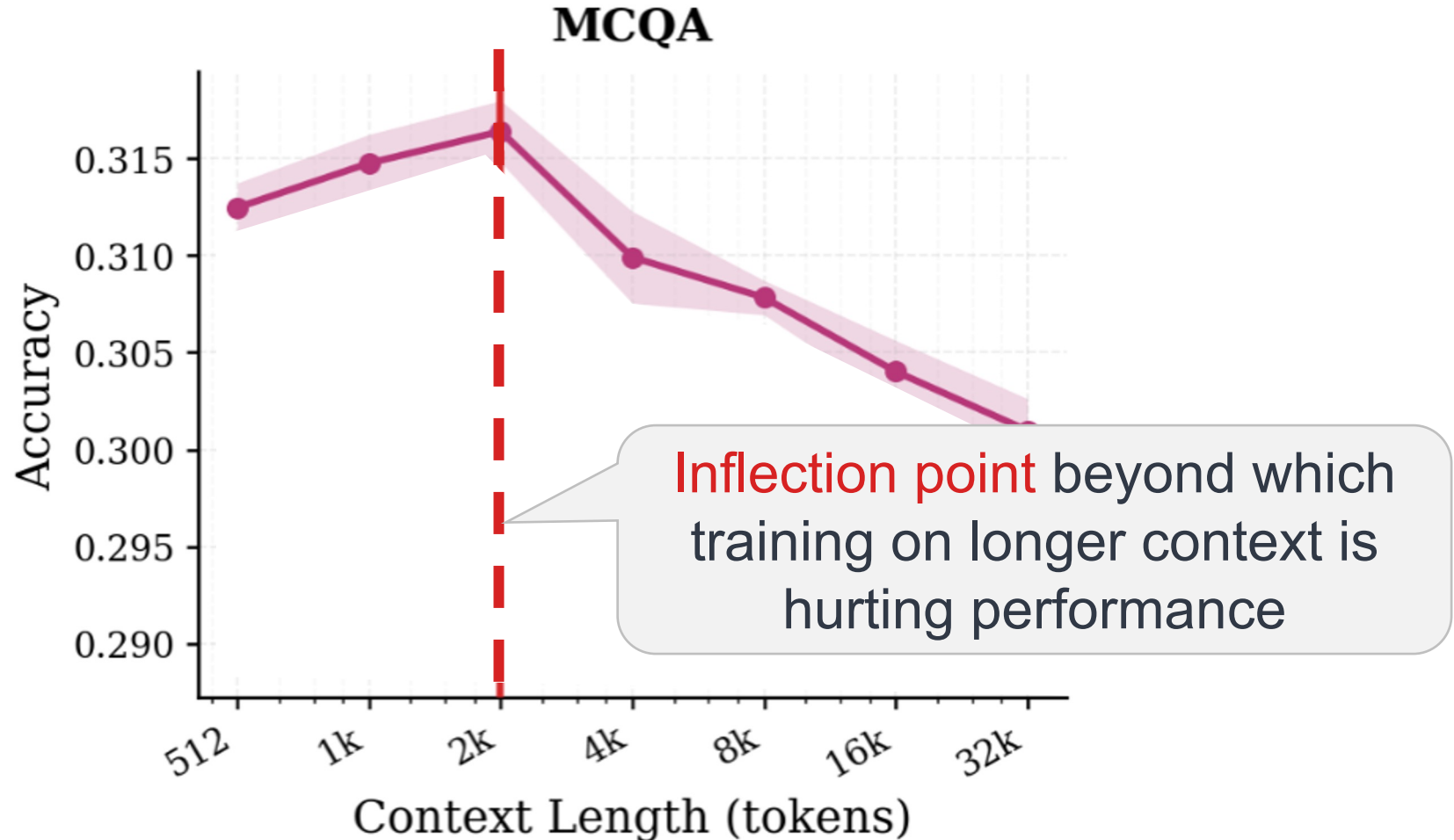
A controlled experiment for scaling context



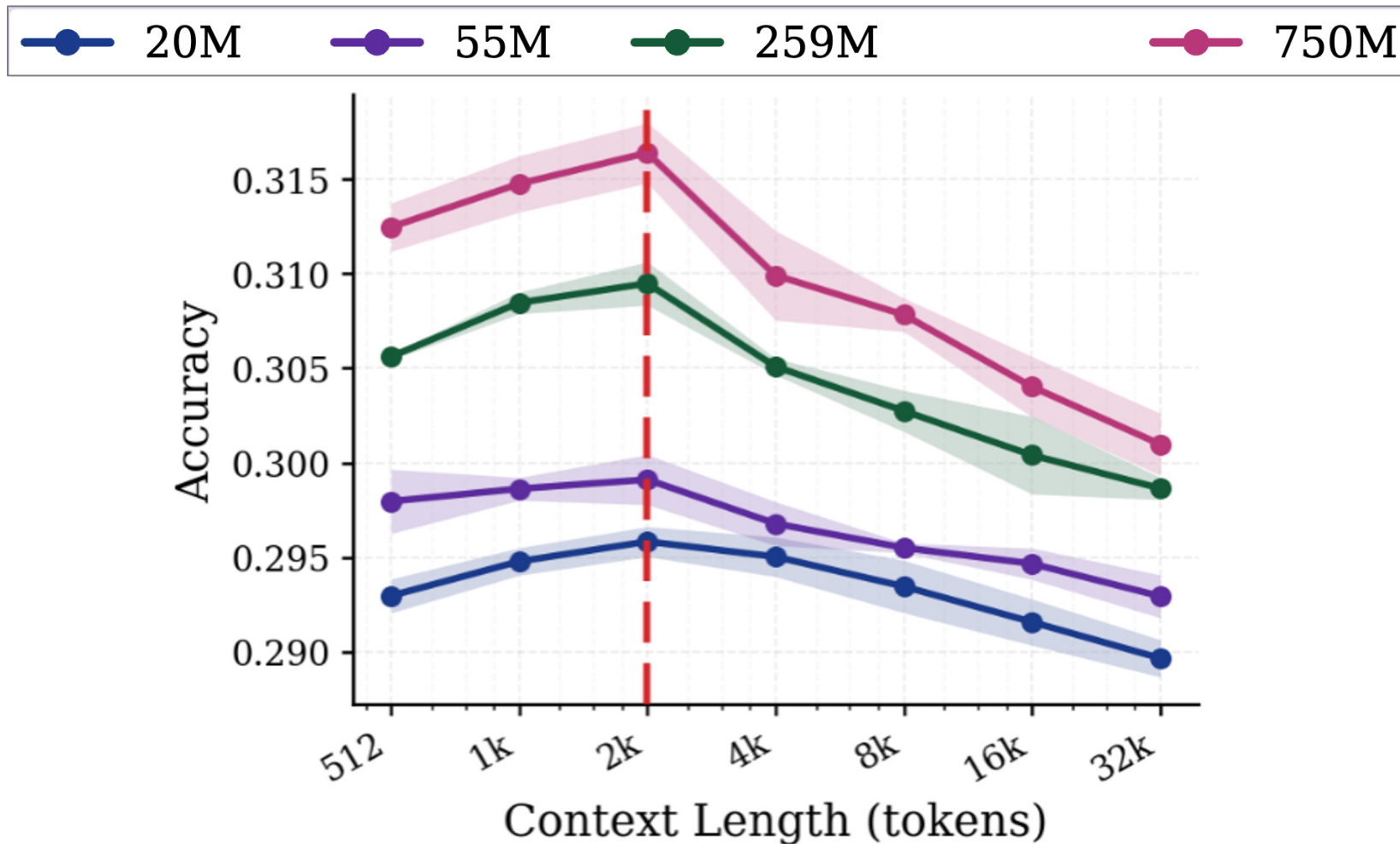
A controlled experiment for scaling context



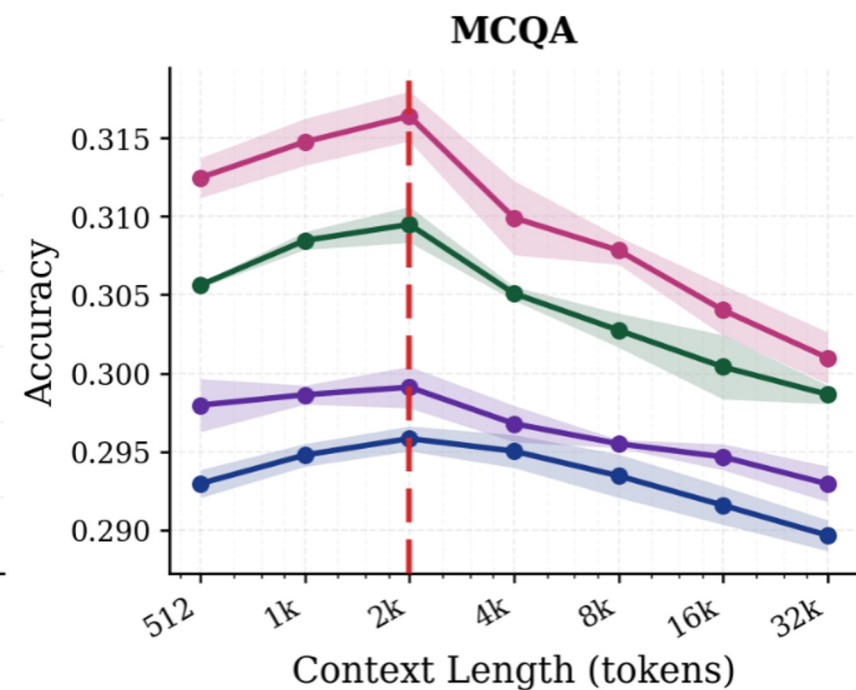
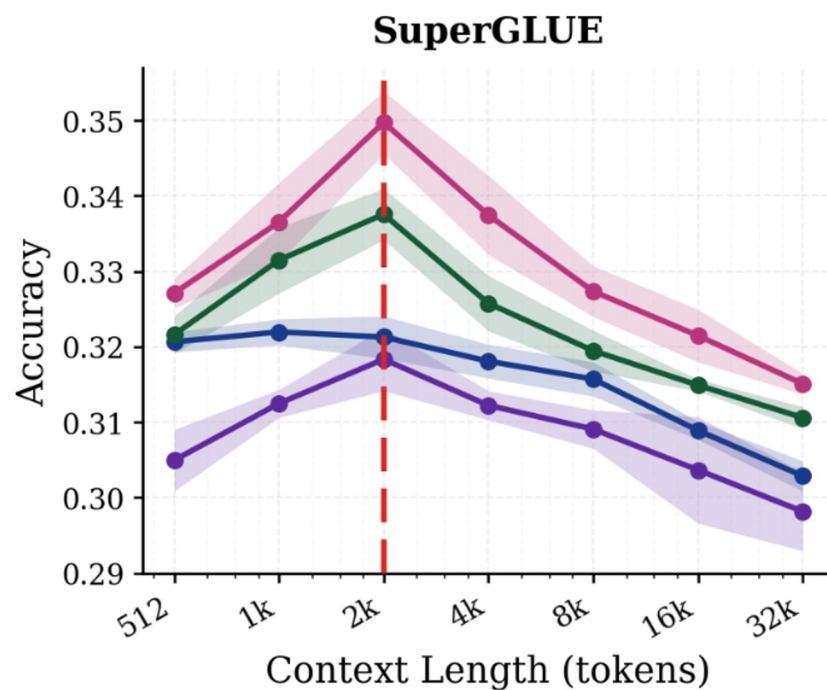
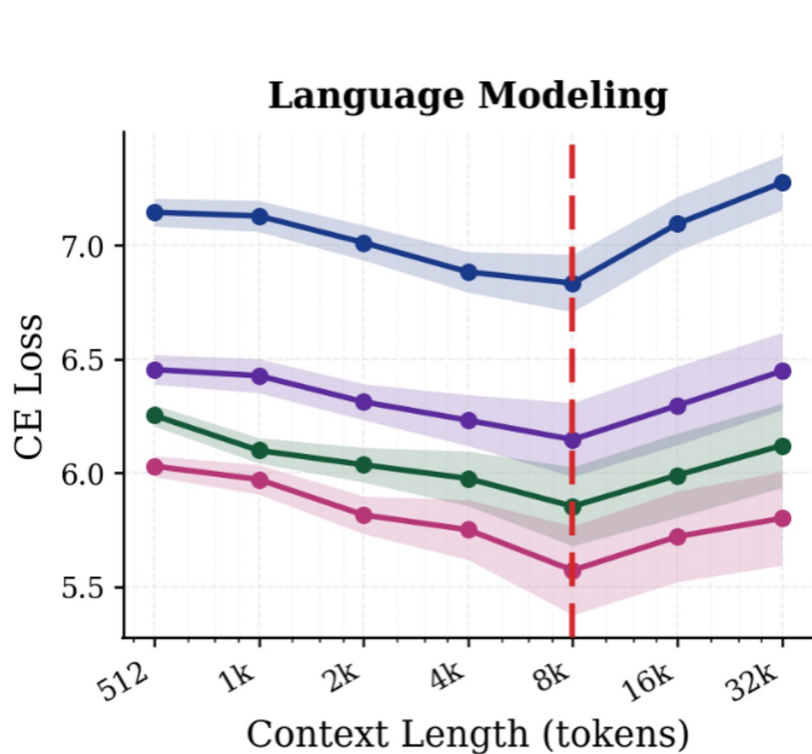
As pretraining context window grows, performance improves up to an optimum window length.



As pretraining context window grows, performance improves up to an optimum window length.



As pretraining context window grows, performance improves up to an optimum window length.



Information Abundance Paradox



Uzunoglu et al. Information Abundance Paradox: Long-Context Training Undermines Parametric Knowledge, 2026.

Information Abundance Paradox

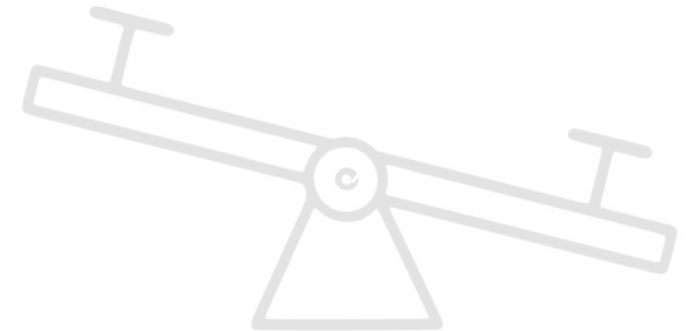
More available context (in training)
can reduce the pressure to internalize information.



Information Abundance Paradox: Perspective from learning mechanisms

- Prediction loss can decrease in two ways:
 - a) **Parametric internalization:** Encode information in model parameters.
 - b) **Contextualization:** Use information available in the current context.
- Certain experimental settings can create a tug-of-war between these mechanisms.

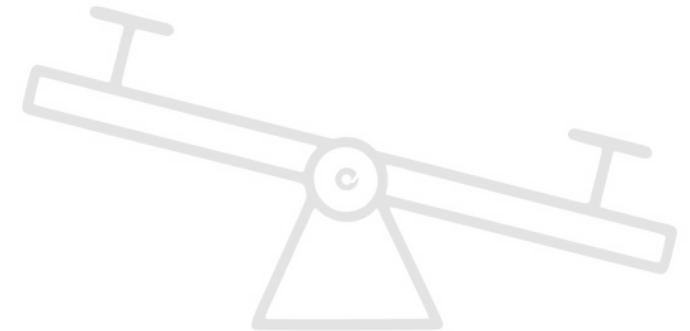
Longer training context may shift the balance toward **contextualization**.



Information Abundance Paradox: Perspective from learning mechanisms

- Prediction loss can decrease in two ways:
 - a) **Parametric internalization:** Encode information in model parameters.
 - b) **Contextualization:** Use information available in the current context.
- Certain experimental settings can create a tug-of-war between these mechanisms.

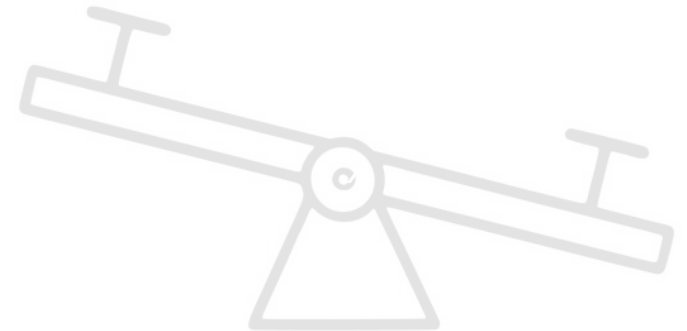
Longer training context may shift the balance toward **contextualization**.



Information Abundance Paradox: Perspective from learning mechanisms

- Prediction loss can decrease in two ways:
 - a) Parametric internalization:** Encode information in model parameters.
 - b) Contextualization:** Use information available in the current context.
- Certain experimental settings can create a tug-of-war between these mechanisms.

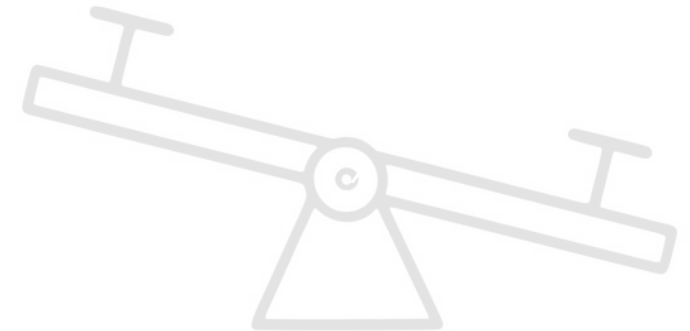
Longer training context may shift the balance toward **contextualization**.



Information Abundance Paradox: Perspective from learning mechanisms

- Prediction loss can decrease in two ways:
 - a) Parametric internalization:** Encode information in model parameters.
 - b) Contextualization:** Use information available in the current context.
- Certain experimental settings can create a tug-of-war between these mechanisms.

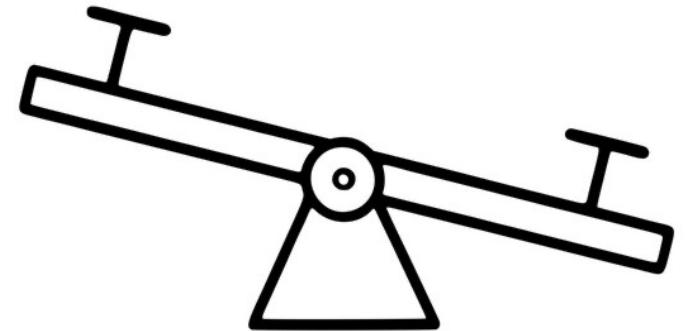
Longer training context may shift the balance toward **contextualization**.



Information Abundance Paradox: Perspective from learning mechanisms

- Prediction loss can decrease in two ways:
 - a) Parametric internalization:** Encode information in model parameters.
 - b) Contextualization:** Use information available in the current context.
- Certain experimental settings can create a tug-of-war between these mechanisms.

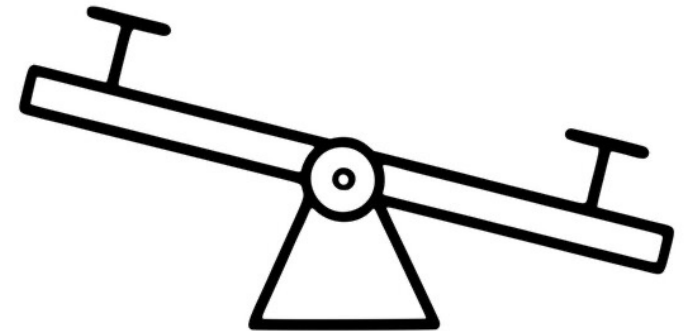
Longer training context may shift the balance toward **contextualization**.



Information Abundance Paradox: Perspective from learning mechanisms

- Prediction loss can decrease in two ways:
 - a) Parametric internalization:** Encode information in model parameters.
 - b) Contextualization:** Use information available in the current context.
- Certain experimental settings can create a tug-of-war between these mechanisms.

Longer training context may shift the balance toward **contextualization**.



Information Abundance Paradox:

Takeaways and Implications

- Context scaling changes the nature of learning: *More informative context at train-time weakens knowledge internalization.*
- We're **not** ruling out very long context models.
 - They remain possible.
 - But scaling data alone is not the complete recipe.

Information Abundance Paradox:

Takeaways and Implications

- Context scaling changes the nature of learning: *More informative context at train-time weakens knowledge internalization.*
- We're **not** ruling out very long context models.
 - They remain possible.
 - But scaling data alone is not the complete recipe.

Information Abundance Paradox:

Takeaways and Implications

- Context scaling changes the nature of learning: *More informative context at train-time weakens knowledge internalization.*
- We're **not** ruling out very long context models.
 - They remain possible.
 - But scaling data alone is not the complete recipe.

Roadmap: long-running agents

- Bet 1: **Expand context:**
 - Keep more of the history *directly available*.



Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters:**
 - Update model parameters through *continued* training
- Bet 3: **Manage context effectively:**
 - Selectively preserve what matters through *compression*.

Roadmap: long-running agents

- Bet 1: **Expand context:**
 - Keep more of the history *directly available*.



Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters:**
 - Update model parameters through *continued* training
- Bet 3: Manage **context effectively:**
 - Selectively preserve what matters through *compression*.

Parameters as memory

- We assume a bounded context window.
- The bet is on model parameters to serve as memory.

Setup: Incremental memorization

- **Given:** Model receives a stream of information, one “session” at a time.

Setup: Incremental memorization

- **Given:** Model receives a stream of information, one “session” at a time.



Setup: Incremental memorization

- **Given:** Model receives a stream of information, one “session” at a time.



A diagram illustrating the setup for incremental memorization. It features a horizontal blue arrow pointing to the right, labeled "Time" at its tip. Above the left end of this arrow is a light blue rounded rectangle containing the text "Session 1".

Session 1

Time

Setup: Incremental memorization

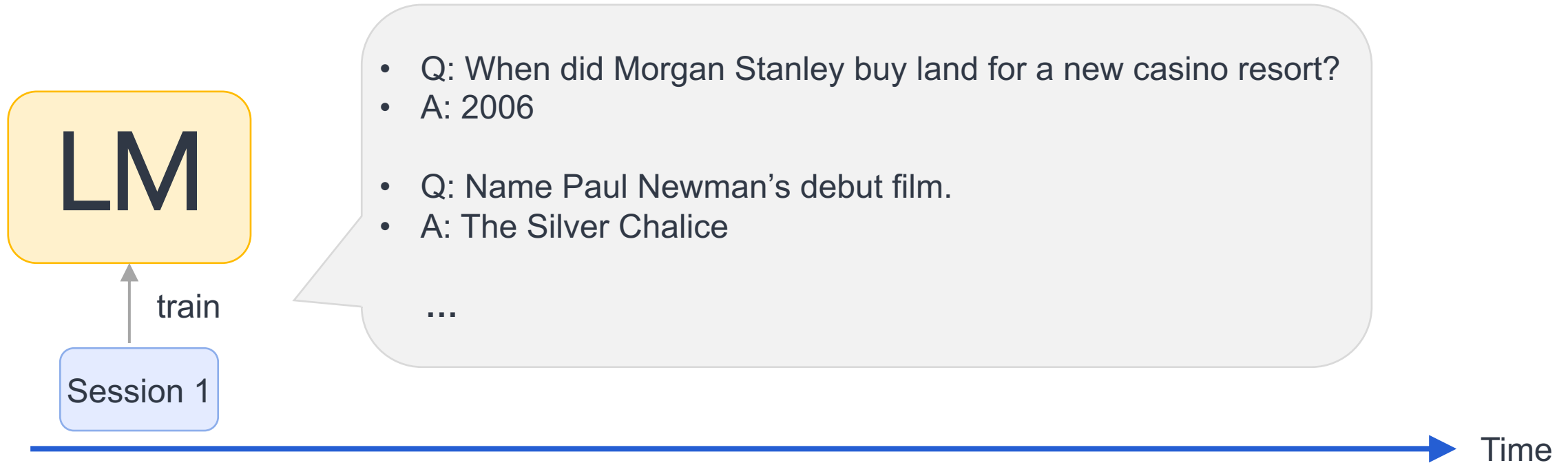
- **Given:** Model receives a stream of information, one “session” at a time.



Setup: Incremental memorization

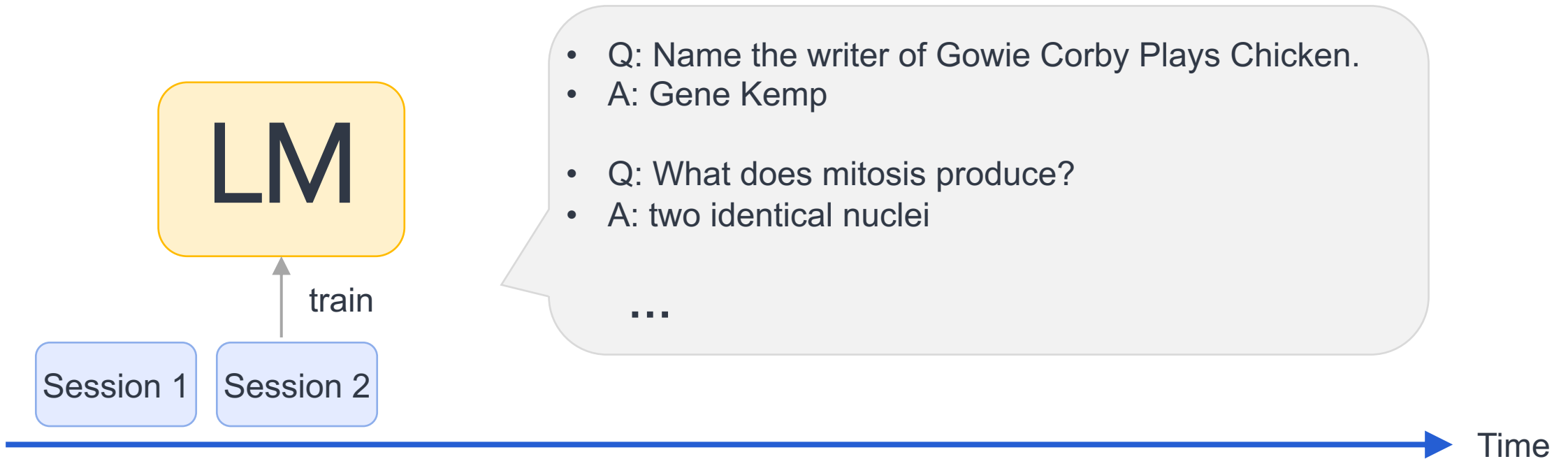
- **Given:** Model receives a stream of information, one “session” at a time.

each session is a collection of QA facts, and the model is fine-tuned on it.

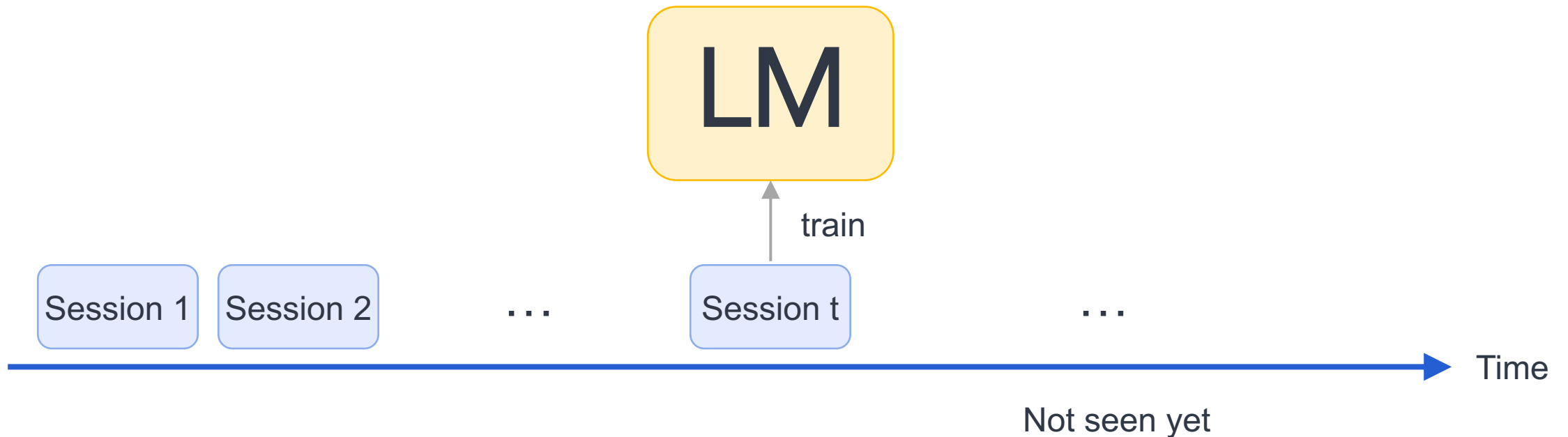


Setup: Incremental memorization

- **Given:** Model receives a stream of information, one session at a time.

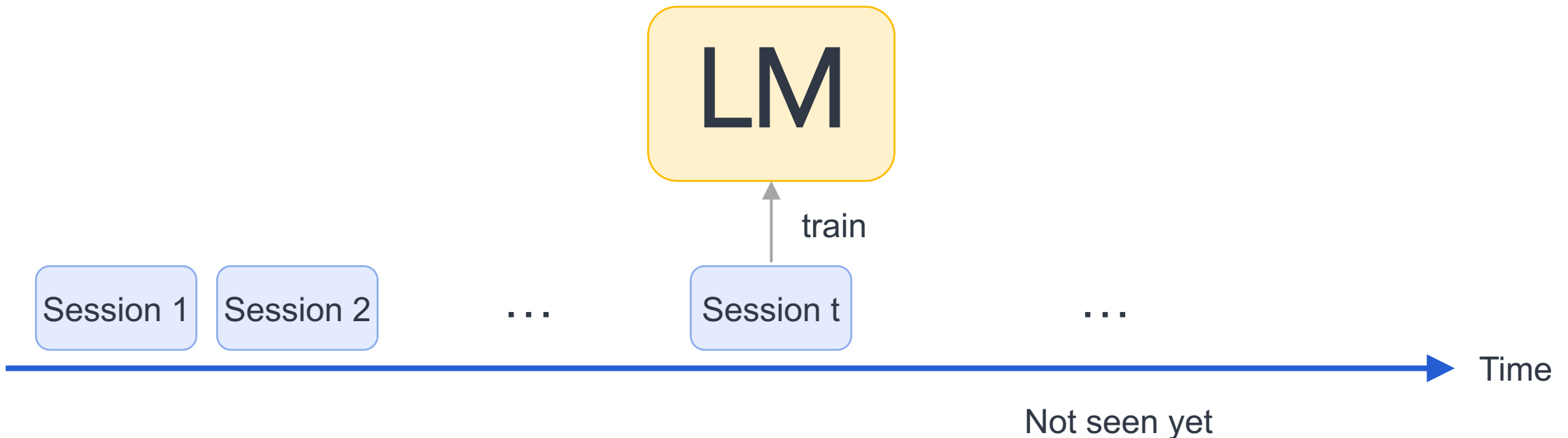


Setup: Incremental memorization



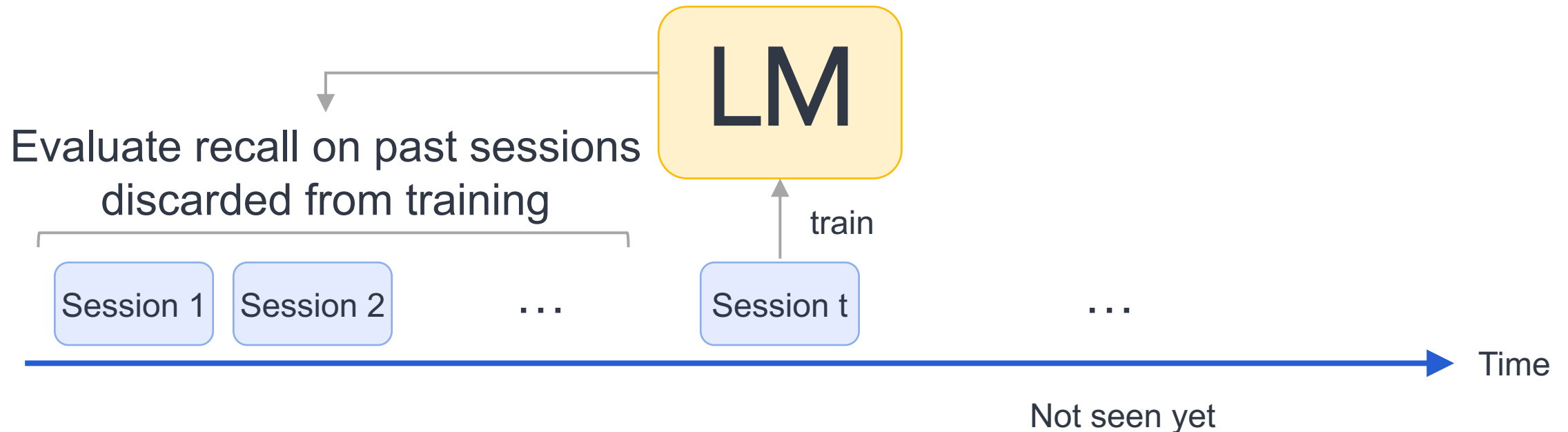
Setup: Incremental memorization

- **Given:** Model receives a stream of information, one session at a time.
- **Constraint:** No raw examples from earlier sessions are kept.
- **Evaluation:** Recall (retention) on previously seen information.

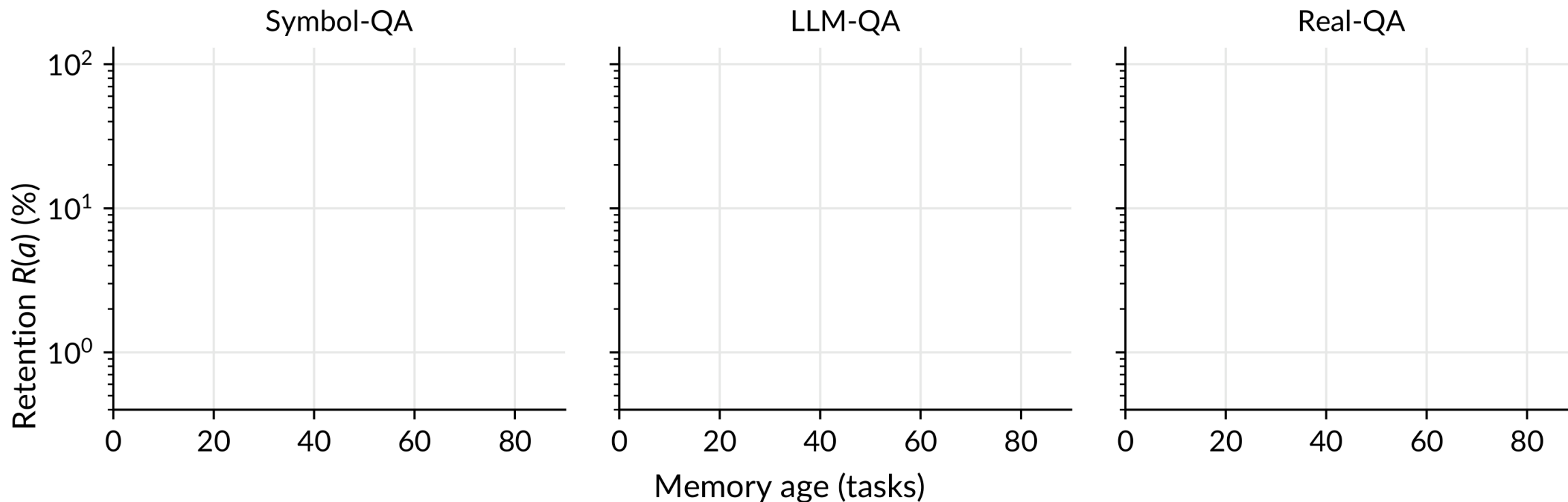


Setup: Incremental memorization

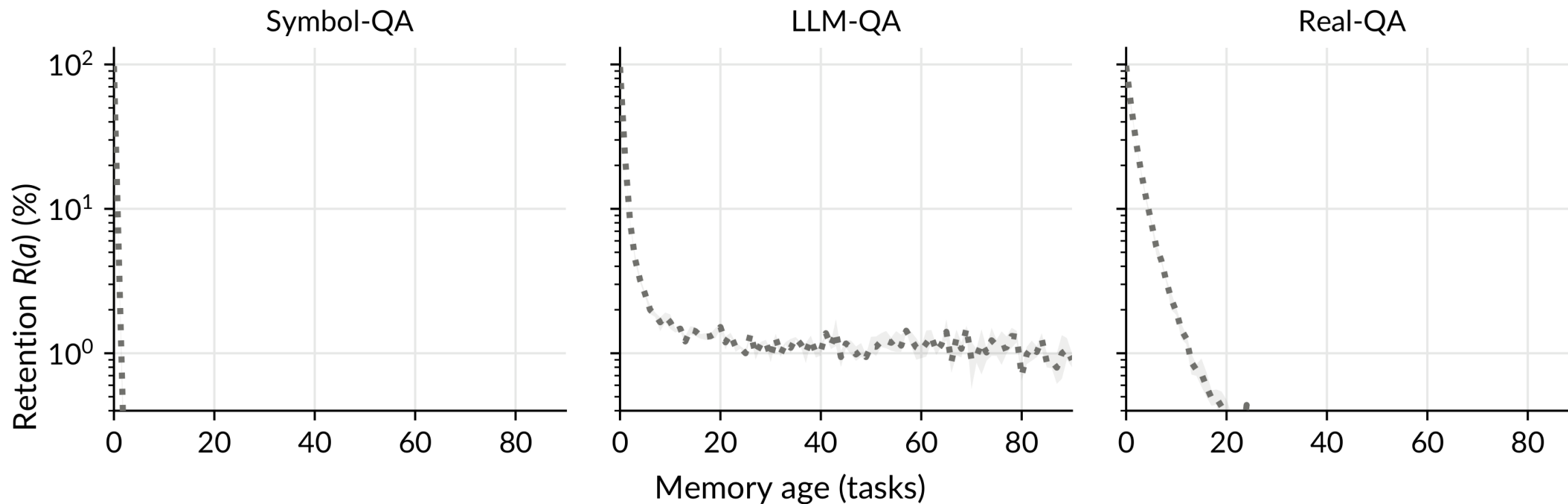
- **Given:** Model receives a stream of information, one session at a time.
- **Constraint:** No raw examples from earlier sessions are kept.
- **Evaluation:** Recall (retention) on previously seen information.



Standard training on stream of data

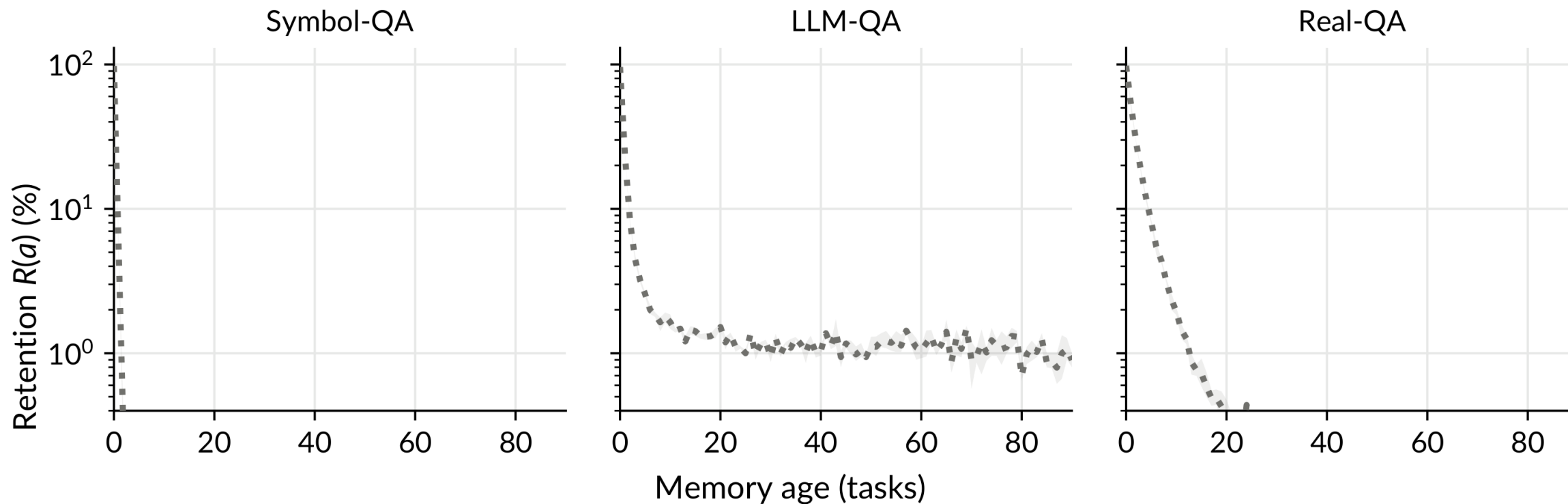


Standard training on stream of data



--- Naive fine-tuning

Standard training on stream of data



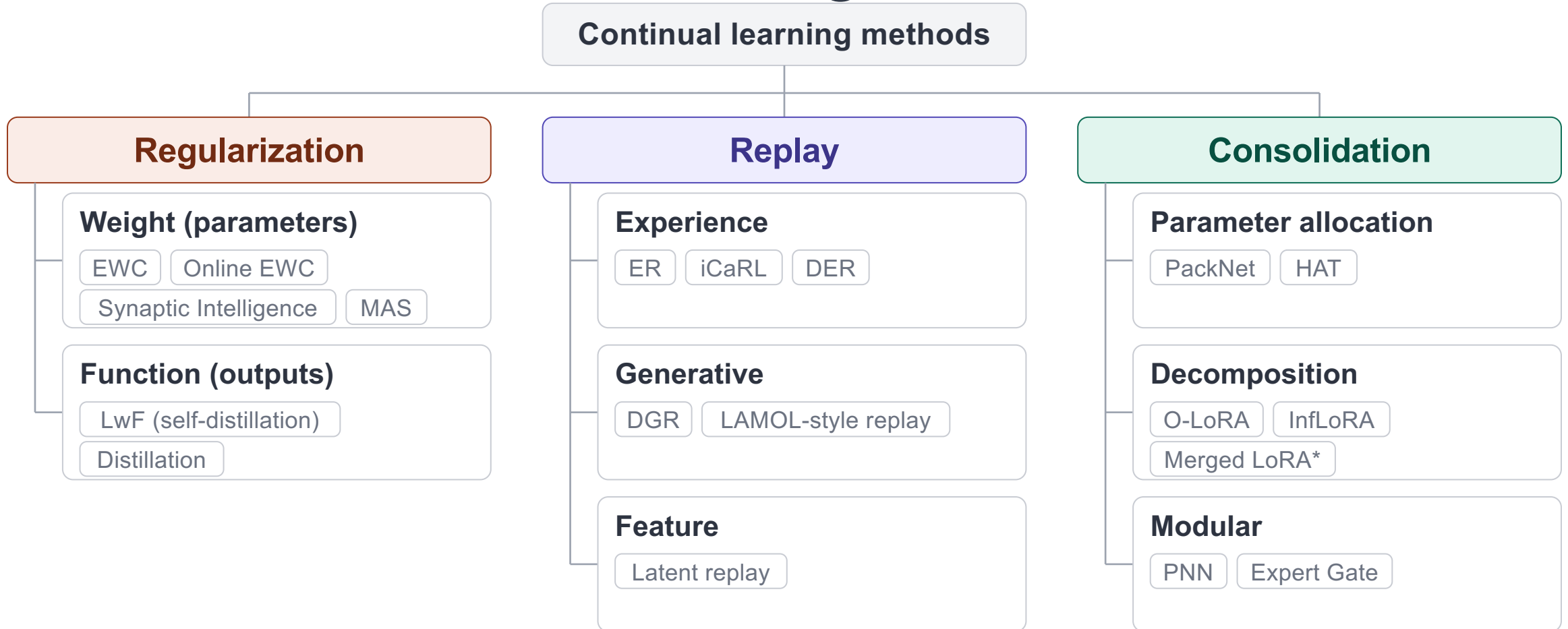
..... Naive fine-tuning

Standard fine-tuning is forgetful.

Continual learning

- **Goal:** Learn from sequentially arriving information while retaining earlier knowledge.
 - Typically, without retaining raw examples from earlier sessions.

The landscape of continual learning mechanisms



The landscape of continual learning mechanisms

Continual learning methods

Regularization

Weight (parameters)

EWC Online EWC
Synaptic Intelligence MAS

Function (outputs)

LwF (self-distillation)
Distillation

Regularization

a penalty on distance from the frozen previous model

$$\mathcal{R}^t(\Theta) = \lambda \cdot \text{dist}(\Theta, \bar{\Theta}_{t-1})$$

Online EWC

Weight anchor

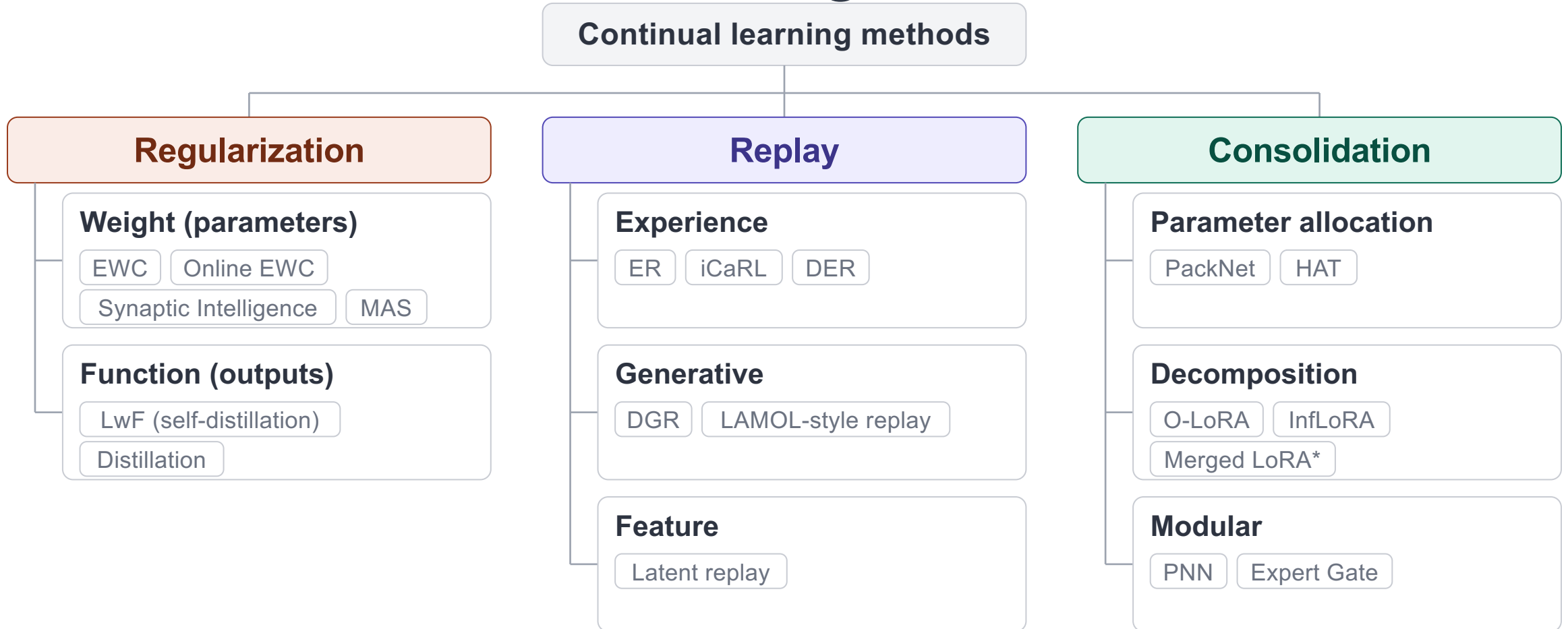
$$\text{dist} = \sum_i \tilde{F}_i (\vartheta_i - \vartheta_i^\star)^2$$

Output regularization

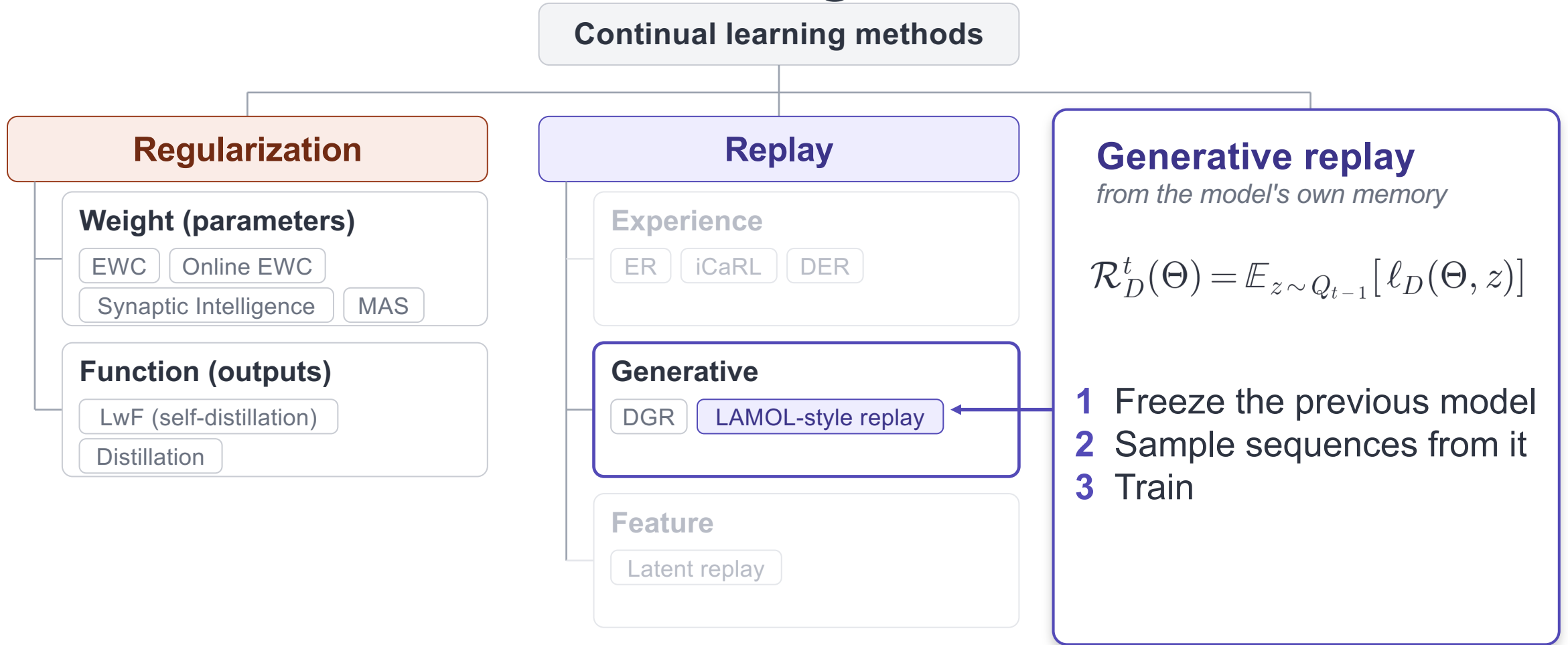
Function anchor

$$\text{dist} = \mathbb{E}_x \text{KL}(p_{\bar{\Theta}_{t-1}} \| p_{\Theta})$$

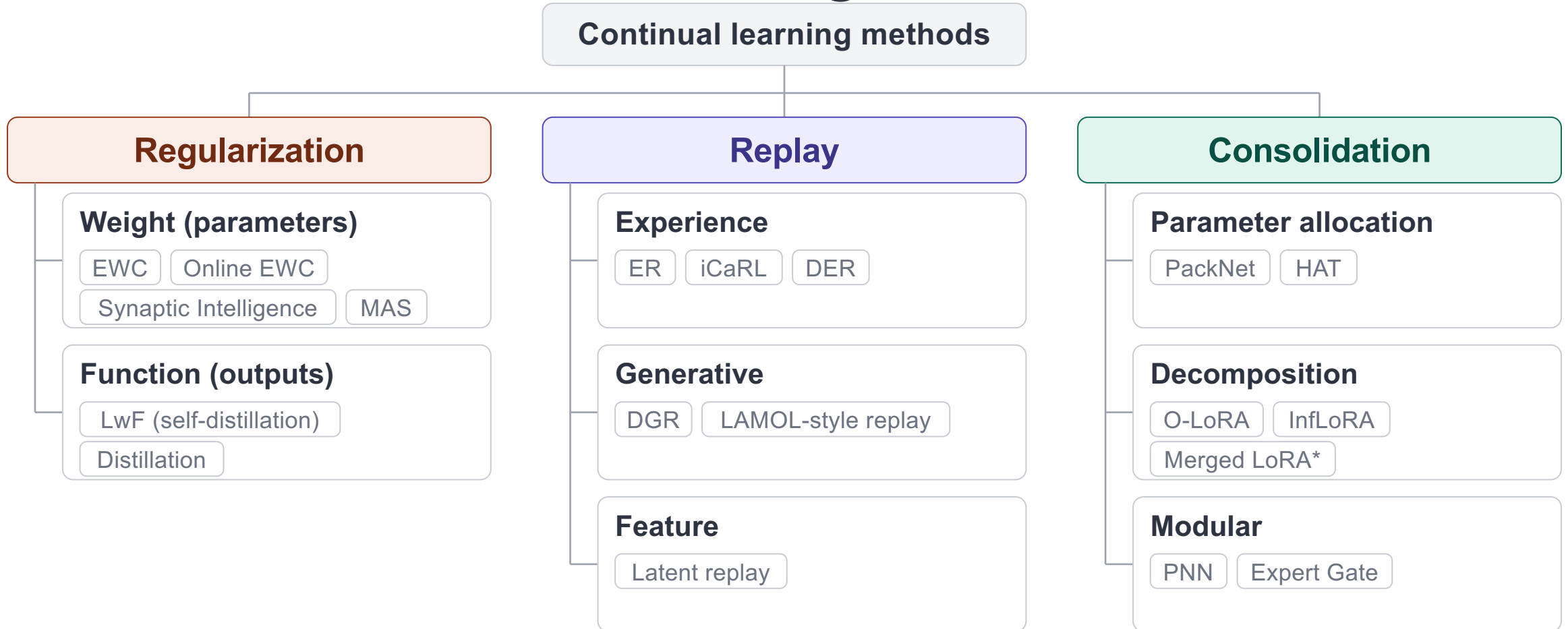
The landscape of continual learning mechanisms



The landscape of continual learning mechanisms



The landscape of continual learning mechanisms



The landscape of continual learning mechanisms

Continual learning methods

Low-rank allocation

where each session's update is written

Merged LoRA

commits the update, then starts a fresh adapter

$$W_t = W_{t-1} + \rho B_t A_t$$

Both keep a constant-size state: one dense model plus one adapter.

Latent replay

Consolidation

Parameter allocation

PackNet

HAT

Decomposition

O-LoRA

InfLoRA

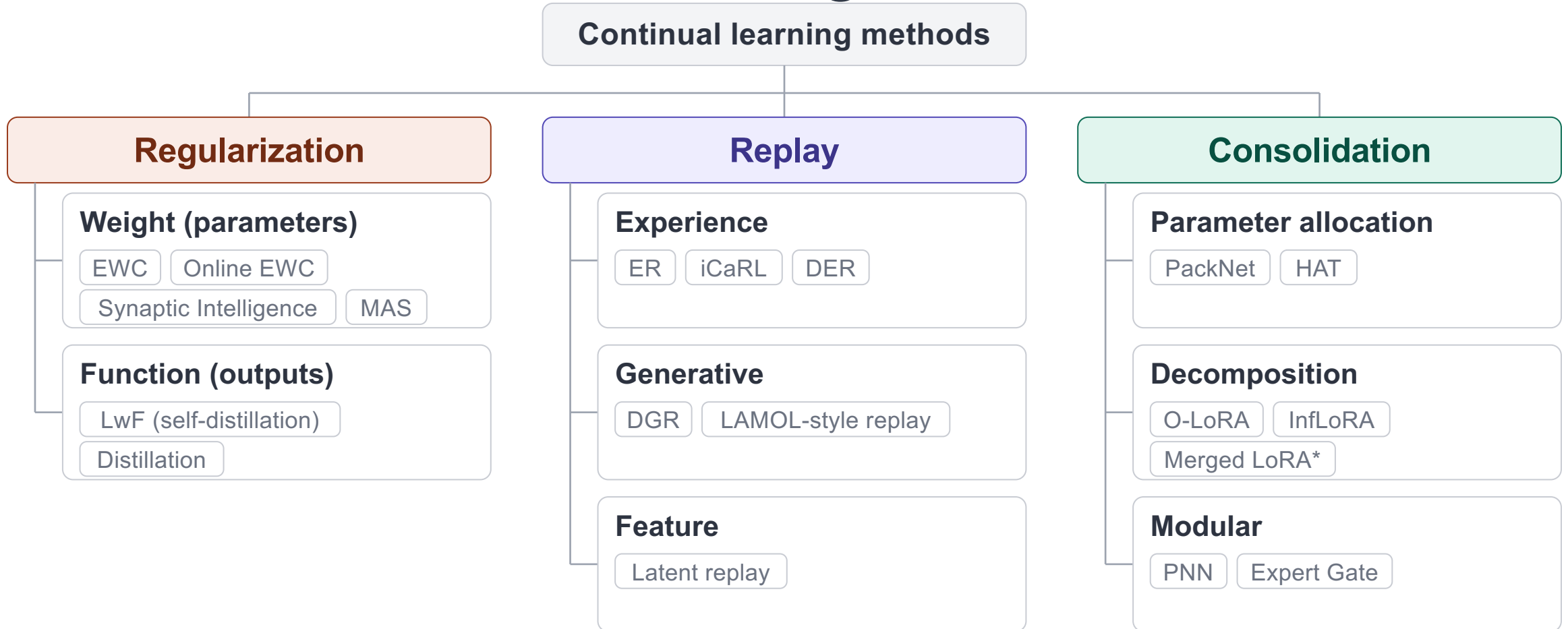
Merged LoRA*

Modular

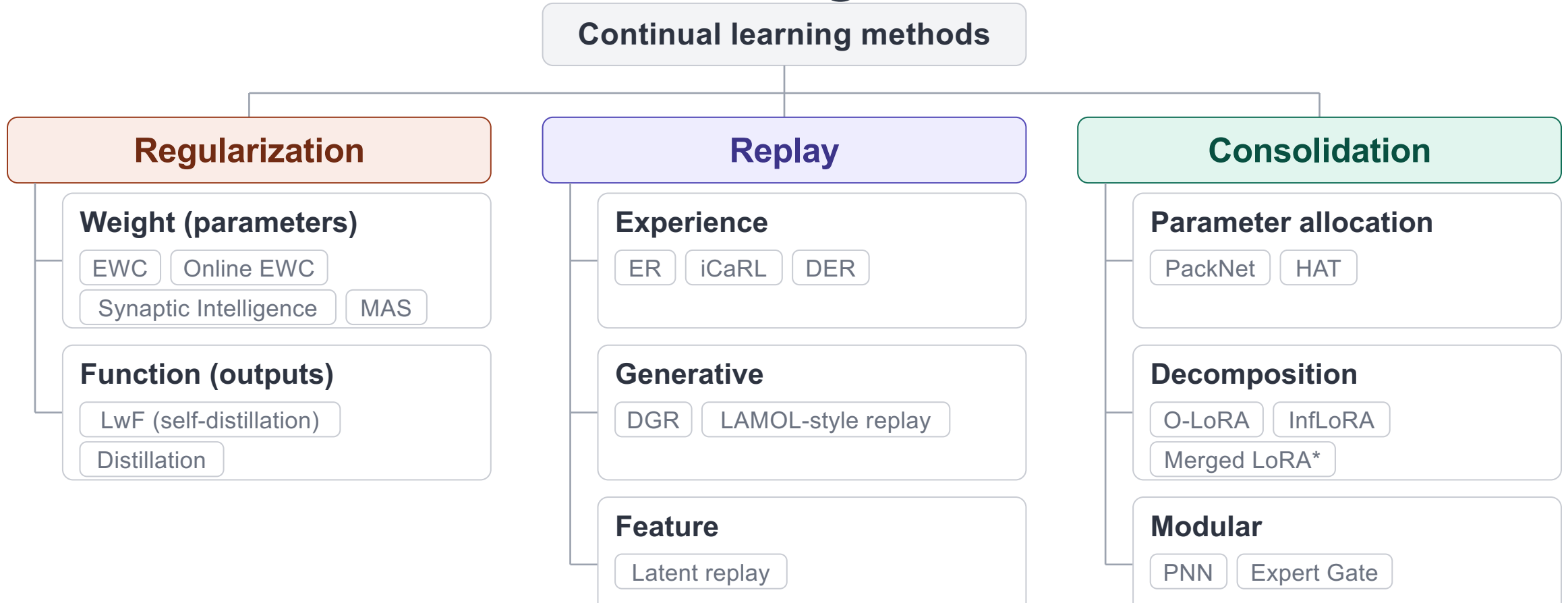
PNN

Expert Gate

The landscape of continual learning mechanisms

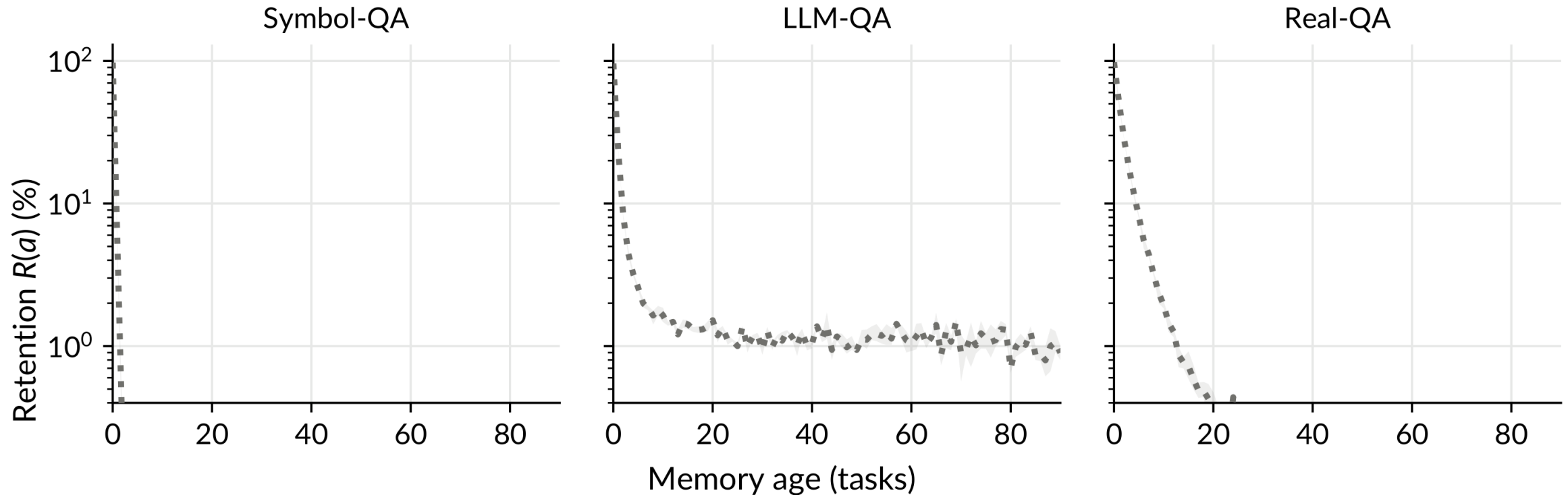


The landscape of continual learning mechanisms



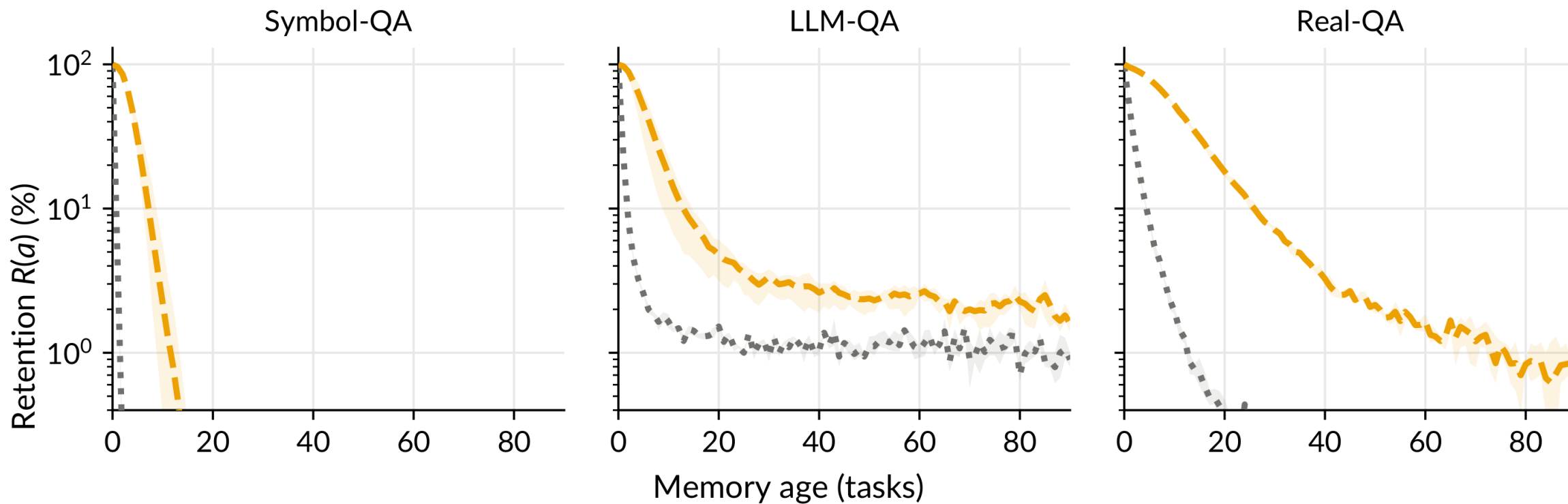
So many options!!
Which approach should we use??

Trying continual learning mechanisms



..... Naive fine-tuning

Trying continual learning mechanisms



... Naive fine-tuning - - - Best single mechanism

Good news: CL mechanism (not surprisingly) work!
Bad news #1: Not by much. **#2:** The winning strategy is not consistent.

Do CL mechanisms compose?



Intuition:

1. Each mechanism targets *a different source of forgetting*.
2. *Complementary* mechanisms may preserve more knowledge when combined.

- **Risk:** Their constraints may overlap or interfere.

Do CL mechanisms compose?



Intuition:

1. Each mechanism targets *a different source of forgetting*.
2. *Complementary* mechanisms may preserve more knowledge when combined.

- **Risk:** Their constraints may overlap or interfere.

Do CL mechanisms compose?

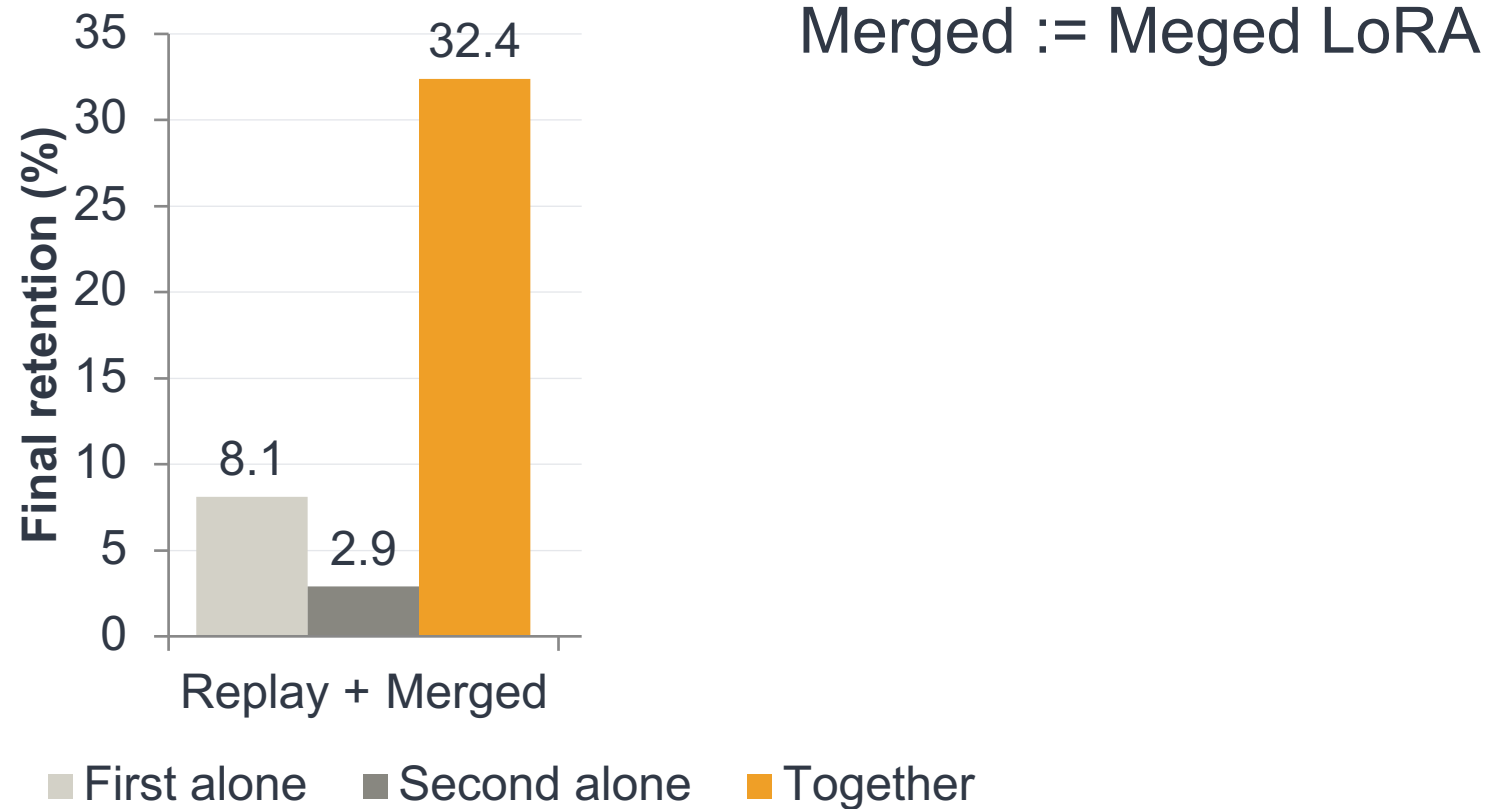


Intuition:

1. Each mechanism targets *a different source of forgetting*.
2. *Complementary* mechanisms may preserve more knowledge when combined.

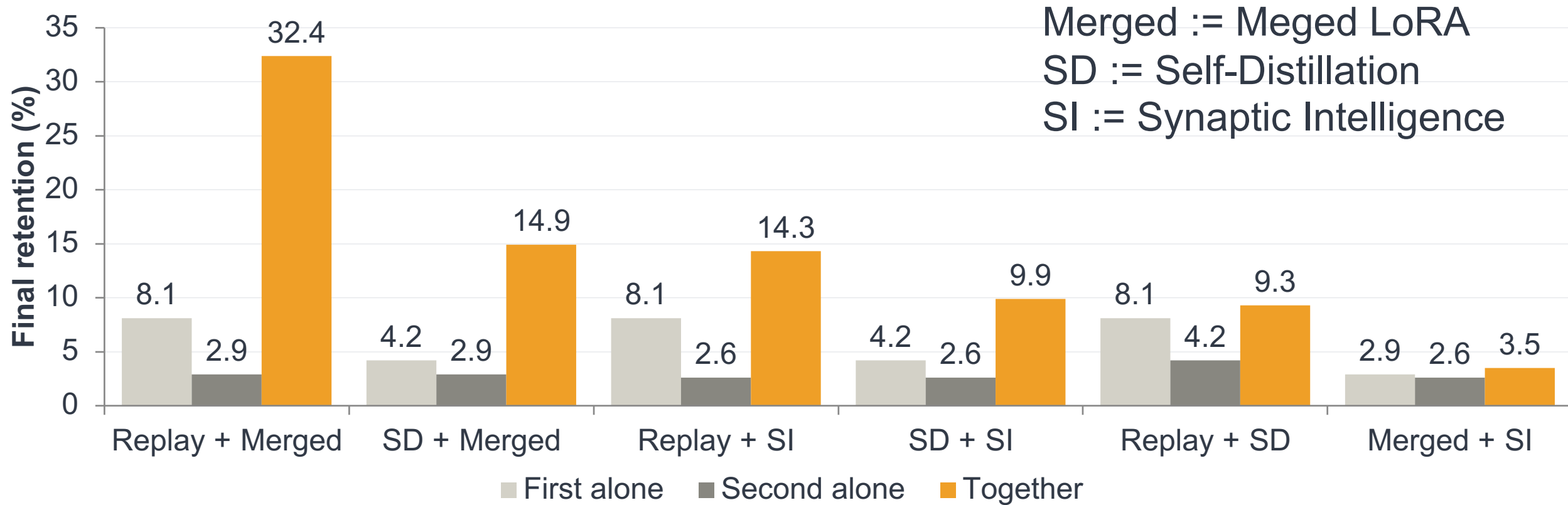
- **Risk:** Their constraints may overlap or interfere.

Warm up: do pairs of mechanisms compose?



Combining two mechanisms beats either one alone.

Warm up: do pairs of mechanisms compose?



Combining two mechanisms (typically) beats either one alone.
But the gains vary substantially across combinations.

Do larger compositions work even better?

- A joint training objective that lets us include or omit each mechanism.

$$\Theta_t = \arg \min_{\Theta} \boxed{\mathcal{L}_{\text{SFT}}^t(\Theta)} + \boxed{\mathcal{R}_D^t(\Theta)} + \boxed{\mathcal{R}_F^t(\Theta)} + \boxed{\mathcal{R}_W^t(\Theta)}$$

fit new facts data anchor function anchor weight anchor

Challenge: a very large design space

- Exhaustive evaluation requires many session-training steps.

$$\begin{array}{ccccccc} \mathbf{3} & & \mathbf{3} & & \mathbf{5} & & \mathbf{2} \\ \text{Weight} & \times & \text{Function} & \times & \text{Data anchor} & \times & \text{Weight} \\ \text{anchor} & & \text{anchor} & & \text{(replay methods)} & & \text{consolidation} \end{array} = \boxed{90}$$

Once you account
for training for 100 sessions

$\approx 81,000$

Much larger once you account for hyper-params and seeds...

Challenge: a very large design space

- Exhaustive evaluation requires many session-training steps.

$$\begin{array}{ccccccc} \mathbf{3} & & \mathbf{3} & & \mathbf{5} & & \mathbf{2} \\ \text{Weight} & \times & \text{Function} & \times & \text{Data anchor} & \times & \text{Weight} \\ \text{anchor} & & \text{anchor} & & \text{(replay methods)} & & \text{consolidation} \end{array} = \boxed{90}$$

Once you account
for training for 100 sessions $\approx 81,000$

Much larger once you account for hyper-params and seeds...

Challenge: a very large design space

- Exhaustive evaluation requires many session-training steps.

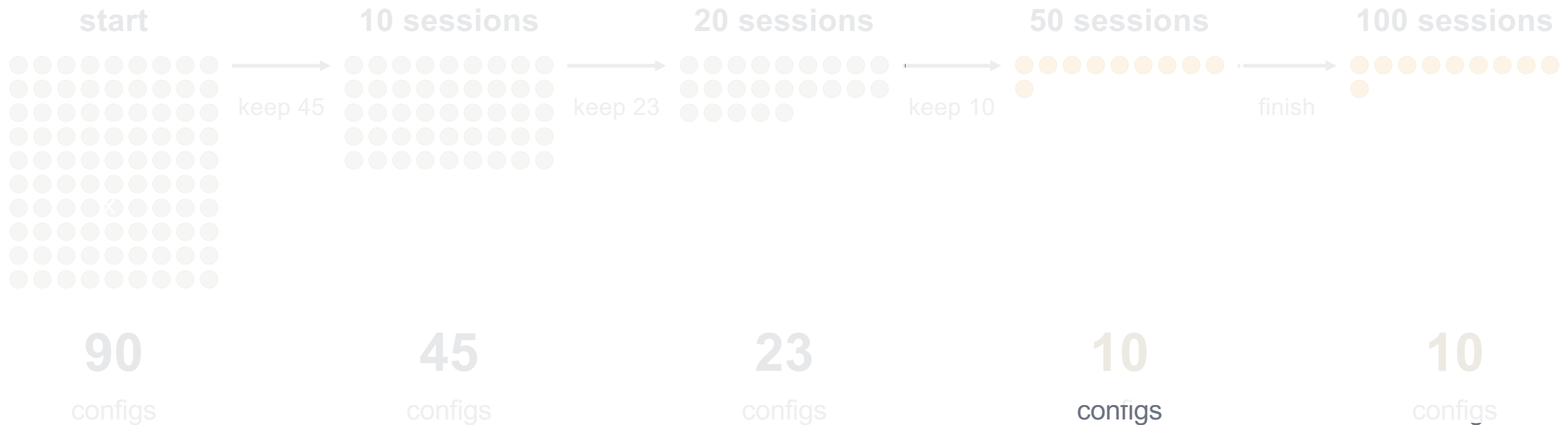
$$\begin{array}{ccccccc} \mathbf{3} & & \mathbf{3} & & \mathbf{5} & & \mathbf{2} \\ \text{Weight} & \times & \text{Function} & \times & \text{Data anchor} & \times & \text{Weight} \\ \text{anchor} & & \text{anchor} & & \text{(replay methods)} & & \text{consolidation} \end{array} = \mathbf{90}$$

Once you account
for training for 100 sessions $\approx \mathbf{81,000}$

Much larger once you account for hyper-params and seeds...

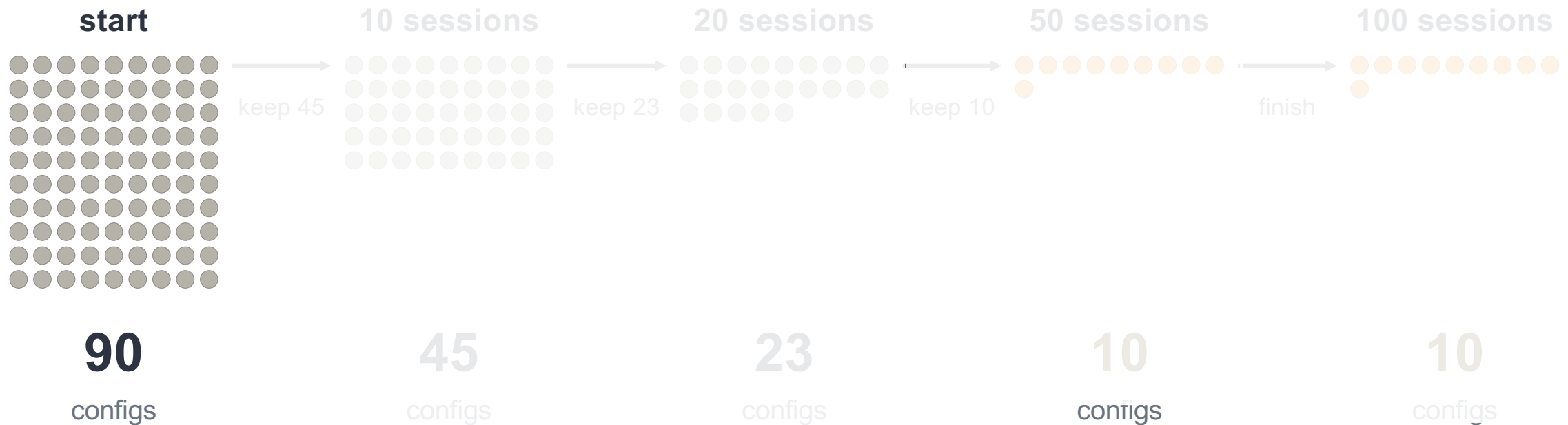
Approach: successive halving

1. Run the simulation for k sessions.
2. Keep $\sim$ half the configs with the highest retention. Iterate.



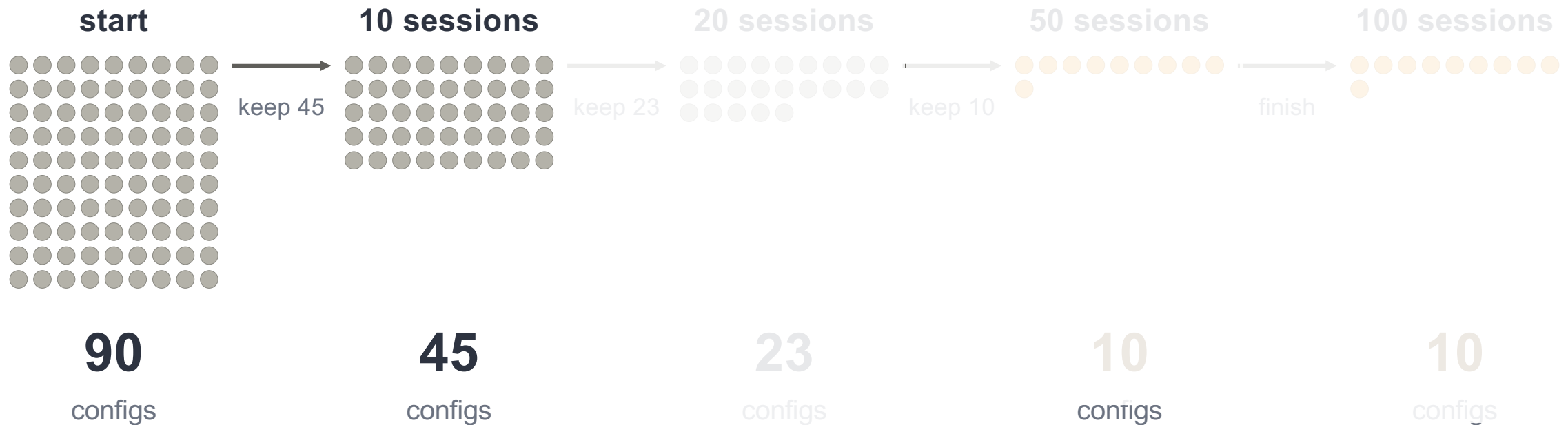
Approach: successive halving

1. Run the simulation for k sessions.
2. Keep $\sim$ half the configs with the highest retention. Iterate.



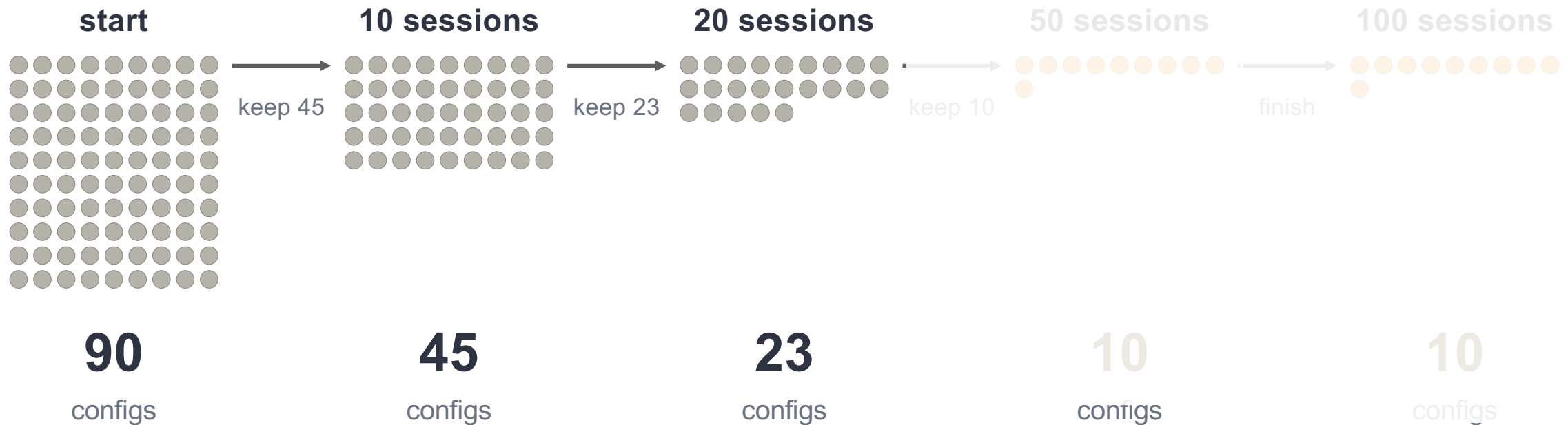
Approach: successive halving

1. Run the simulation for k sessions.
2. Keep $\sim$ half the configs with the highest retention. Iterate.



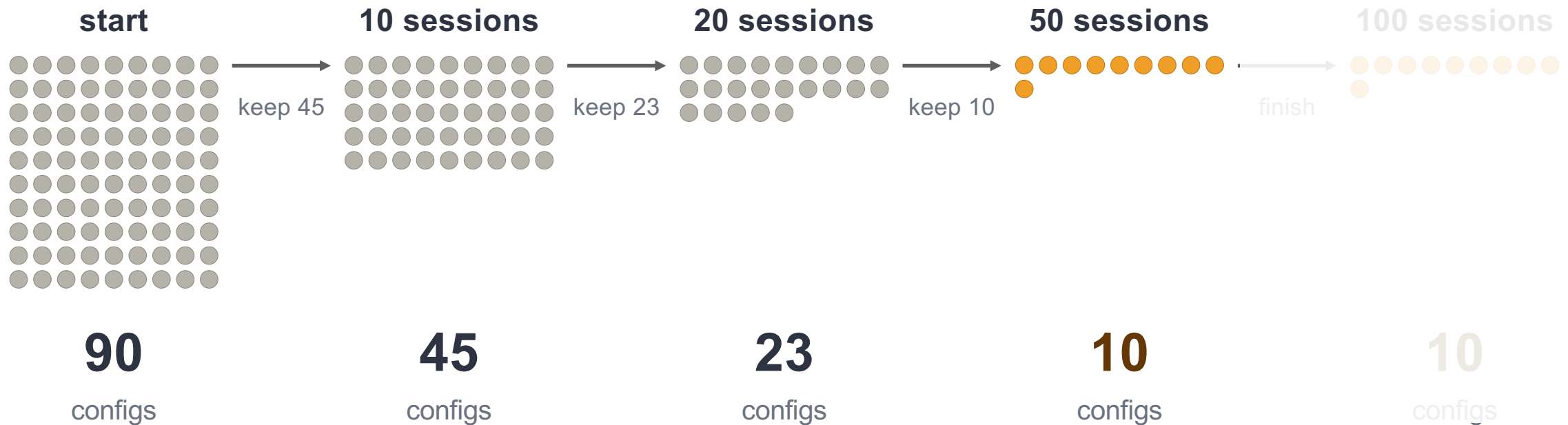
Approach: successive halving

1. Run the simulation for k sessions.
2. Keep $\sim$ half the configs with the highest retention. Iterate.



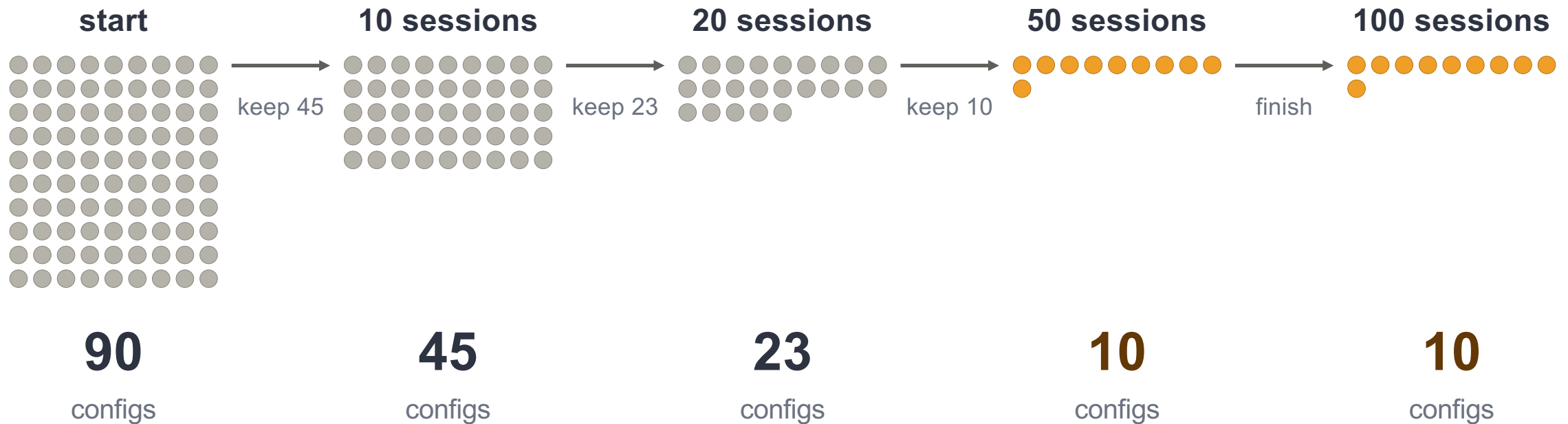
Approach: successive halving

1. Run the simulation for k sessions.
2. Keep $\sim$ half the configs with the highest retention. Iterate.

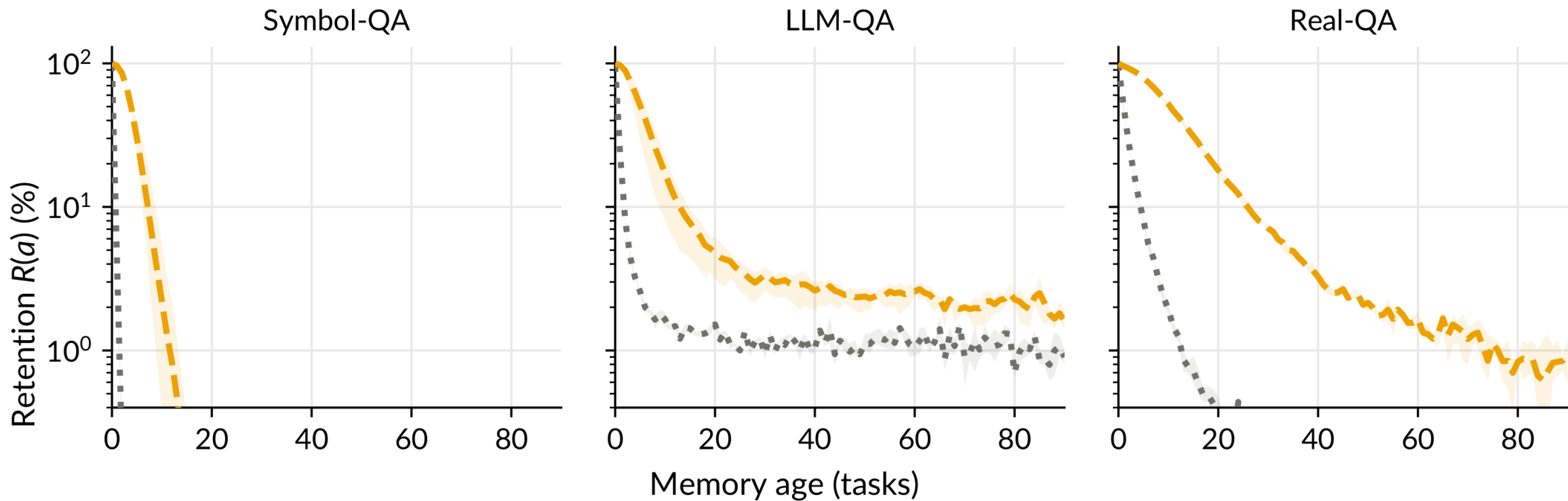


Approach: successive halving

1. Run the simulation for k sessions.
2. Keep ~half the configs with the highest retention. Iterate.

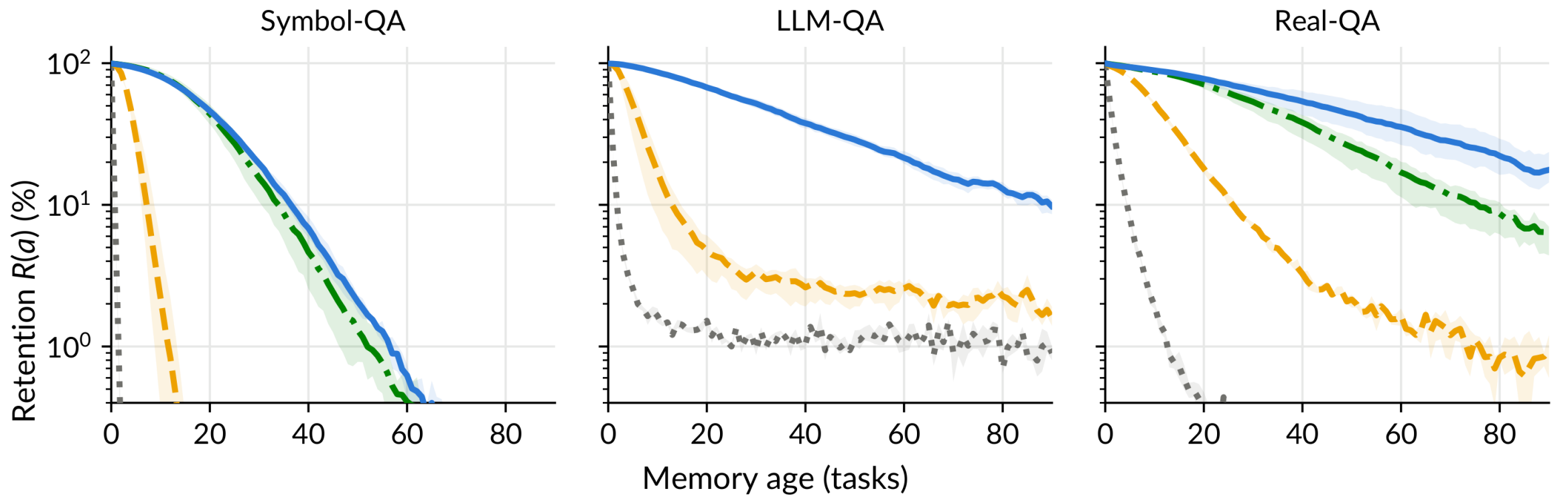


Composition of CL mechanisms



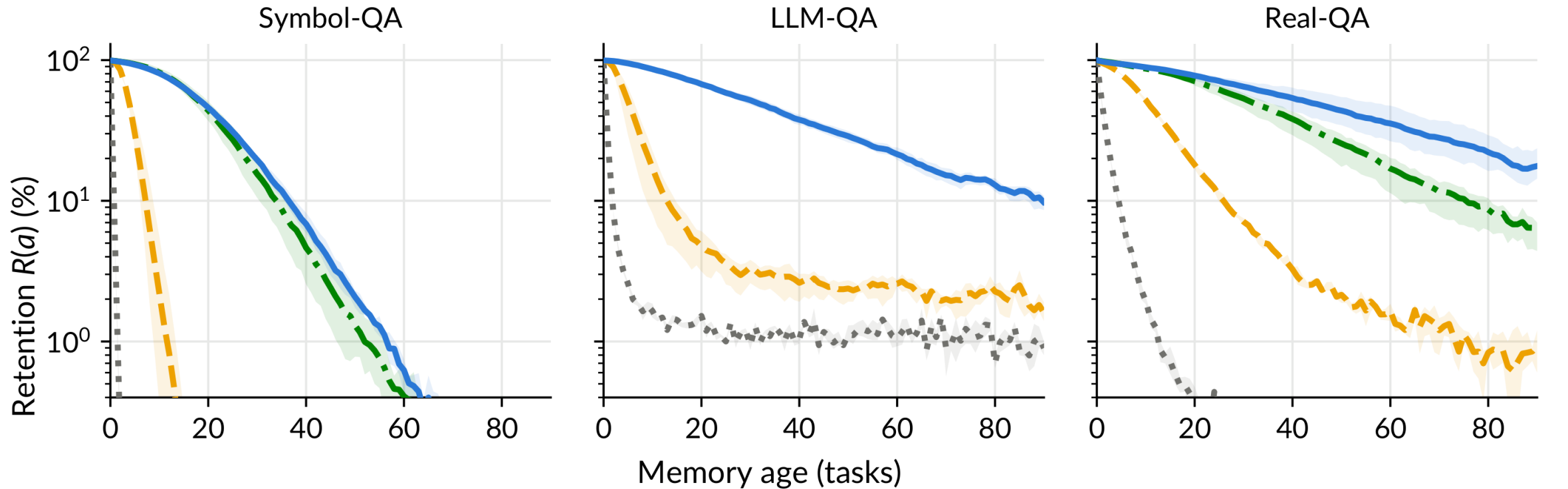
..... Naive fine-tuning - - - Best single mechanism

Standard training on stream of data



Naive fine-tuning Best single mechanism All mechanisms combined Dataset-specific best composition

Standard training on stream of data



..... Naive fine-tuning - - - Best single mechanism - · - All mechanisms combined — Dataset-specific best composition

Good news: Continual learning mechanism compose!
Bad news: They're insufficient!

Takeaways

- Standard fine-tuning is forgetful.
- Standalone CL mechanisms retain little knowledge after many-session horizon.
- But the CL mechanisms are composable!
- Still far from solved: memories keep fading.

Takeaways

- Standard fine-tuning is forgetful.
- Standalone CL mechanisms retain little knowledge after many-session horizon.
- But the CL mechanisms are composable!
- Still far from solved: memories keep fading.

Takeaways

- Standard fine-tuning is forgetful.
- Standalone CL mechanisms retain little knowledge after many-session horizon.
- But the CL mechanisms are composable!
- Still far from solved: memories keep fading.

Takeaways

- Standard fine-tuning is forgetful.
- Standalone CL mechanisms retain little knowledge after many-session horizon.
- But the CL mechanisms are composable!
- Still far from solved: memories keep fading.

Takeaways

- Standard fine-tuning is forgetful.
- Standalone CL mechanisms retain little knowledge after many-session horizon.
- But the CL mechanisms are composable!
- Still far from solved: memories keep fading.

Roadmap: long-running agents

- Bet 1: **Expand context:**
 - Keep more of the history *directly available*.



Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters:**
 - Update model parameters through *continued* training
- Bet 3: Manage **context effectively:**
 - Selectively preserve what matters through *compression*.

Roadmap: long-running agents

- Bet 1: **Expand context:**
 - Keep more of the history *directly available*.



Work within a **bounded** context:

- Bet 2: **Internalize knowledge in model parameters:**
 - Update model parameters through *continued* training
- Bet 3: Manage **context effectively:**
 - Selectively preserve what matters through *compression*.

Why bounded context?

Challenge of scaling context

Why bounded context?

Challenge of scaling context

- There are fundamental challenges in how to effectively scale the context window of models (cf. our earlier discussion).

Why bounded context?

Cost

\$ \$ \$

Why bounded context?

Context rot

- Increasing context lowers attention fidelity

Why bounded context?

Context rot

- Increasing context lowers attention fidelity

The effective context window of models is much smaller than what they're claimed to be.

Models	Claimed Length	Effective Length
Llama2 (7B)	4K	-
Gemini-1.5-Pro	1M	>128K
GPT-4	128K	64K
Llama3.1 (70B)	128K	64K
Qwen2 (72B)	128K	32K
Command-R-plus (104B)	128K	32K
GLM4 (9B)	1M	64K
Llama3.1 (8B)	128K	32K
GradientAI/Llama3 (70B)	1M	16K
Mixtral-8x22B (39B/141B)	64K	32K

Hsieh et al. RULER: What's the Real Context Size of Your Long-Context Language Models? COLM, 2024.

The common mechanism to shorten context:
“Compaction”

The common mechanism to shorten context:

“Compaction”

- Compaction :=
 1. condenses that context into a short summary,
 2. releases the memory, and
 3. generation continues from the summary.

The common mechanism to shorten context:

“Compaction”

- Compaction :=
 1. condenses that context into a short summary,
 2. releases the memory, and
 3. generation continues from the summary.

The common mechanism to shorten context:

“Compaction”

- Compaction :=
 1. condenses that context into a short summary,
 2. releases the memory, and
 3. generation continues from the summary.

The common mechanism to shorten context:

“Compaction”

- Compaction :=
 1. condenses that context into a short summary,
 2. releases the memory, and
 3. generation continues from the summary.

Compaction in today's agents

- Nowadays compaction is used in all agentic tools.
- It used to be that it triggered near the point of exceeding the window.
- Increasingly compaction is frequent and invisible.
- It works more or less, **but**:
 - Hard to study it scientifically.
 - Not perfect and not doing it at the right time will be harmful.

Compaction in today's agents

- Nowadays compaction is used in all agentic tools.
- It used to be that it triggered near the point of exceeding the window.
- Increasingly compaction is frequent and invisible.
- It works more or less, **but**:
 - Hard to study it scientifically.
 - Not perfect and not doing it at the right time will be harmful.

Compaction in today's agents

- Nowadays compaction is used in all agentic tools.
- It used to be that it triggered near the point of exceeding the window.
- Increasingly compaction is frequent and invisible.
- It works more or less, **but**:
 - Hard to study it scientifically.
 - Not perfect and not doing it at the right time will be harmful.

Compaction in today's agents

- Nowadays compaction is used in all agentic tools.
- It used to be that it triggered near the point of exceeding the window.
- Increasingly compaction is frequent and invisible.
- It works more or less, **but**:
 - Hard to study it scientifically.
 - Not perfect and not doing it at the right time will be harmful.

Research question

Research question

Can *off-the-shelf* LLMs decide
when and *how* to compact their own history?

Research question

Can *off-the-shelf* LLMs decide *when* and *how* to compact their own history?

How?

Research question

Can *off-the-shelf* LLMs decide *when* and *how* to compact their own history?

When?

How?

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline

P

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

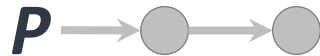
No Compression baseline

P → ●

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline



QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline



QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline



QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline



QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline



QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline



QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline



QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline



Final Answer:

X None

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

No Compression baseline



Final Answer:

X None

Run out of context runway + costly.

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

***Fixed-interval* compaction (every k turns or % of context limit)**

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

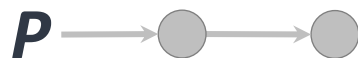
***Fixed-interval* compaction (every k turns or % of context limit)**

P

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

***Fixed-interval* compaction (every k turns or % of context limit)**



QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

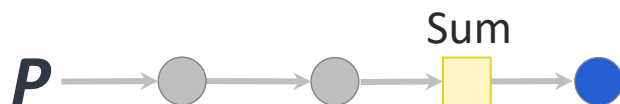
***Fixed-interval* compaction (every k turns or % of context limit)**



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

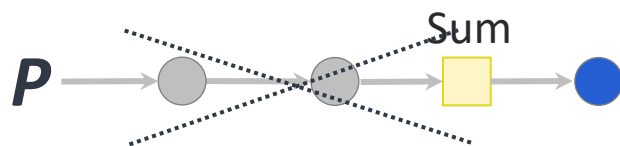
***Fixed-interval* compaction (every k turns or % of context limit)**



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

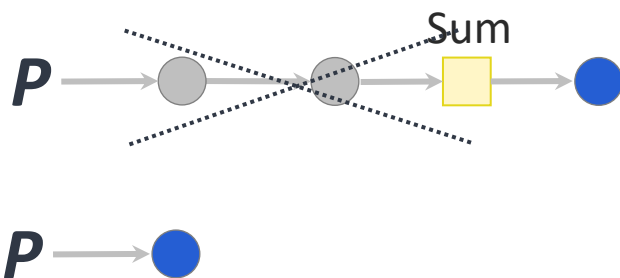
***Fixed-interval* compaction (every k turns or % of context limit)**



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

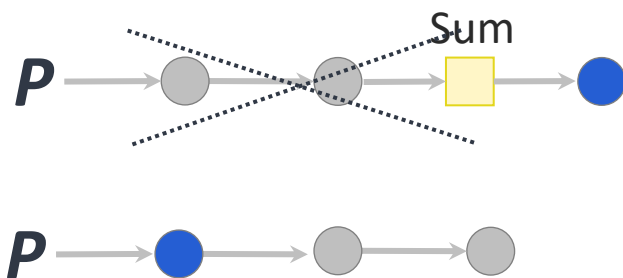
Fixed-interval compaction (every k turns or % of context limit)



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

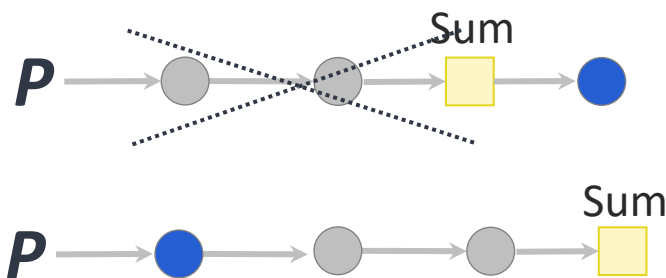
Fixed-interval compaction (every k turns or % of context limit)



QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

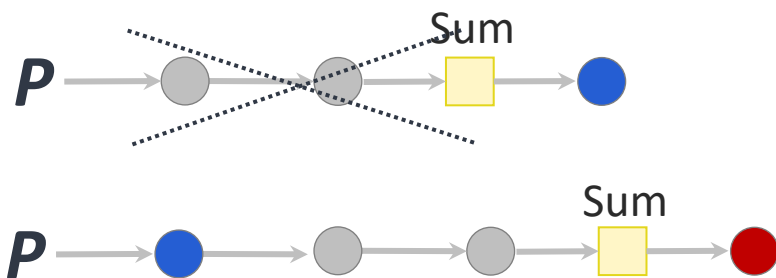
***Fixed-interval* compaction (every k turns or % of context limit)**



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

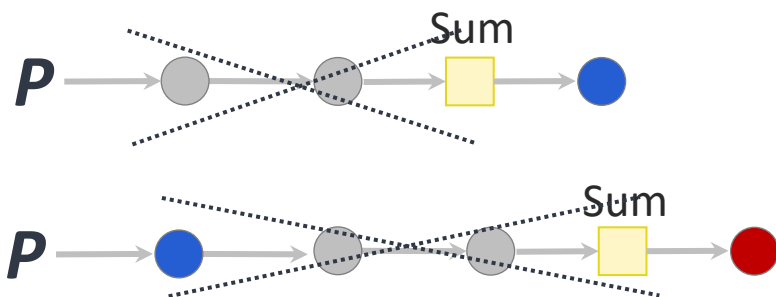
Fixed-interval compaction (every k turns or % of context limit)



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

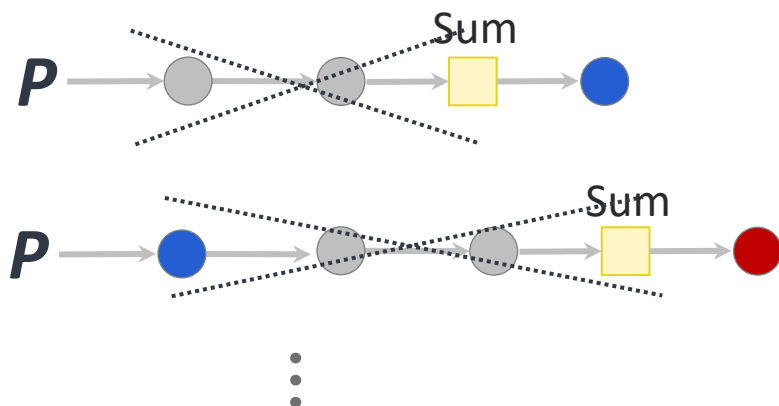
Fixed-interval compaction (every k turns or % of context limit)



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

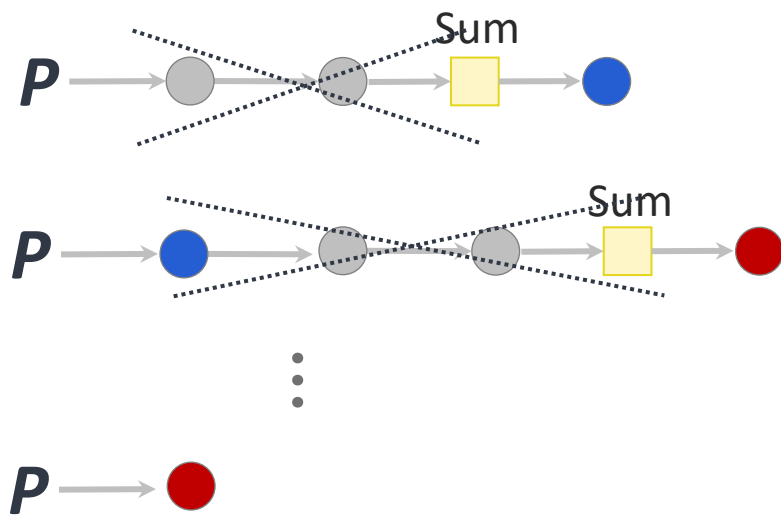
Fixed-interval compaction (every k turns or % of context limit)



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

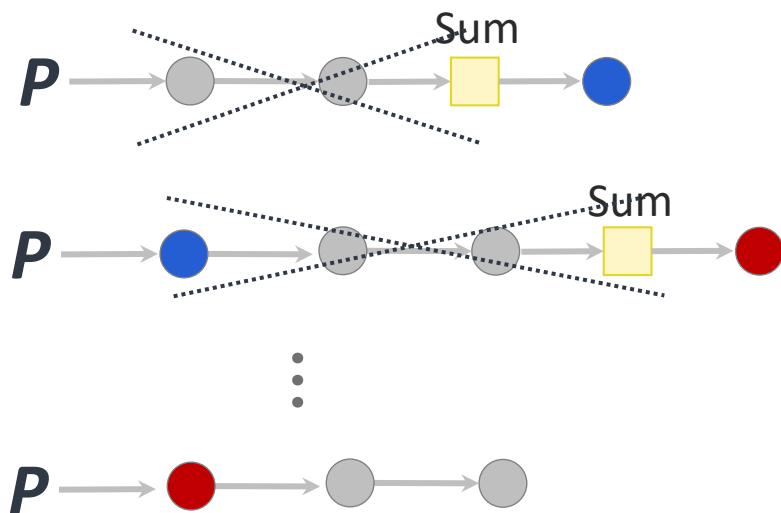
Fixed-interval compaction (every k turns or % of context limit)



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

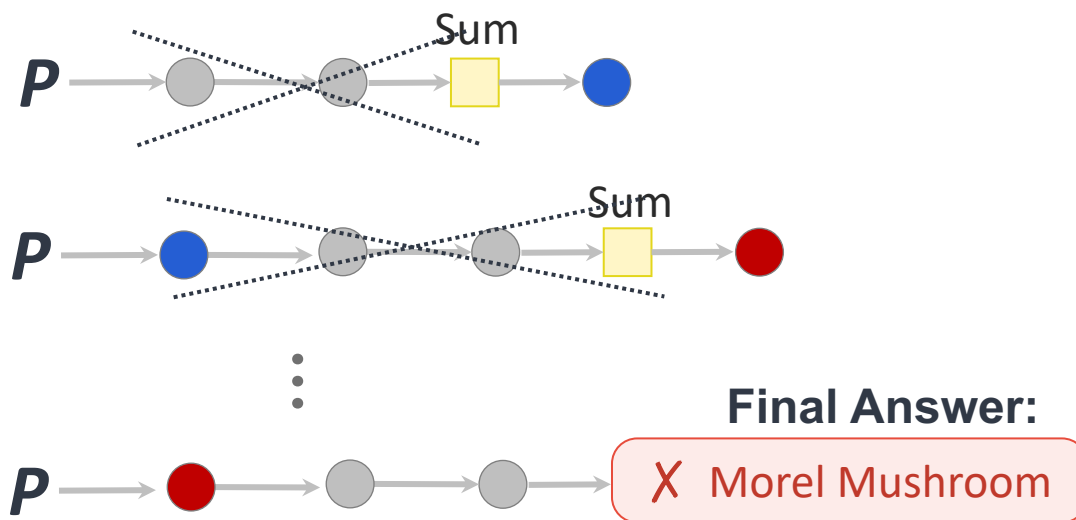
Fixed-interval compaction (every k turns or % of context limit)



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

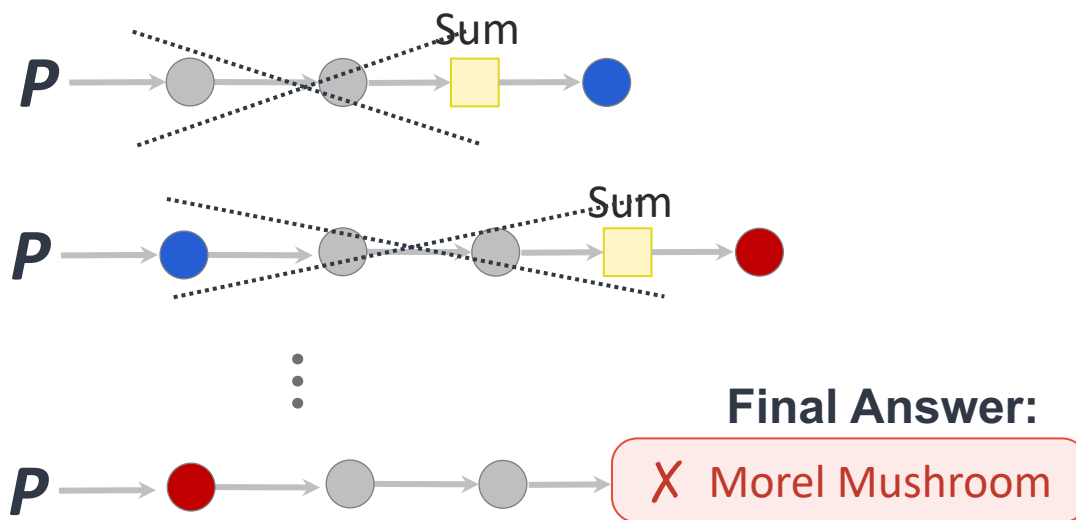
Fixed-interval compaction (every k turns or % of context limit)



QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

Fixed-interval compaction (every k turns or % of context limit)

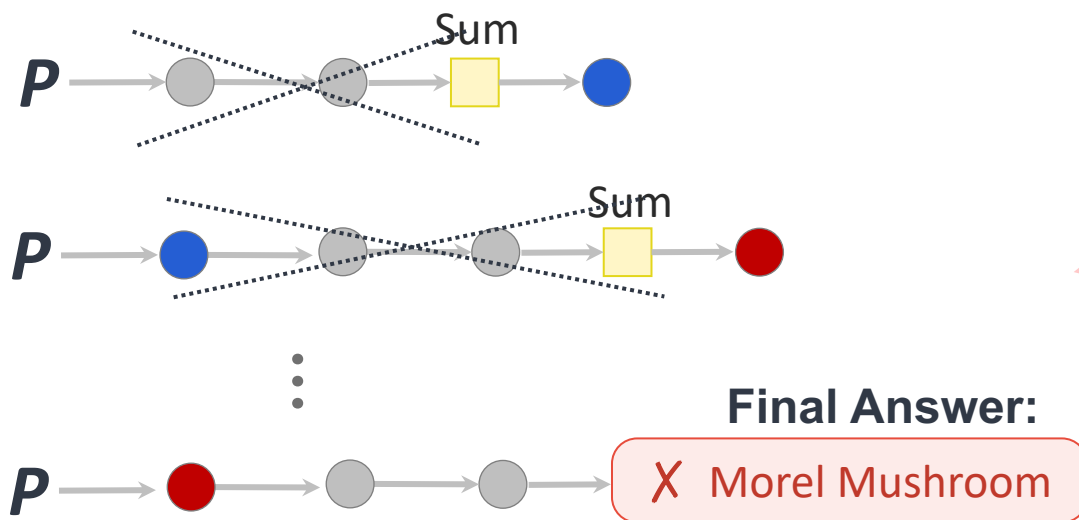


AgentFold, 2025; FoldAct, 2025, Context-Folding, 2025; Reasoning Cache, 2026; InftyThink, 2026; IterResearch, 2026; ReSuM, 2026;...

QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

Fixed-interval compaction (every k turns or % of context limit)



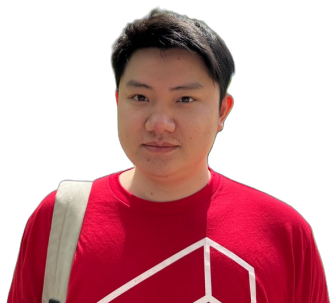
poorly timed summarization
— drops verified facts;
restates intent vaguely.

Self-Compact

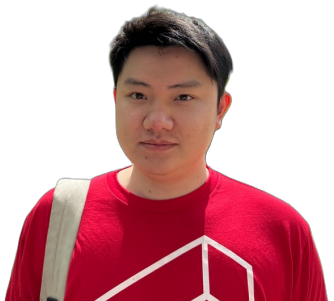
Self-Compacting Language Model Agents

Tianjian Li♠ Jingyu Zhang♠ William Jurayj♠ Xi Wang♠ Chuanyang Jin♠
Mehrdad Farajtabar♥ Eric Nalisnick♠ Daniel Khashabi♠

NeurIPS 2026

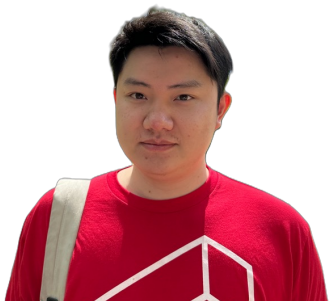


Self-Compact



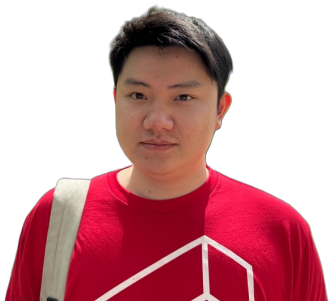
Self-Compact

1. **Compaction tool (C)**: how to compact (summarize)
2. **Rubric (R)**: when to fire



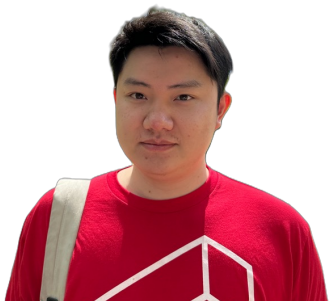
Self-Compact

1. **Compaction tool (C)**: how to compact (summarize)
2. **Rubric (R)**: when to fire



Self-Compact

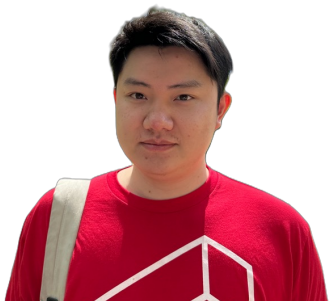
1. **Compaction tool (C)**: how to compact (summarize)
2. **Rubric (R)**: when to fire



Self-Compact

1. **Compaction tool (C)**: how to compact (summarize)
2. **Rubric (R)**: when to fire

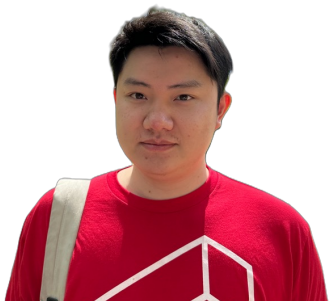
- Run reasoning as usual.
- After each turn, run **R** $\in$ {COMPACT, CONTINUE}.
- If CONTINUE, continue thinking (remove rubric from history).
- If COMPACT, call **C** and replace the context w/ summary.



Self-Compact

1. **Compaction tool (C)**: how to compact (summarize)
2. **Rubric (R)**: when to fire

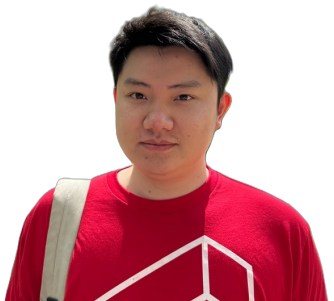
- Run reasoning as usual.
- After each turn, run **R** $\in$ {COMPACT, CONTINUE}.
- If CONTINUE, continue thinking (remove rubric from history).
- If COMPACT, call **C** and replace the context w/ summary.



Self-Compact

1. **Compaction tool (C)**: how to compact (summarize)
2. **Rubric (R)**: when to fire

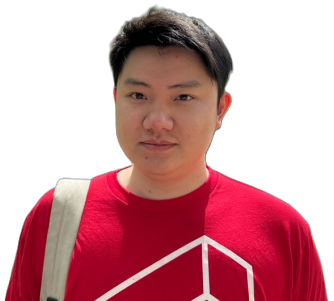
- Run reasoning as usual.
- After each turn, run **R** $\in$ {COMPACT, CONTINUE}.
- If CONTINUE, continue thinking (remove rubric from history).
- If COMPACT, call **C** and replace the context w/ summary.



Self-Compact

1. **Compaction tool (C)**: how to compact (summarize)
2. **Rubric (R)**: when to fire

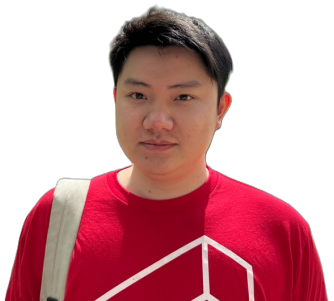
- Run reasoning as usual.
- After each turn, run **R** $\in$ {**COMPACT**, **CONTINUE**}.
- If **CONTINUE**, continue thinking (remove rubric from history).
- If **COMPACT**, call **C** and replace the context w/ summary.



Self-Compact

1. **Compaction tool (C)**: how to compact (summarize)
2. **Rubric (R)**: when to fire

- Run reasoning as usual.
- After each turn, run **R** $\in$ {COMPACT, CONTINUE}.
- If CONTINUE, continue thinking (remove rubric from history).
- If COMPACT, call **C** and replace the context w/ summary.



Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

P

Self-Compact, visually

QUESTION (BrowseComp)

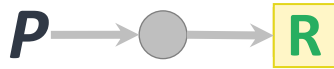
Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

P → ●

Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.



Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

$P \rightarrow \bullet \rightarrow \boxed{R} \in \{\text{COMPACT}, \text{CONTINUE}\}$

Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

$P \rightarrow \bullet \rightarrow R \in \{\text{COMPACT}, \text{CONTINUE}\}$

Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.



Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

$P \rightarrow \bullet \rightarrow \bullet \rightarrow \boxed{R} \in \{\text{COMPACT, CONTINUE}\}$

Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (P): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

$P \rightarrow \bullet \rightarrow \bullet \rightarrow \boxed{R} \in \{\text{COMPACT}, \text{CONTINUE}\}$

Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.



Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.



Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

P → ● → ● → ● → **R** ∈ {COMPACT, CONTINUE}

Self-Compact, visually

QUESTION (BrowseComp)

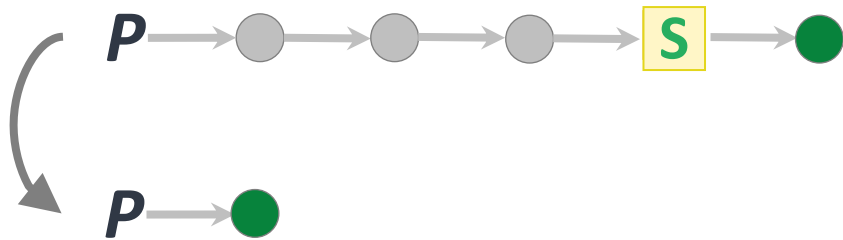
Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

P → ● → ● → ● → **R** ∈ {**COMPACT**, **CONTINUE**}

Self-Compact, visually

QUESTION (BrowseComp)

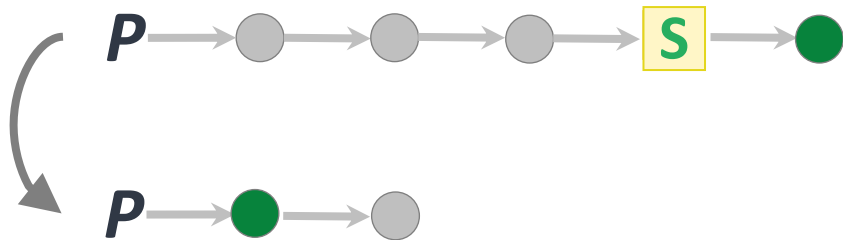
Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.



Self-Compact, visually

QUESTION (BrowseComp)

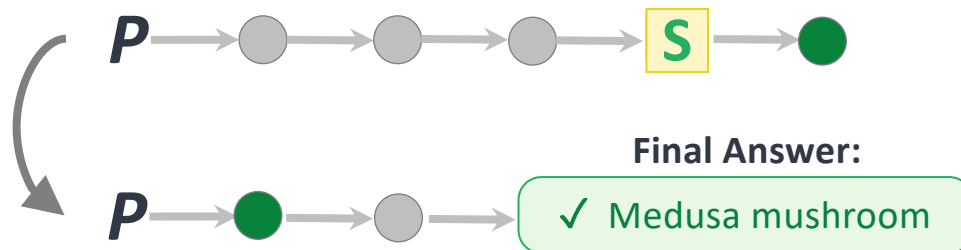
Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.



Self-Compact, visually

QUESTION (BrowseComp)

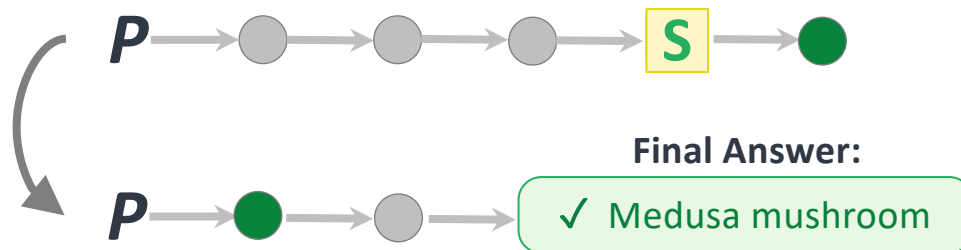
Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.



Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.

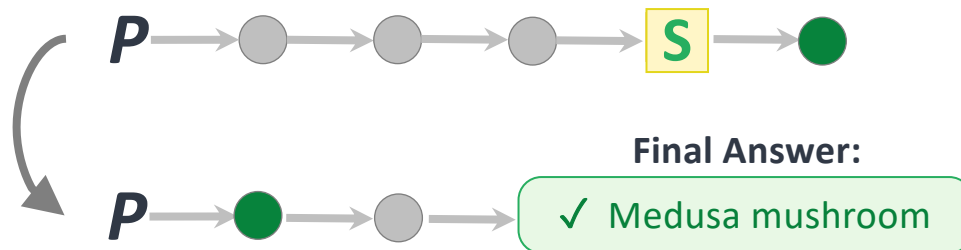


adaptive summarization —
consolidates facts; lets model
escape from loops.

Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.



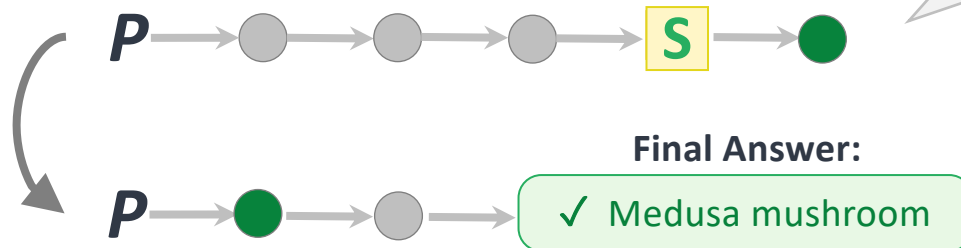
Shortens a long context
(less tokens to attend)

adaptive summarization —
consolidates facts; lets model
escape from loops.

Self-Compact, visually

QUESTION (BrowseComp)

Input prompt (*P*): Identify a rare fungus appearing in clusters after rainfall, with raised scales on its cap, named by a French expert in the 1980s, with potential antifungal properties. Its English name matches a 1980s film character; the film was inspired by a 1970s bronze statuette. Two-word name; first word has 3 syllables, ends in vowel.



Minimal overhead
because of KV-reuse

adaptive summarization —
consolidates facts; lets model
escape from loops.

Shortens a long context
(less tokens to attend)

Self-Compact: Making it work

- The Rubric prompt is the crucial design component.
- Guidelines for when to compact we found effective:
 1. A subtask is complete:
 - Preserve its result; discard unnecessary intermediate work.
 2. The model is stuck:
 - Consolidate findings; compaction gives it an opportunity to break out.

Self-Compact: Making it work

- The Rubric prompt is the crucial design component.
- Guidelines for when to compact we found effective:
 1. A subtask is complete:
 - Preserve its result; discard unnecessary intermediate work.
 2. The model is stuck:
 - Consolidate findings; compaction gives it an opportunity to break out.

Self-Compact: Making it work

- The Rubric prompt is the crucial design component.
- Guidelines for when to compact we found effective:
 1. A subtask is complete:
 - Preserve its result; discard unnecessary intermediate work.
 2. The model is stuck:
 - Consolidate findings; compaction gives it an opportunity to break out.

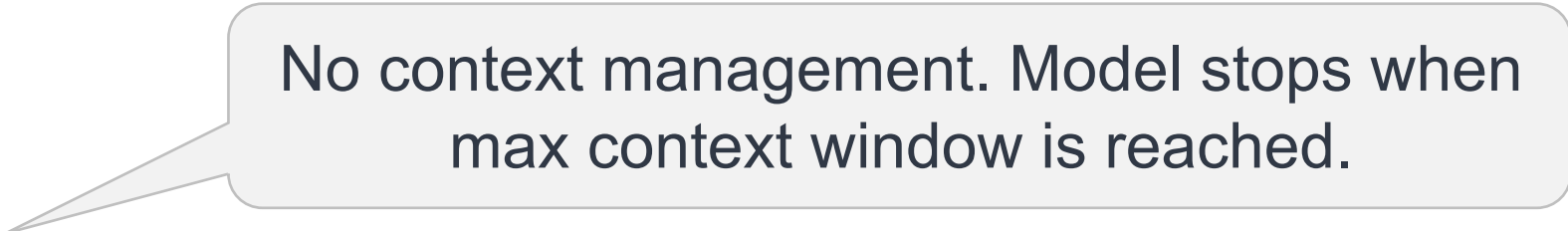
Self-Compact: Making it work

- The Rubric prompt is the crucial design component.
- Guidelines for when to compact we found effective:
 1. A subtask is complete:
 - Preserve its result; discard unnecessary intermediate work.
 2. The model is stuck:
 - Consolidate findings; compaction gives it an opportunity to break out.

Empirical Results: Agentic Search

- No Compaction
- Fixed-Interval Summary
- SelfCompact (Ours)

Empirical Results: Agentic Search



No context management. Model stops when max context window is reached.

- No Compaction
- Fixed-Interval Summary
- SelfCompact (Ours)

Empirical Results: Agentic Search

- No Compaction
- Fixed-Interval Summary
- SelfCompact (Ours)

No context management. Model stops when max context window is reached.

Summarization is triggered when 30% of the max context window is consumed *

Empirical Results: Agentic Search

- No Compaction
- Fixed-Interval Summary
- SelfCompact (Ours)

No context management. Model stops when max context window is reached.

Summarization is triggered when 30% of the max context window is consumed *

Let the model decide when to compact.

* Uses scaffold of Lee et al. 2026. “Agentic Aggregation for Parallel Scaling of Long-Horizon Agentic Tasks”

Empirical Results: Agentic Search

- Mimo-V2-Flash (309B; 15B active)

Accuracy



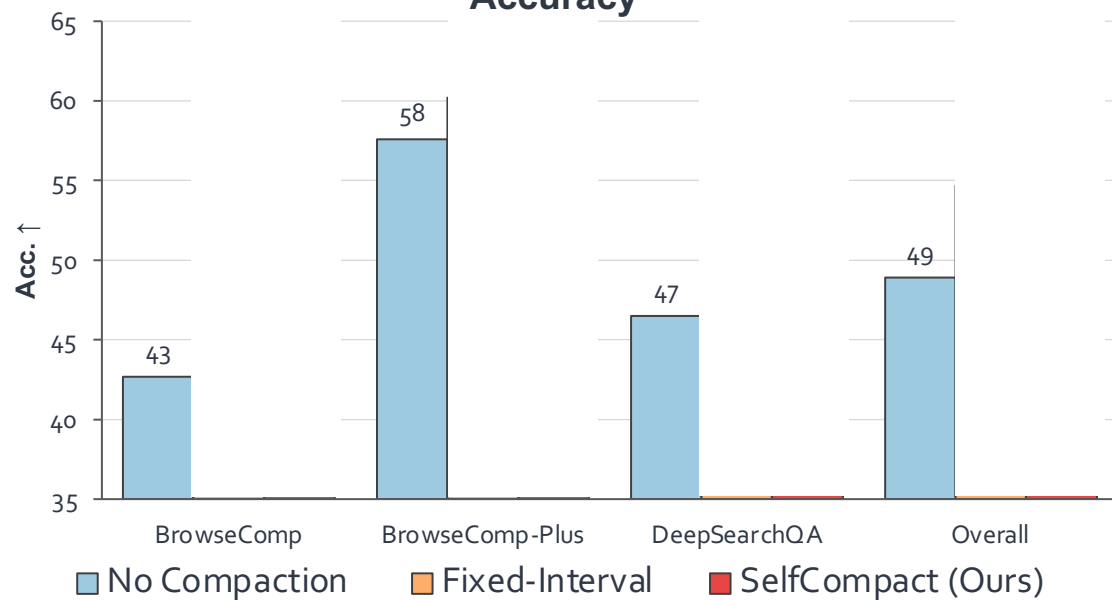
Cost



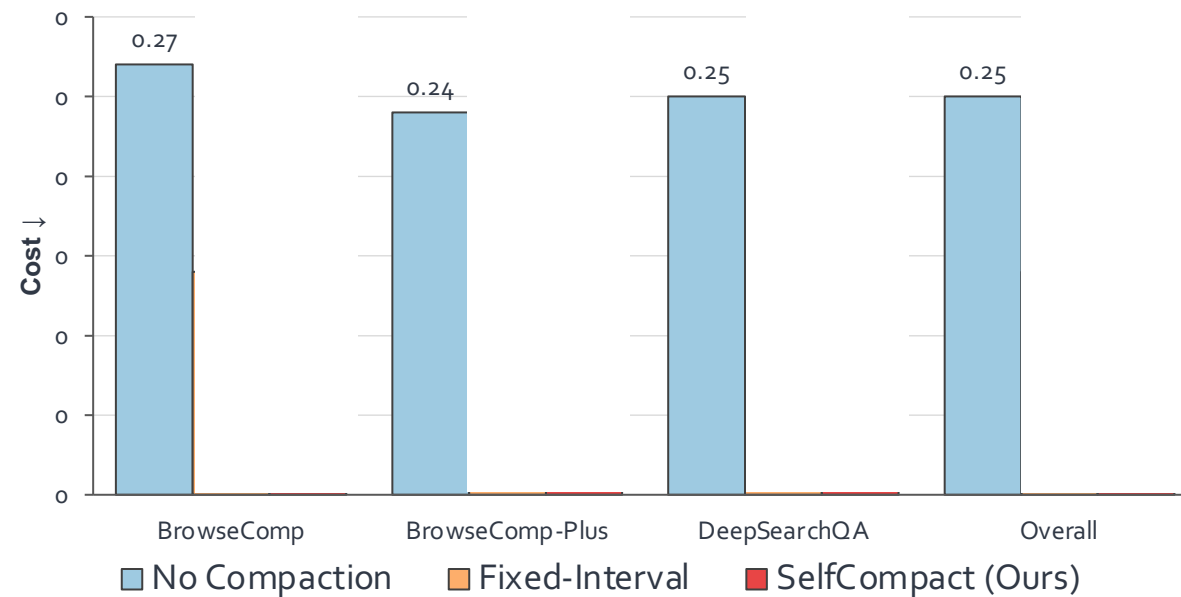
Empirical Results: Agentic Search

- Mimo-V2-Flash (309B; 15B active)

Accuracy



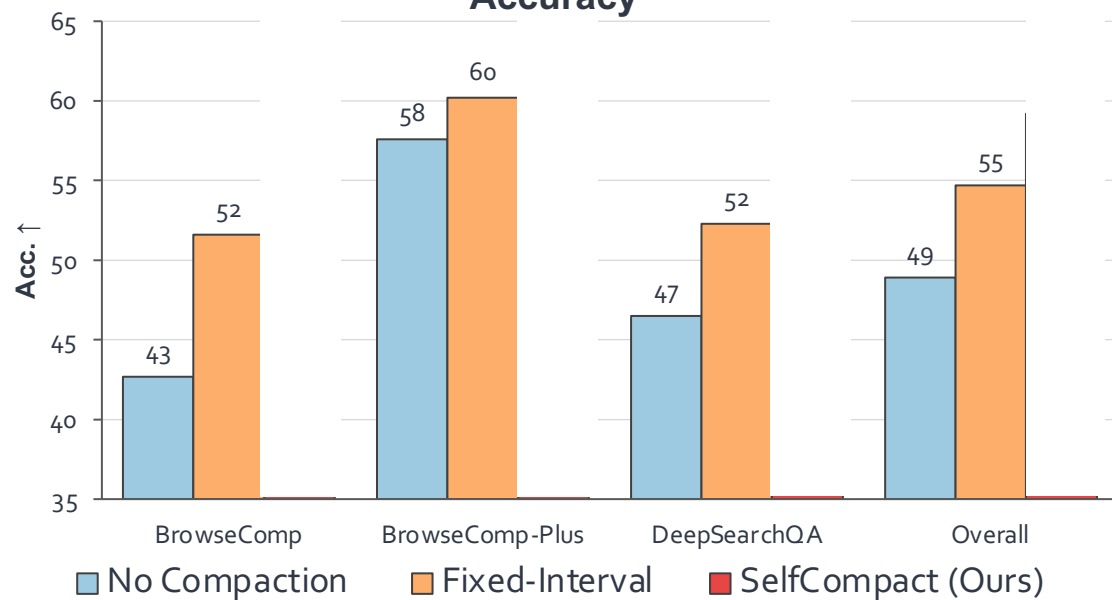
Cost



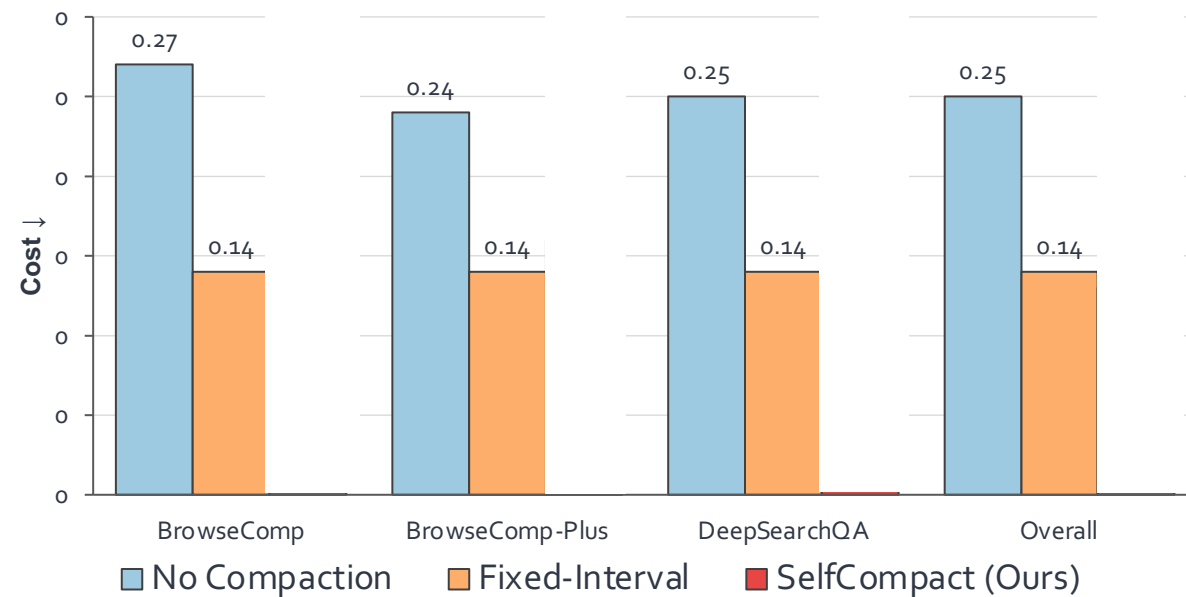
Empirical Results: Agentic Search

- Mimo-V2-Flash (309B; 15B active)

Accuracy

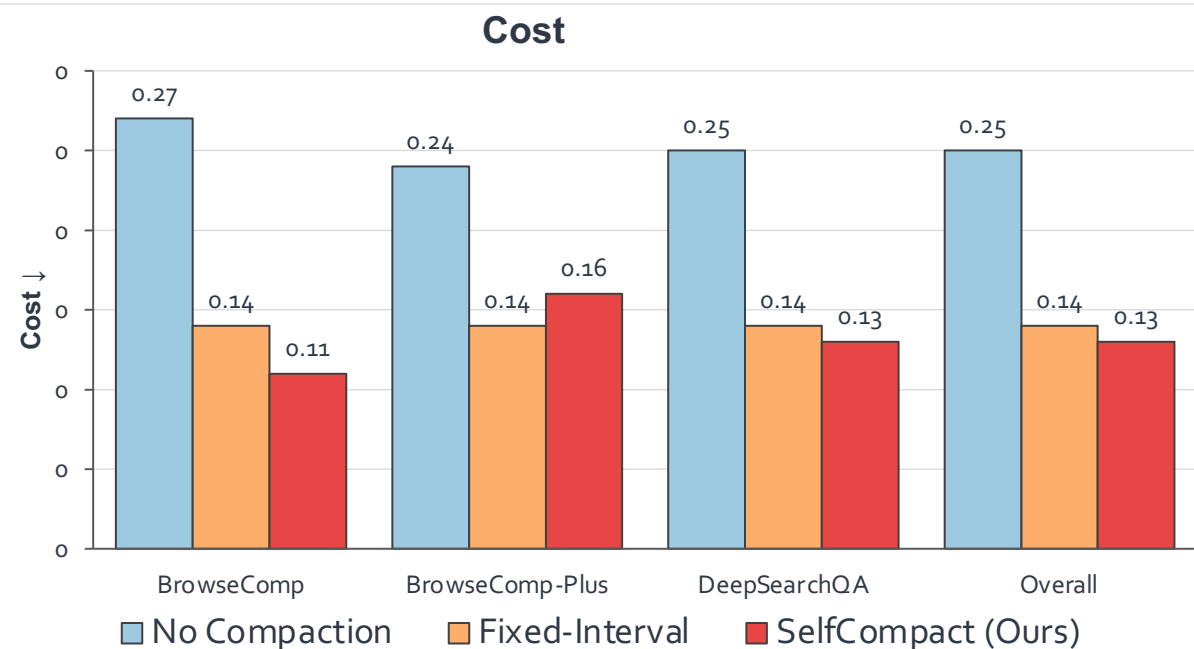
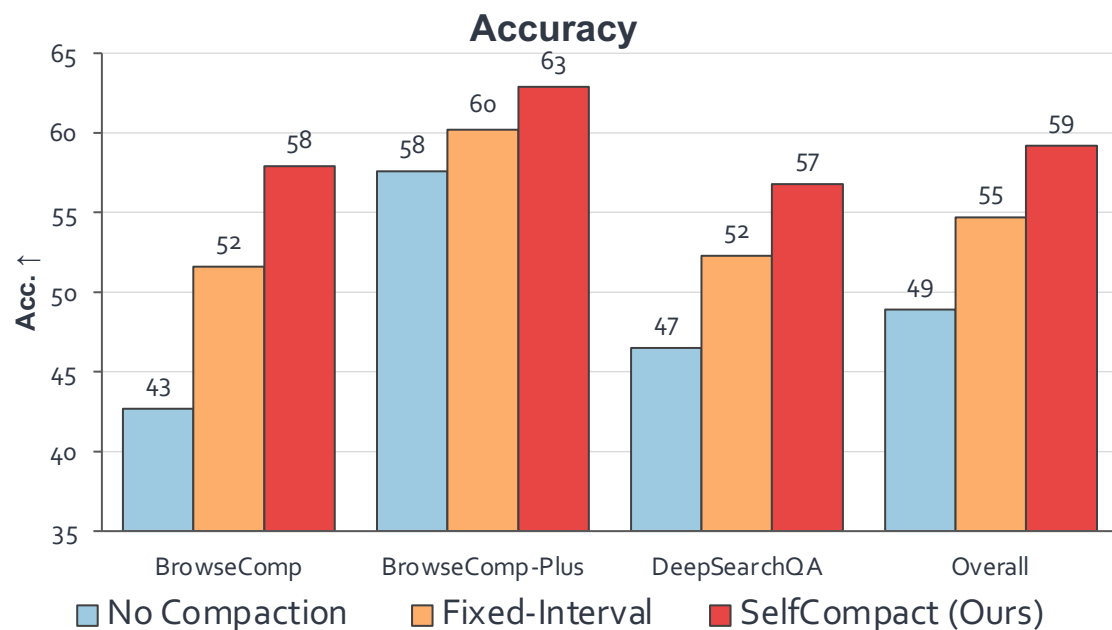


Cost



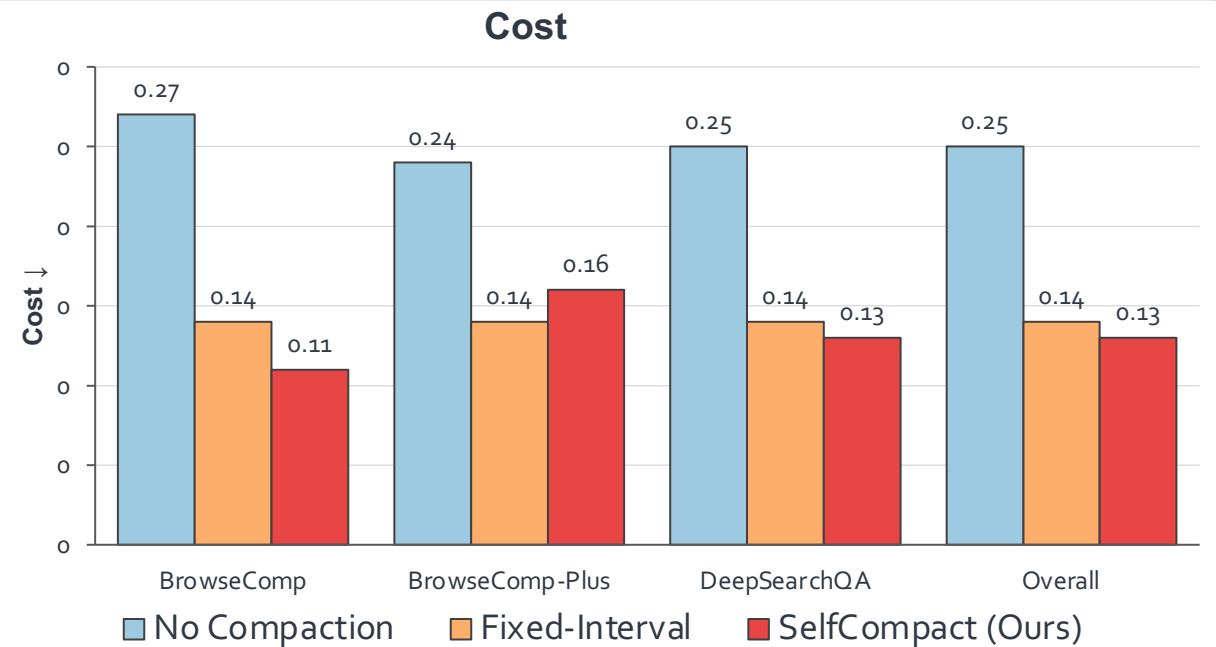
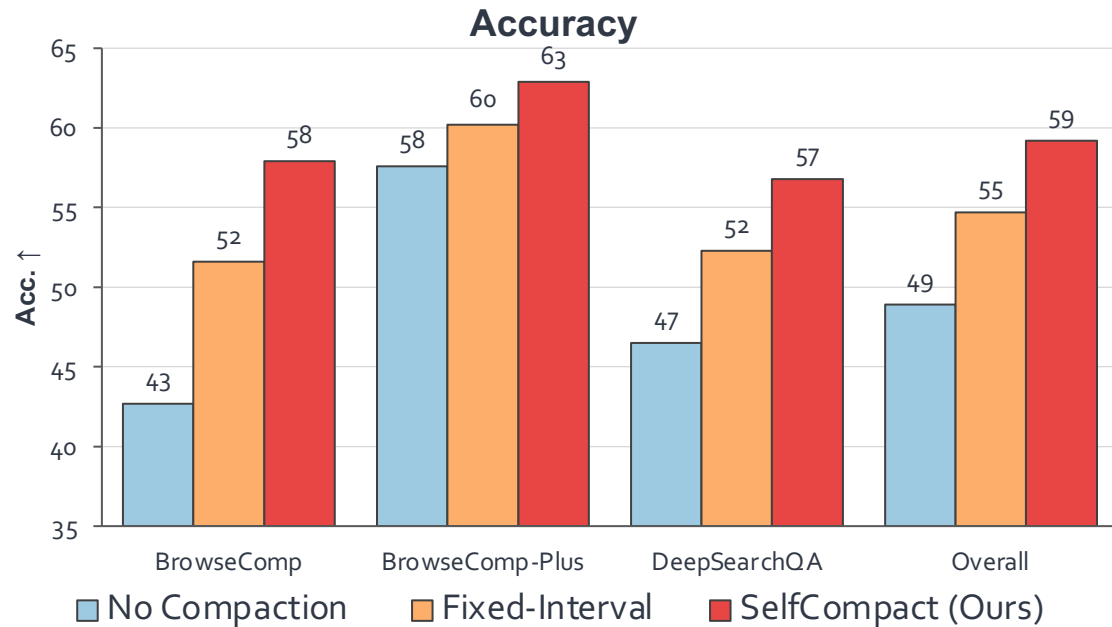
Empirical Results: Agentic Search

- Mimo-V2-Flash (309B; 15B active)



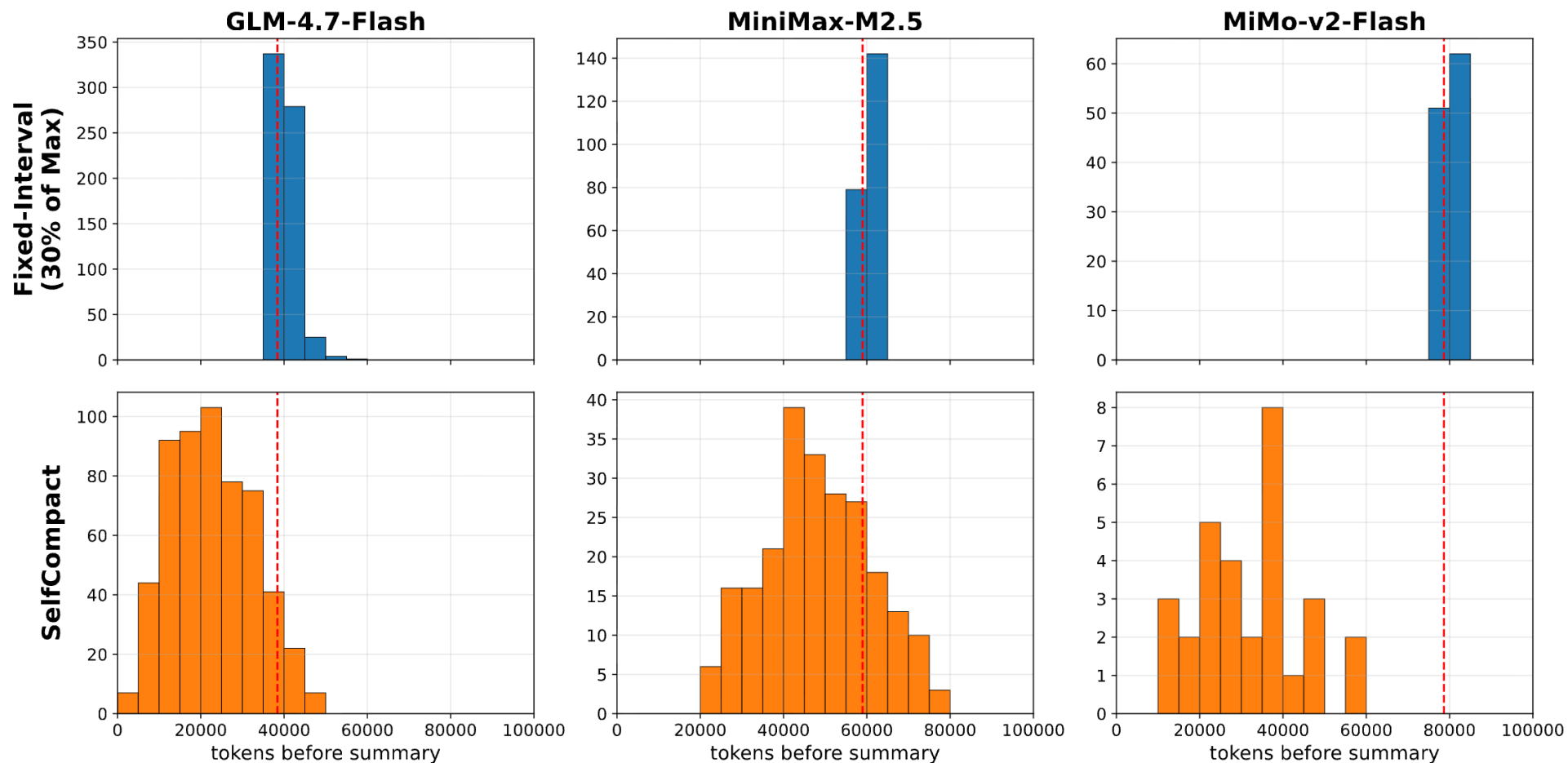
Empirical Results: Agentic Search

- Mimo-V2-Flash (309B; 15B active)

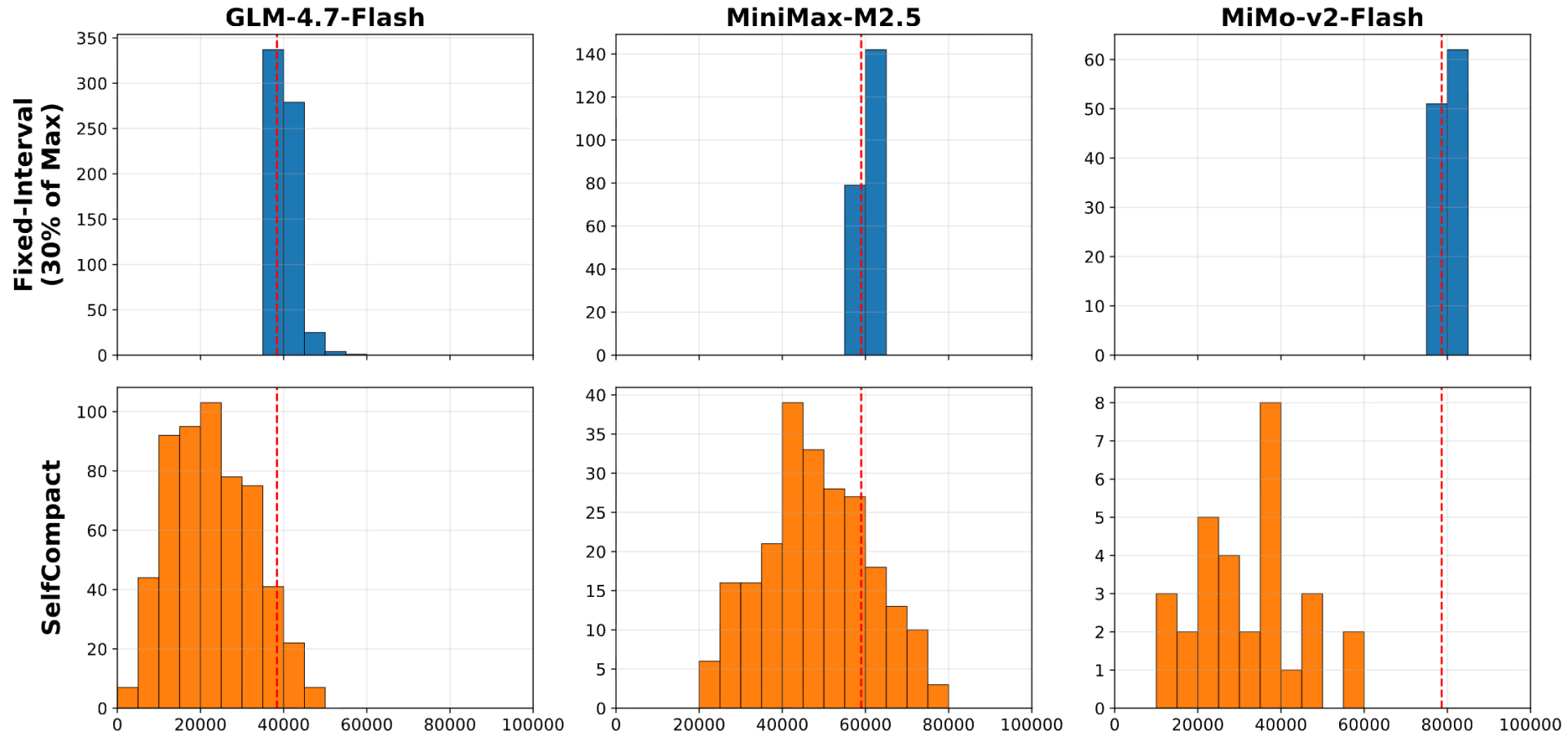


Improves accuracy of fixed-interval compaction while maintaining cost.

Why it works: compaction frequency



Why it works: compaction frequency



Fixed-Interval triggers at the threshold by construction;
SelfCompact spreads them across the budget.

“Compacting” the content (takeaways)

- Self-Compact: rubric-gated adaptive compaction.
 - Lightweight and training-free.
 - A meta-cognitive capability.
 - Strong empirical results
- Our study is quite simple and likely there is a lot of headroom.
- Call for scientific study of context management.

“Compacting” the content (takeaways)

- Self-Compact: rubric-gated adaptive compaction.
 - Lightweight and training-free.
 - A meta-cognitive capability.
 - Strong empirical results
- Our study is quite simple and likely there is a lot of headroom.
- Call for scientific study of context management.

“Compacting” the content (takeaways)

- Self-Compact: rubric-gated adaptive compaction.
 - Lightweight and training-free.
 - A meta-cognitive capability.
 - Strong empirical results
- Our study is quite simple and likely there is a lot of headroom.
- Call for scientific study of context management.

The long and short of it:

Toward agents that learn and remember over time

- **Context scaling:**

- There are fundamental challenges in scaling context window of LLMs.
- Long-running agents [probably] need more than longer histories.

- **Internalizing knowledge in parametric memory:**

- Continual learning improves retention.
- But memories still fade.

- **Context management:**

- Context management (e.g., compaction) as a useful hill to climb.
- But unclear how far we can climb this hill.

The long and short of it:

Toward agents that learn and remember over time

- **Context scaling:**

- There are fundamental challenges in scaling context window of LLMs.
- Long-running agents [probably] need more than longer histories.

- **Internalizing knowledge in parametric memory:**

- Continual learning improves retention.
- But memories still fade.

- **Context management:**

- Context management (e.g., compaction) as a useful hill to climb.
- But unclear how far we can climb this hill.

The long and short of it:

Toward agents that learn and remember over time

- **Context scaling:**

- There are fundamental challenges in scaling context window of LLMs.
- Long-running agents [probably] need more than longer histories.

- **Internalizing knowledge in parametric memory:**

- Continual learning improves retention.
- But memories still fade.

- **Context management:**

- Context management (e.g., compaction) as a useful hill to climb.
- But unclear how far we can climb this hill.

The long and short of it:

Toward agents that learn and remember over time

- **Context scaling:**

- There are fundamental challenges in scaling context window of LLMs.
- Long-running agents [probably] need more than longer histories.

- **Internalizing knowledge in parametric memory:**

- Continual learning improves retention.
- But memories still fade.

- **Context management:**

- Context management (e.g., compaction) as a useful hill to climb.
- But unclear how far we can climb this hill.

The long and short of it:

Toward agents that learn and remember over time

- **Context scaling:**
 - There are fundamental challenges in scaling context window of LLMs.
 - Long-running agents [probably] need more than longer histories.
- **Internalizing knowledge in parametric memory:**
 - Continual learning improves retention.
 - But memories still fade.
- **Context management:**
 - Context management (e.g., compaction) as a useful hill to climb.
 - But unclear how far we can climb this hill.

Open Problems

- **Fundamentals:** What limits are inherent, and what reflects limitations of current methods?
- **Integration:** How should context, external memory, and parameter updates work together?
- **Evaluation:** How do we design tasks that inherently require learning and remembering over long horizons?

Open Problems

- **Fundamentals:** What limits are inherent, and what reflects limitations of current methods?
- **Integration:** How should context, external memory, and parameter updates work together?
- **Evaluation:** How do we design tasks that inherently require learning and remembering over long horizons?

Open Problems

- **Fundamentals:** What limits are inherent, and what reflects limitations of current methods?
- **Integration:** How should context, external memory, and parameter updates work together?
- **Evaluation:** How do we design tasks that inherently require learning and remembering over long horizons?

Open Problems

- **Fundamentals:** What limits are inherent, and what reflects limitations of current methods?
- **Integration:** How should context, external memory, and parameter updates work together?
- **Evaluation:** How do we design tasks that inherently require learning and remembering over long horizons?

Thanks to our wonderful collaborators on these projects!

Core contributors:



Tianjian Li



Zheyuan Zhang



Alvin Zhang



Arda Uzunoglu

Other contributors:

Tianmin Shu, Benjamin Van Durme, Eric Nalisnick, Jingyu Zhang,
William Jurayj, Xi Wang, Chuanyang Jin, Mehrdad Farajtabar

Sponsors:

