Integrating Language Models with Evolving External Knowledge

Di Wu

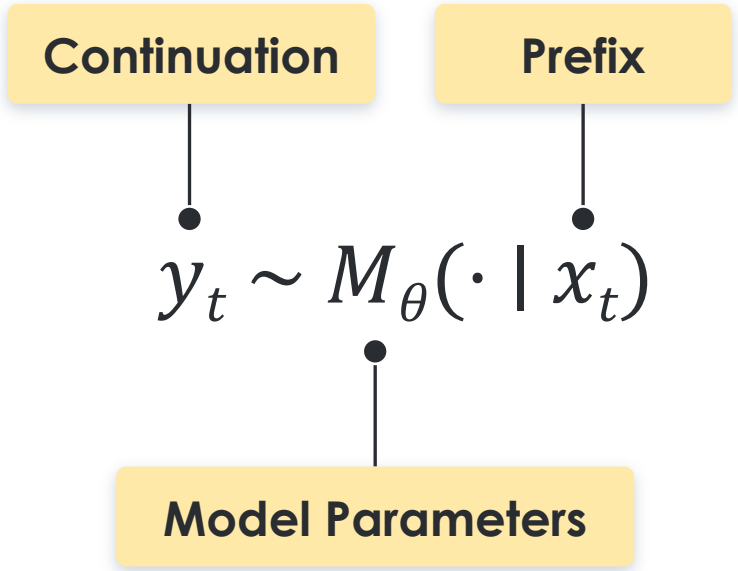
Doctoral Defense, Computer Science

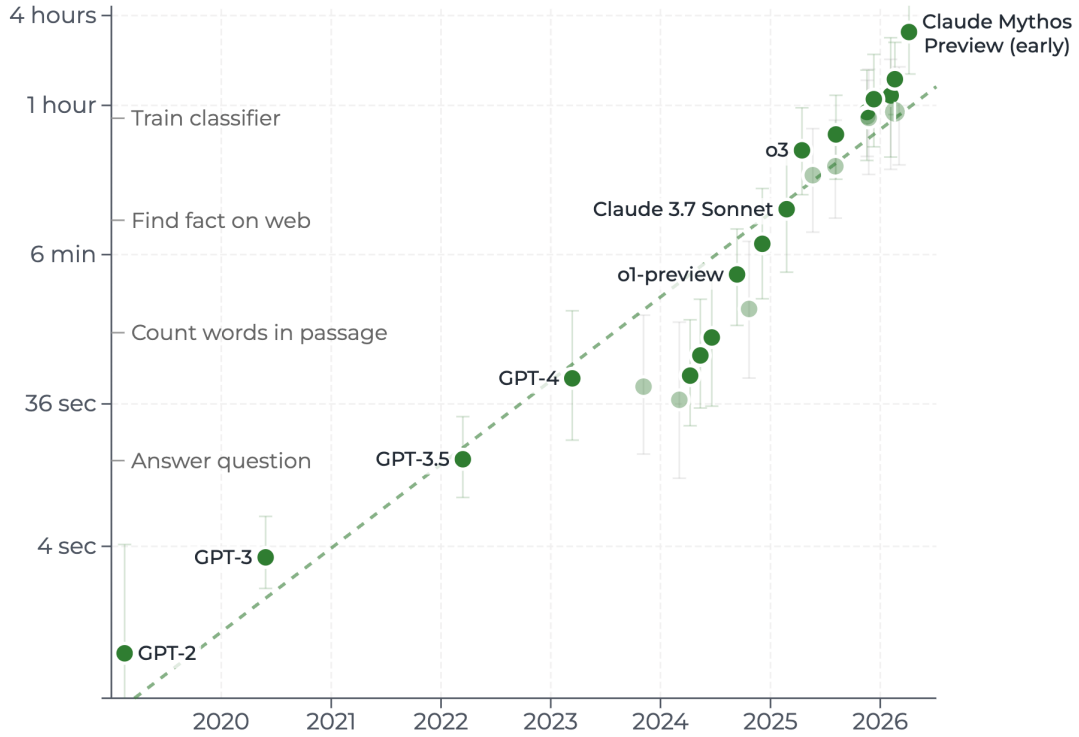
November 9, 2026

UCLA

LLMs become stronger and stronger…

- Large language models (LLMs) achieve promising utility by scaling next token prediction.



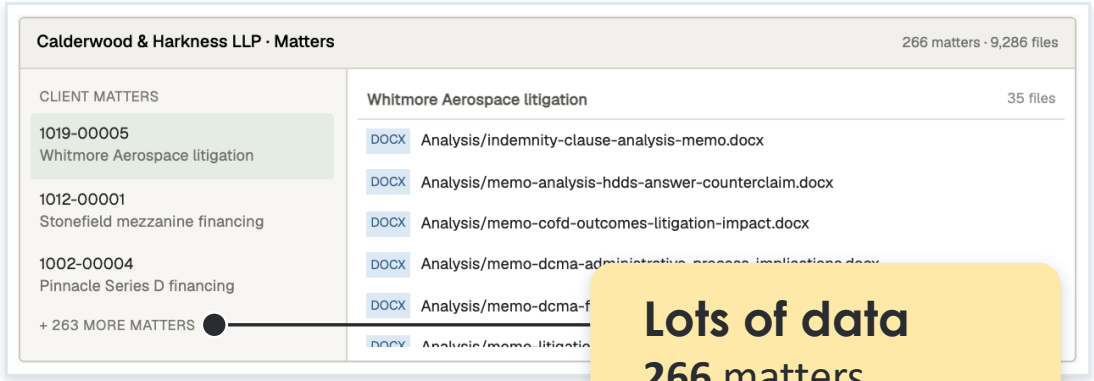


Source: [the METR project]

The Knowledge Gap

- However, deploying an LLM in a constantly evolving world requires knowledge augmentation.
- Deployed in companies, an LLM needs to navigate **complex private knowledge**.

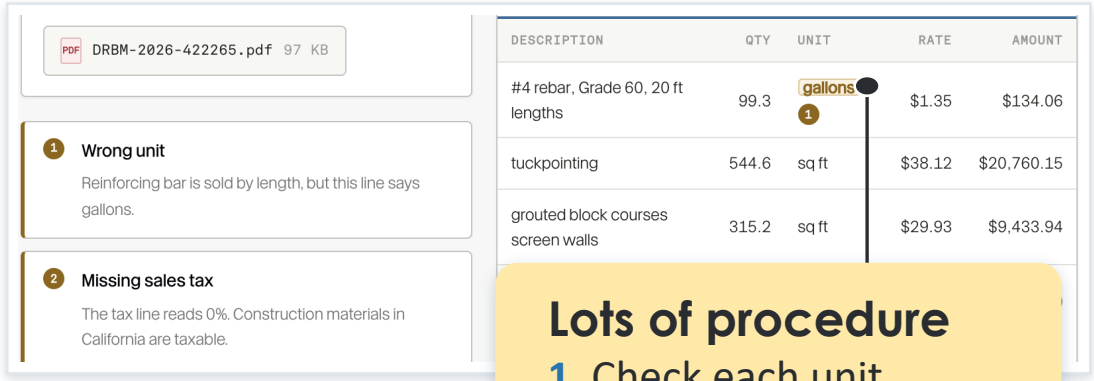
"What's **our** most recent M&A deal with a working-capital adjustment?"



Lots of data
266 matters
9,286 files
100M tokens

Source: Engram × Harvey

"Please **process** this invoice."



Lots of procedure
1 Check each unit
2 Check the tax rate
3 Then enter the bill

Source: ApprenticeBench

The Knowledge Gap

- However, deploying an LLM in a constantly evolving world requires knowledge augmentation.
- Deployed as personal assistants, an LLM needs to cater to **user background and preferences**.



"Remember **the bag of beans I got yesterday**? How can I improve my pour over?"

Personal Setup

1. Light roast, Land and Water
2. Hario V60
3. Last brew: 253.4 g in 2:43

Knowledge Augmentation

- Knowledge augmentation uses an external memory to bridge LLMs with the evolving world.

$$y_t \sim M_\theta(\cdot \mid x_t, c_t), \quad \text{where} \quad c_t = R(x_t, D_t)$$

Knowledge Augmentation

- Knowledge augmentation uses an external memory to bridge LLMs with the evolving world.

$$y_t \sim M_\theta(\cdot \mid x_t, c_t), \quad \text{where} \quad c_t = R(x_t, D_t)$$

Datastore

The private worlds


Private information


Scattered in pieces


Constantly **changing**

memory D_t

facts

Light roast, Land and Water

Hario V60

Last brew: 253.4 g in 2:43

procedure

1 Check each unit

2 Check the tax rate

3 Then enter the bill

Knowledge Augmentation

- Knowledge augmentation uses an external memory to bridge LLMs with the evolving world.

$$y_t \sim M_\theta(\cdot \mid x_t, c_t), \quad \text{where} \quad c_t = R(x_t, D_t)$$

Datastore

Retriever

request x_t

memory D_t



Home

"Remember **the bag of beans I got yesterday**? How can I improve my pour over?"

facts

Light roast, Land and Water

Hario V60

Last brew: 253.4 g in 2:43

c_t



Work

"Please **process** this invoice."

procedure

1 Check each unit

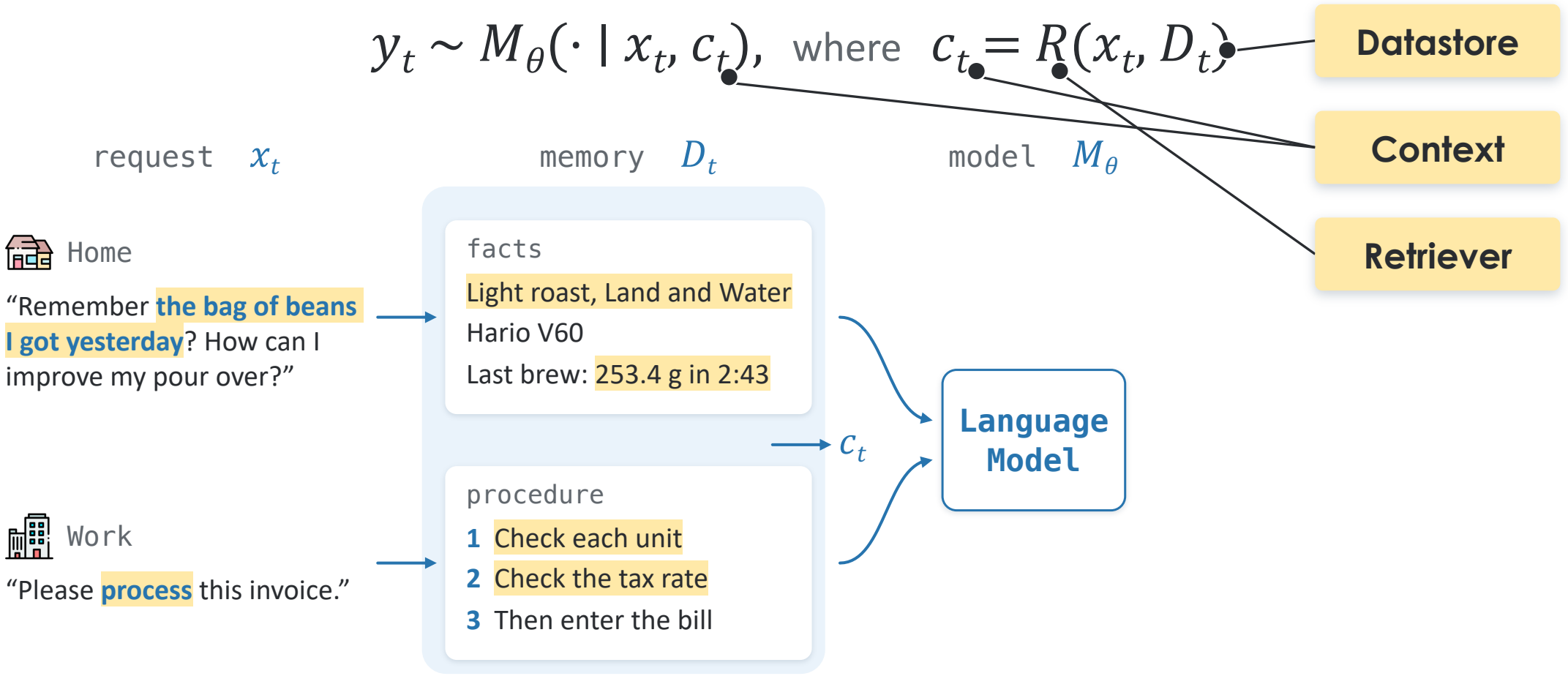
2 Check the tax rate

3 Then enter the bill

Knowledge Augmentation

- Knowledge augmentation uses an external memory to bridge LLMs with the evolving world.

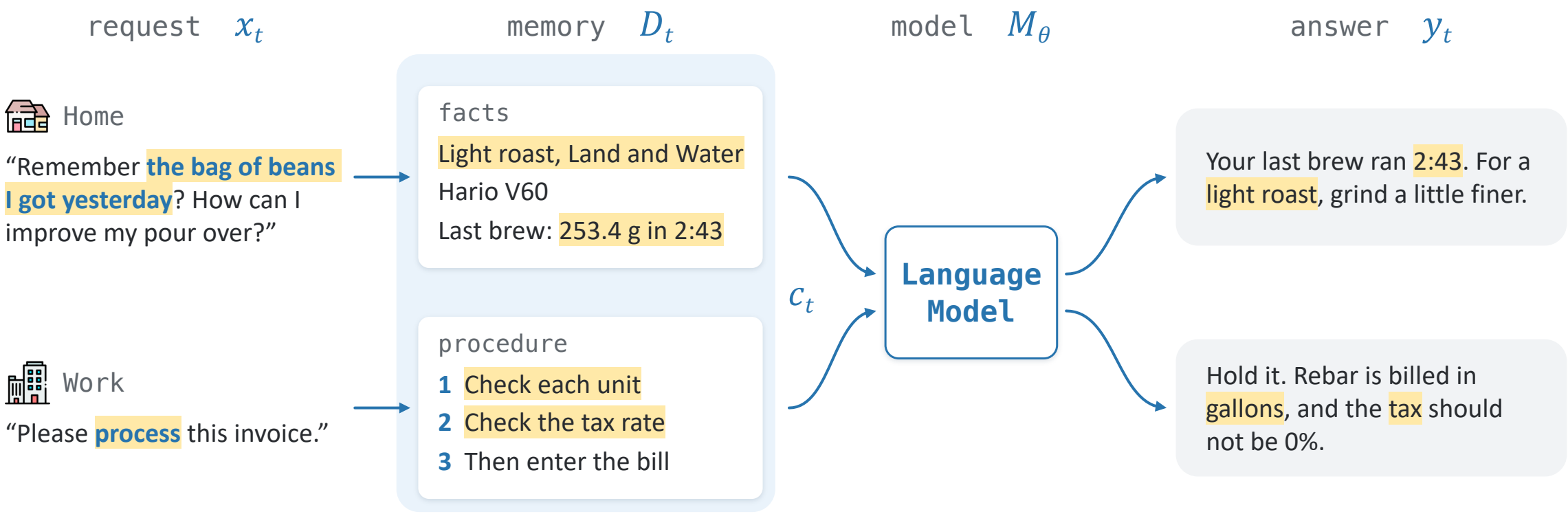
$$y_t \sim M_\theta(\cdot \mid x_t, c_t), \quad \text{where} \quad c_t = R(x_t, D_t)$$



Knowledge Augmentation

- Knowledge augmentation uses an external memory to bridge LLMs with the evolving world.

$$y_t \sim M_\theta(\cdot \mid x_t, c_t), \quad \text{where} \quad c_t = R(x_t, D_t)$$



Challenge 1: Evaluation

- Faithfully simulating real deployment is hard, and without it the gap stays hidden.

① $y_t \sim M_\theta(\cdot \mid x_t, c_t)$, where $c_t = R(x_t, D_t)$



answer y_t

"Your last brew ran 2:43.
For a light roast, grind a little finer."

Challenge 2: LLM Knowledge Boundary

- The model must know its knowledge boundary: when to look past it, and where.

2

$$y_t \sim M_\theta(\cdot \mid x_t, c_t), \quad \text{where} \quad c_t = R(x_t, D_t)$$

"Remember **the bag of beans I got yesterday**?



memory D_t

Light roast · last brew 2:43

How can I **improve my pour over**?"



Parametric knowledge M_θ

Light roasts need a finer grind.

Challenge 3: Datastore Representation

- Memory must discover, structure, and evolve reusable knowledge from raw history.

$$y_t \sim M_\theta(\cdot \mid x_t, c_t), \quad \text{where} \quad c_t = R(x_t, \underset{\textcircled{3}}{D_t})$$

raw history

user	new bag from Land and Water
agent	open invoice INV-2026-42291
user	it's a light roast, very fruity
agent	line 1: rebar, 120 gallons
review	check the units on every line
user	253.4 g in 2:43 this morning
agent	tax 0%, enter the bill
review	check the tax rate first
user	thanks, that helps

distill →

memory D_t

facts

~~Medium roast, old bag~~

Light roast, Land and Water

Last brew: 253.4 g in 2:43

procedure

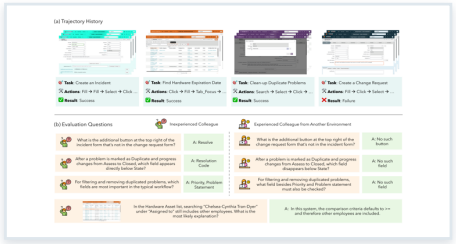
1 Check each unit

2 Check the tax rate

Thesis Statement

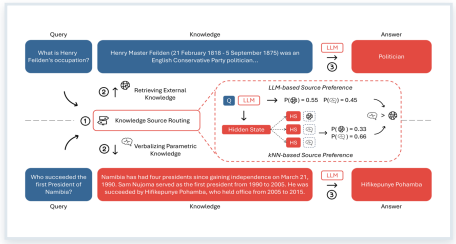
- My thesis advances knowledge-augmented LLMs by synthesizing **faithful evaluation**, **selective knowledge access**, and **continual datastore adaptation**.

Evaluation



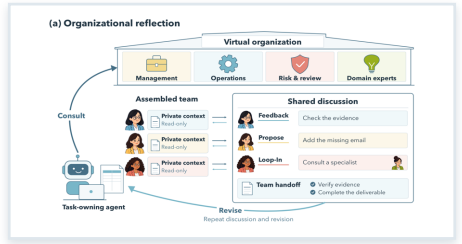
LongMemEval-V2

Model



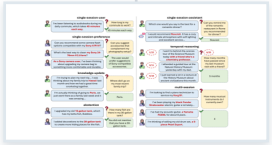
Self-Routing RAG

Datastore



ReOrg

1



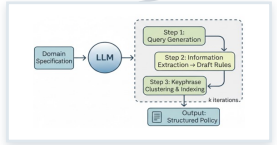
LongMemEval

2



Repoformer

3



Deep Policy Research

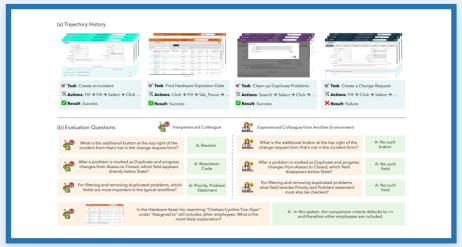


Reasoning Memory

Part I: Building Challenging Memory Evaluations

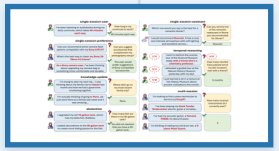
Evaluation

What must memory deliver?



LongMemEval-V2

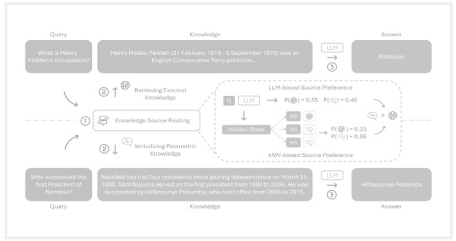
1



LongMemEval

Model

When and where to look?



Self-Routing RAG

2



Repoformer

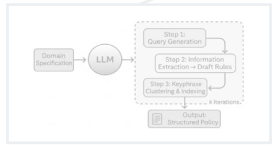
Datastore

What is worth keeping?



ReOrg

3



Deep Policy Research

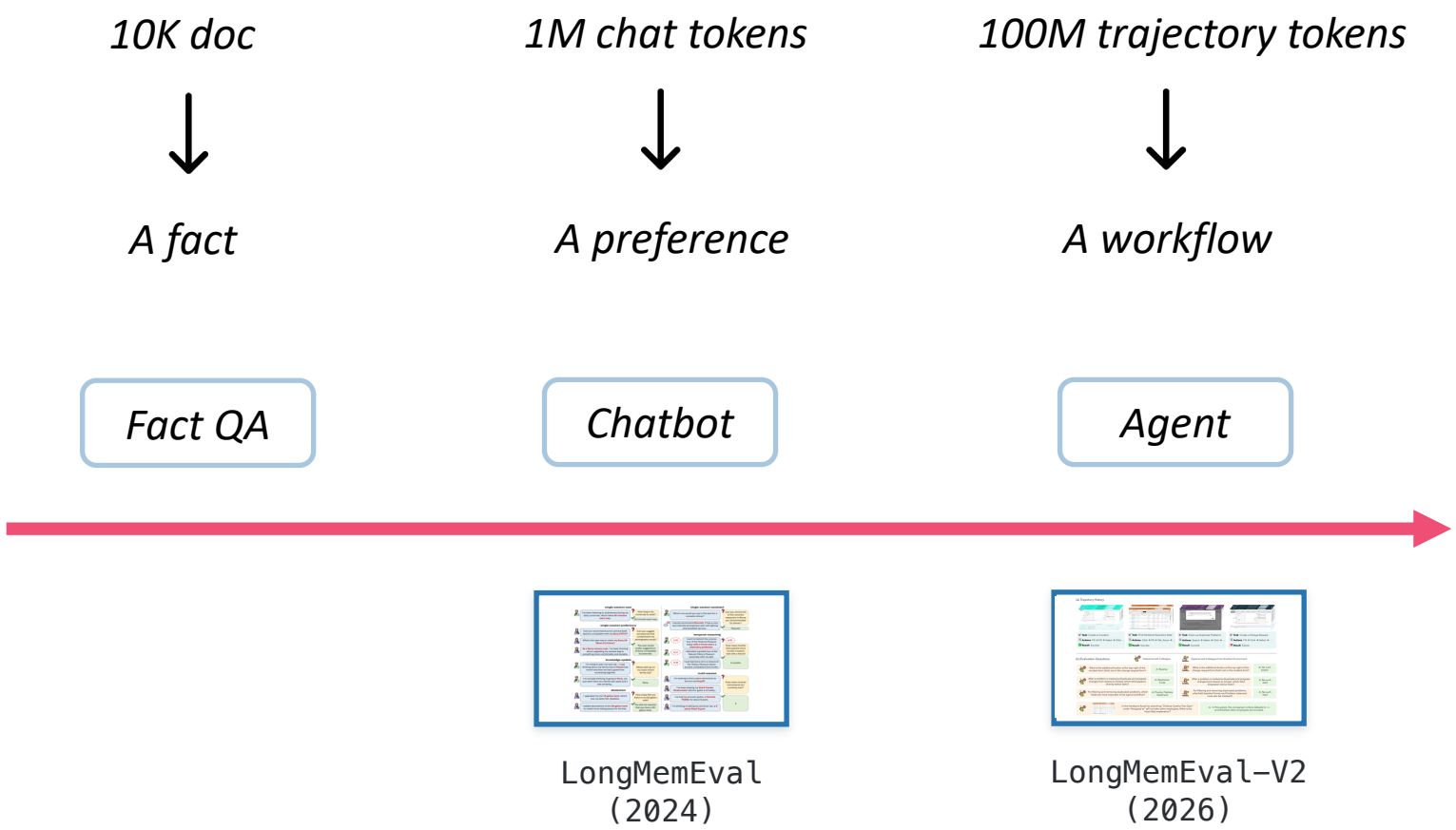


Reasoning Memory

UCLA

Evolving Evaluation with Deployment

- As LLM applications evolve, rigorously constructed novel evaluations are required to reveal the private knowledge needed at deployment.

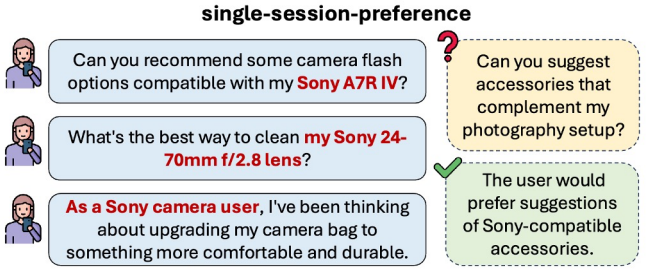


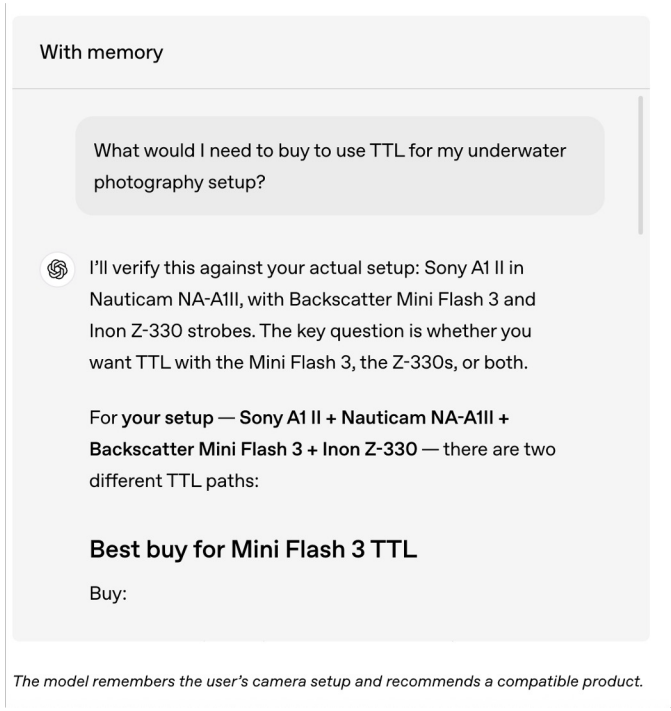
Evolving Evaluation with Deployment

- LongMemEval defined a wide memory capability matrix and revealed the limitations of frontier **chat** systems. It has been widely adopted by both the academia and the industry.

System	LLM	Accuracy
Offline Reading	GPT-4o	0.9184
ChatGPT	GPT-4o	0.5773
	GPT-4o-mini	0.7113
Coze	GPT-4o	0.3299
	GPT-3.5-turbo	0.2474

(a) Commercial memory-augmented chat assistants exhibit weak performance on LONGMEMEVAL. The accuracy of ChatGPT and Coze degrades by a large amount compared to directly reading the context ("Offline Reading") with the same LLM. Specifically, ChatGPT and Coze instantiated with GPT-4o exhibits 37% and 64% performance drop, respectively.



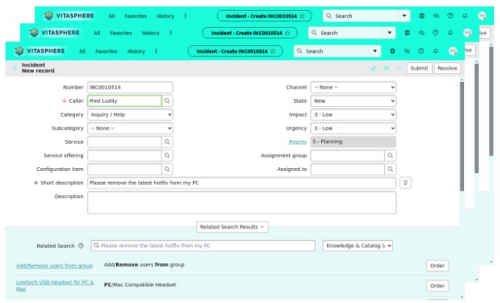


Dreaming. OpenAI, 2026.

Evolving Evaluation with Deployment

- Recently, LLM deployments have evolve into **multi-turn agentic work**.
- Correspondingly, how should memory capability be characterized?

Large Context

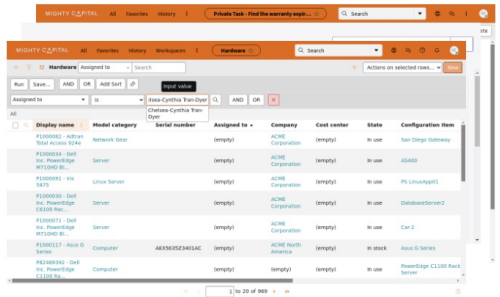


🎯 **Task**: Create an Incident

🛠️ **Actions**: Fill → Fill → Select → Click …

✅ **Result**: Success

Long Horizon

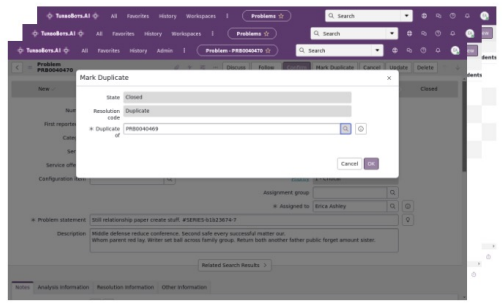


🎯 **Task**: Find Hardware Expiration Date

🛠️ **Actions**: Click → Fill → Tab_Focus → …

✅ **Result**: Success

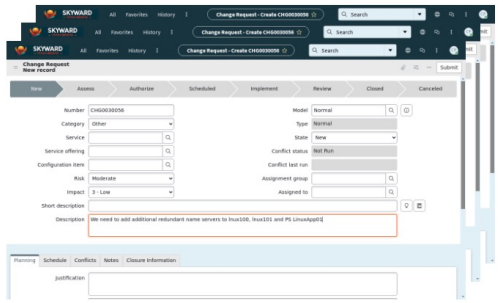
Higher-Order Knowledge



🎯 **Task**: Clean-up Duplicate Problems

🛠️ **Actions**: Search → Select → Click → …

✅ **Result**: Success



🎯 **Task**: Create a Change Request

🛠️ **Actions**: Fill → Click → Select → …

❌ **Result**: Failure

LongMemEval-V2 (LME-V2)

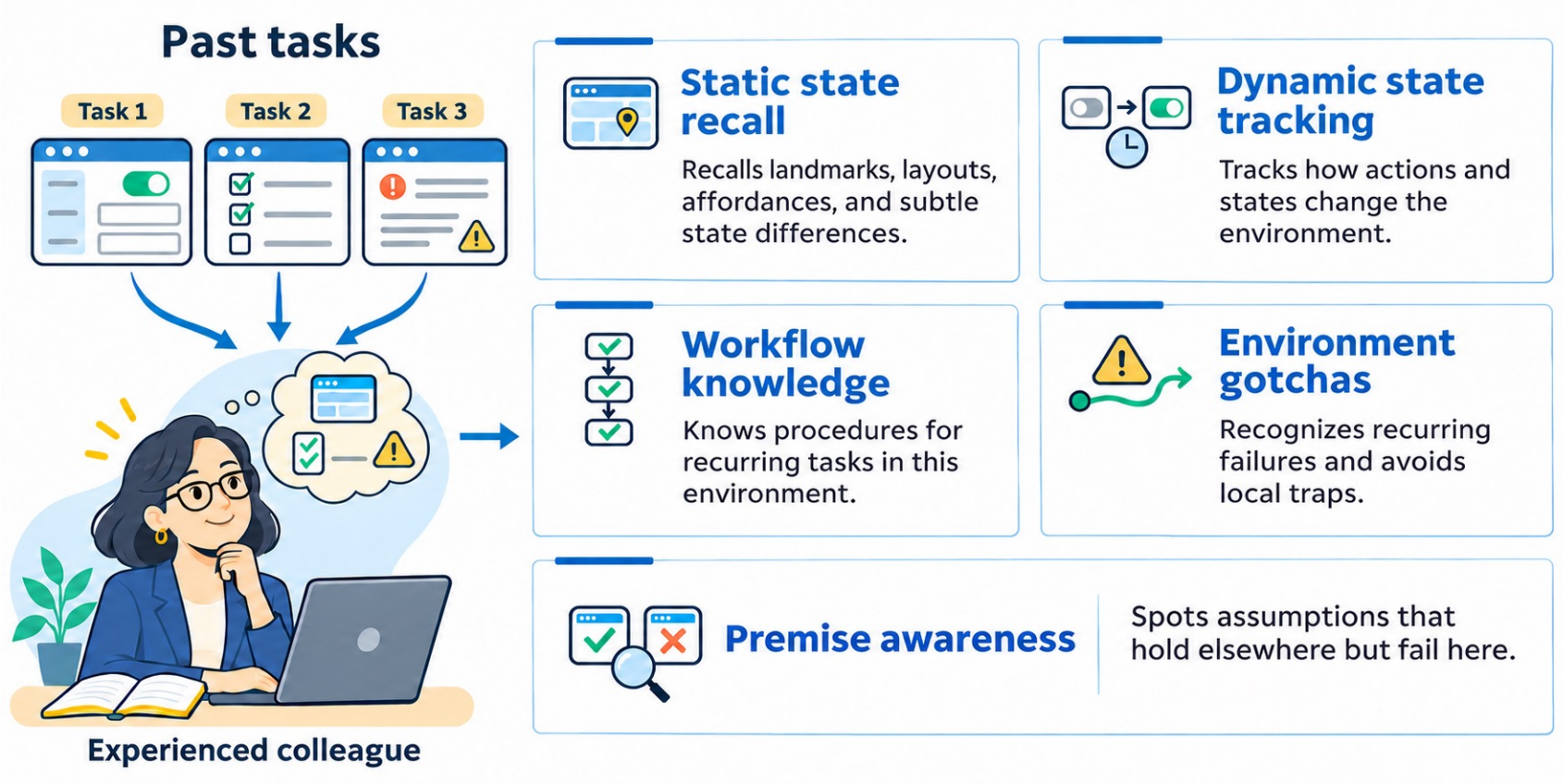
- In this work, we posit a rationale behind evaluations built for long-horizon agent memory:

"A high-quality memory makes an agent an **experienced colleague** in a specialized environment."

LongMemEval-V2: Evaluating Long-Term Agent Memory Toward Experienced Colleagues (2026)

What does an Experienced Colleague Memorize?

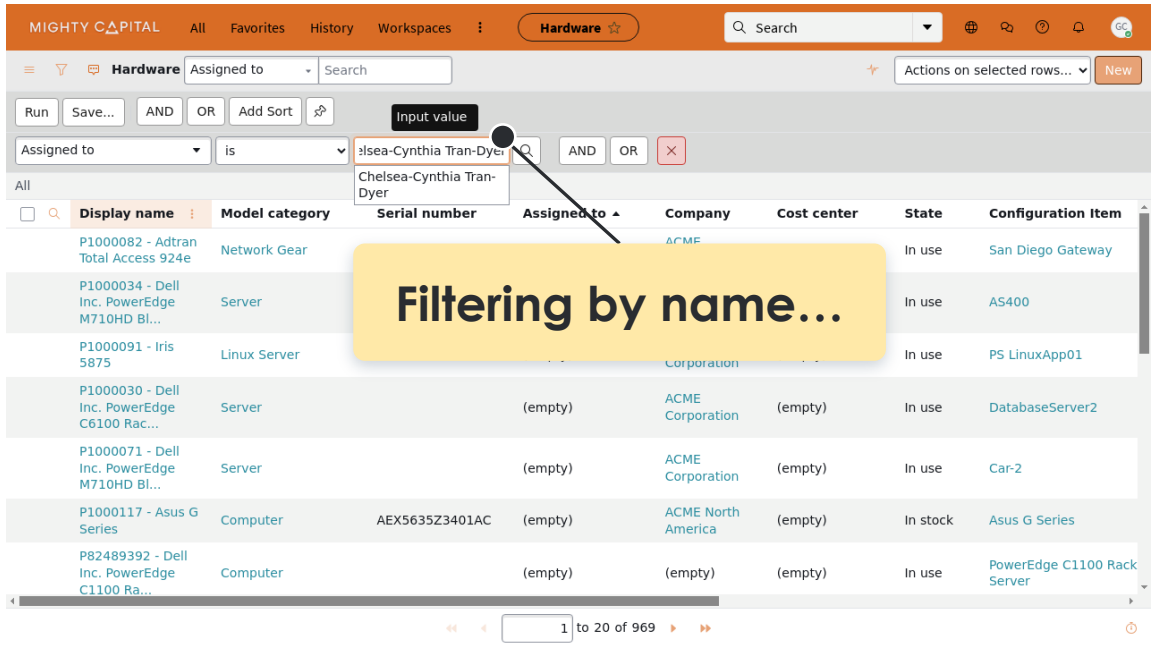
- LongMemEval-V2 defines a comprehensive memory capability matrix:

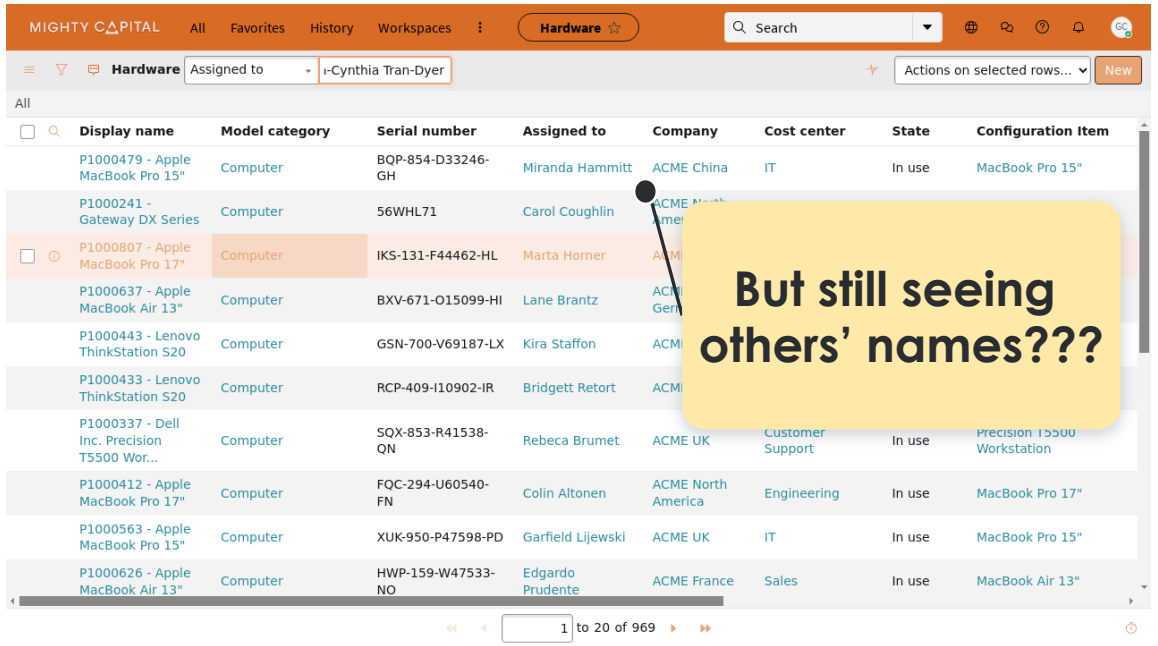


Example

A key novel ability: **Environment Gotchas**

- An experienced colleague is aware of common issues and bugs in the current environment.





Example

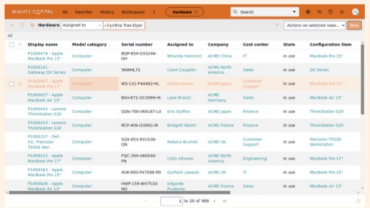
A key novel ability: **Environment Gotchas**

- An experienced colleague is aware of common issues and bugs in the current environment.



Inexperienced Colleague





In the Hardware Asset list, searching “Chelsea-Cynthia Tran-Dyer” under “Assigned to” still includes other employees. What is the most likely explanation?

Example

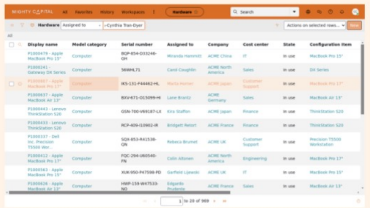
A key novel ability: **Environment Gotchas**

- An experienced colleague is aware of common issues and bugs in the current environment.



Inexperienced Colleague





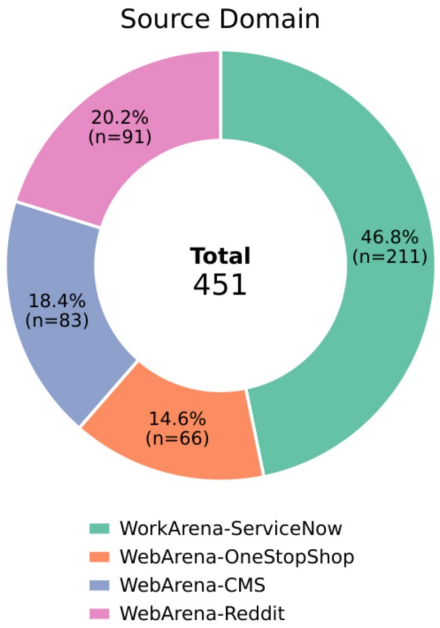
In the Hardware Asset list, searching “Chelsea-Cynthia Tran-Dyer” under “Assigned to” still includes other employees. What is the most likely explanation?

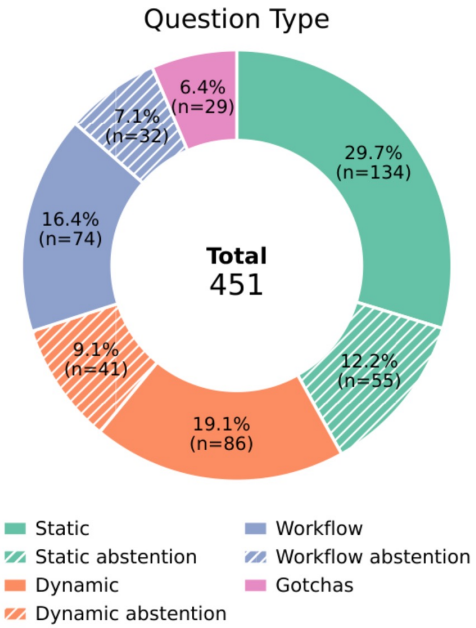
A: In this system, the comparison criteria defaults to >= and therefore other employees are included.

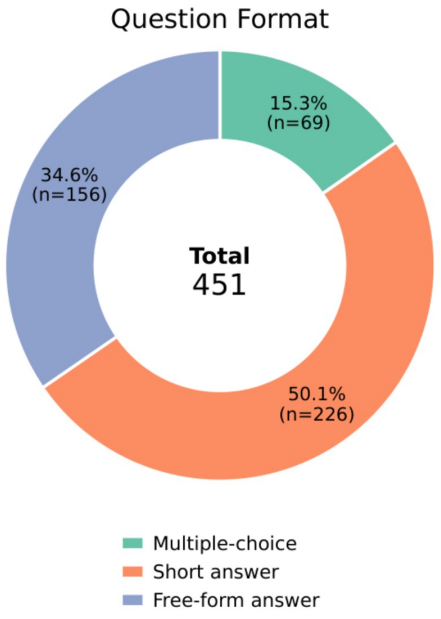
Benchmark Construction

- We design a novel framework to chain together sessions to form complex history.

1 Collect web-agent trajectories → **2 Annotate** memory questions → **3 Label** answer evidence → **4 Assemble** history haystacks





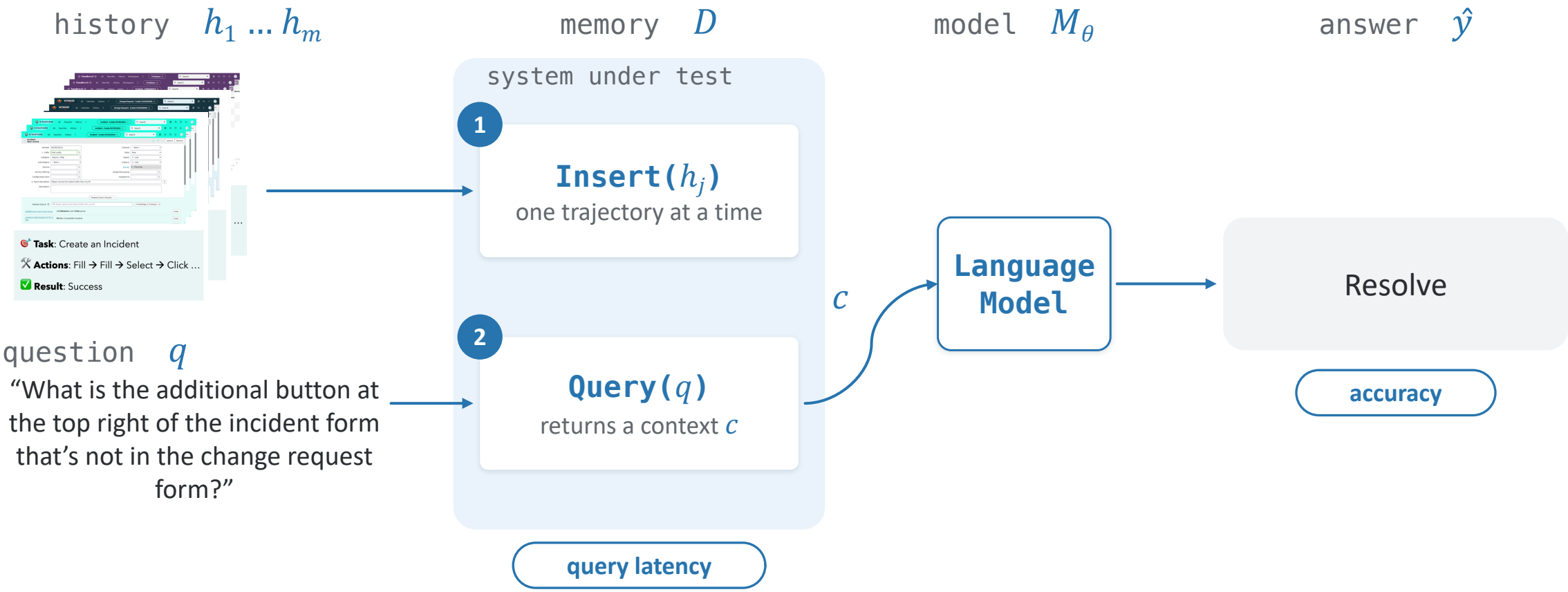


Average Haystack Cost

	Oracle	Small	Medium
Trajectories	**1.39** 1x	**100** 72x	**498** 359x
States	**39.3** 1x	**2.5K** 64x	**14.9K** 378x
Tokens	**310.8K** 1x	**25.6M** 82x	**114.8M** 370x

LME-V2: Evaluation Style

We evaluate memory systems on ***context gathering***.



Evaluation Results

- Coding agents exhibit promising long-context processing performance, but fail to be as efficient and effective compared to our full system, which augments coding agents with memory-specific manifests, tools, and workflows.

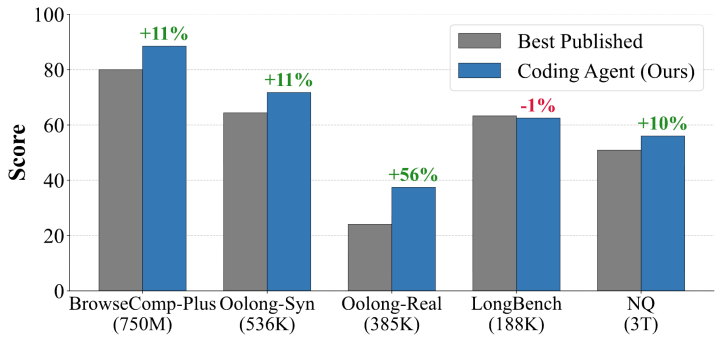


Figure 1. Coding agents significantly outperform best published results across five long-context benchmarks spanning from 188K to three trillion tokens. Green percentages indicate relative improvement over prior state-of-the-art.

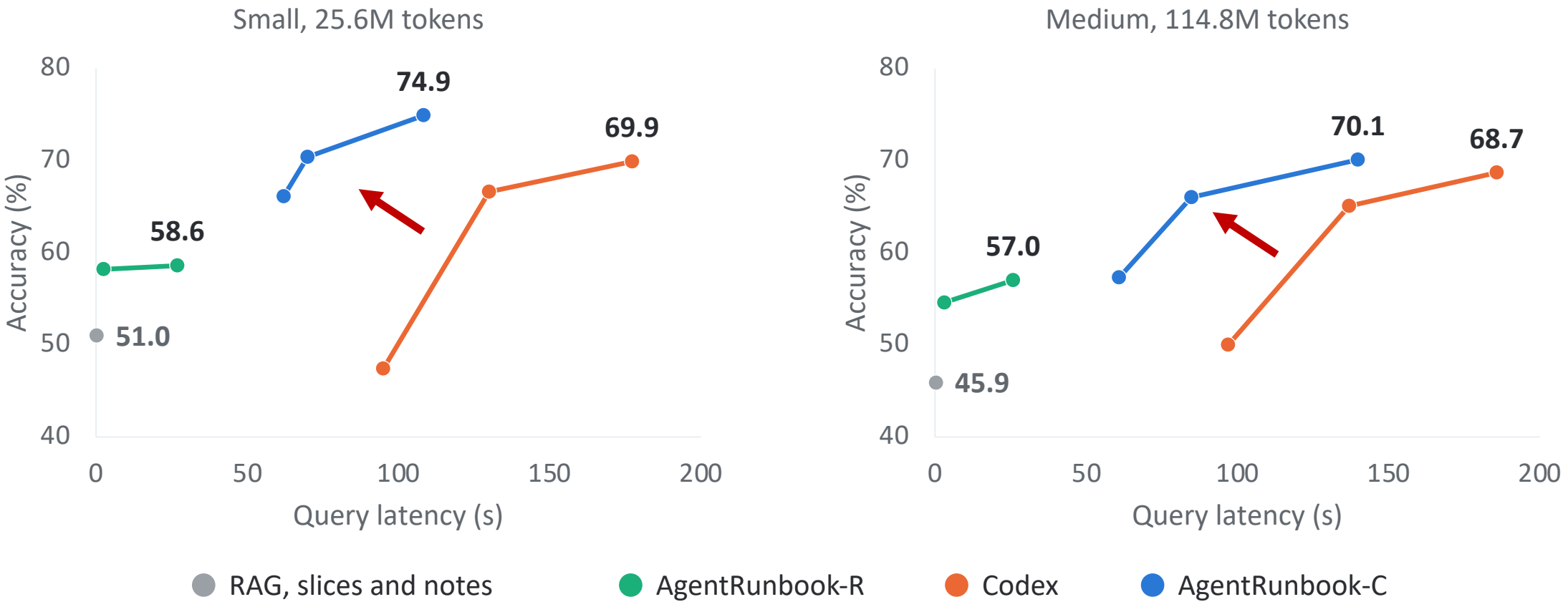
[Coding Agents are Effective Long-Context Processors. Cao et al., 2026.]

LME-V2-Small, accuracy (%) from the fixed Qwen3.5-9B reader

Method	Overall	Static	Dynamic	Workflow	Gotchas	Latency
No retrieval	1.3	0.0	0.8	9.4	13.8	0 s
Coding agents, GPT-5.4-mini controller with xhigh reasoning						
Codex	69.9	80.4	67.0	57.5	**58.6**	177.2 s
AgentRunbook-C	**74.9**	82.0	**72.4**	**72.6**	48.3	**108.3 s**
– workflow	70.1	77.2	67.7	63.2	**58.6**	167.9 s
– manifest artifacts	74.7	**84.7**	70.9	69.8	44.8	155.0 s
– helper functions	71.4	78.3	**72.4**	66.0	41.4	145.9 s

Evaluation Results

- Our system advances the accuracy-latency frontier compared to the off-the-shelf coding agents and RAG systems.



Section Summary

- As LLM applications evolve, rigorously constructed novel evaluations are required to reveal the external knowledge requirements at deployment.

- In LongMemEval and LongMemEval-V2, we establish the capability matrix for chatbot/agent memory, create challenging scenarios to show the gap of frontier systems.

- In both works, we then discuss memory methods designs and simple ways to bridge the gap.

Part II: Selective Memory Access

Evaluation

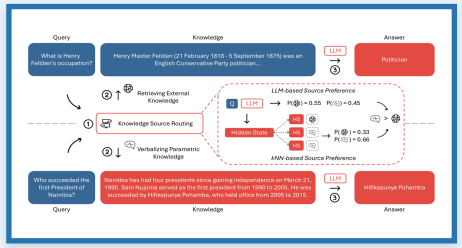
What must memory deliver?



LongMemEval-V2

Model

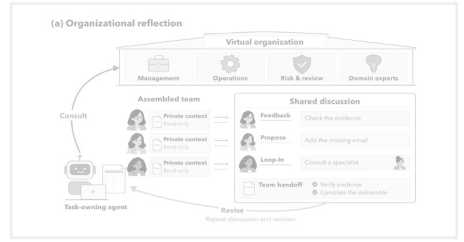
When and where to look?



Self-Routing RAG

Datastore

What is worth keeping?



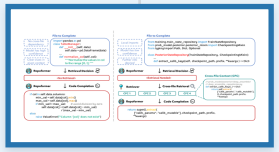
ReOrg

1



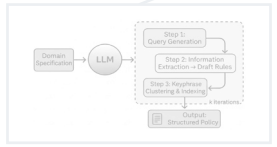
LongMemEval

2



Repoformer

3



Deep Policy Research



Reasoning Memory

UCLA

Selective Retrieval: Formulation

- The second part of my thesis work focuses on the problem of **selective retrieval**, which explicitly decides whether the external memory access is necessary.

Answer $y \sim M_\theta(\cdot \mid x, c)$

Always $c = R(x, D)$

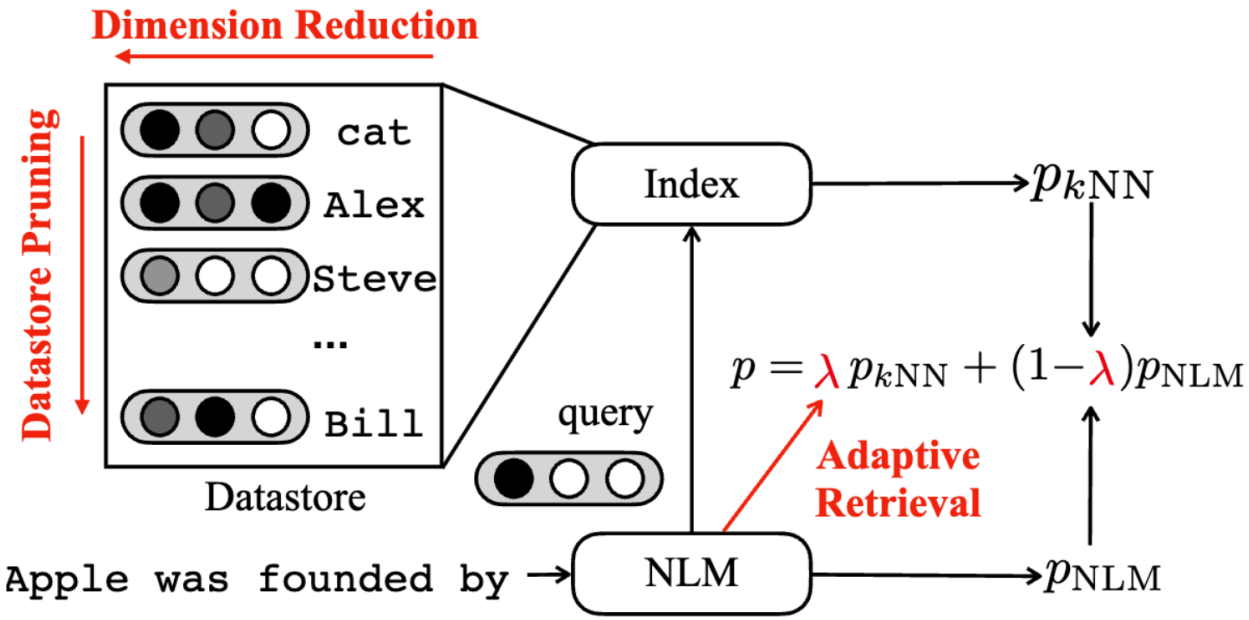
Inefficient, as a lot of queries do not require private context.

Irrelevant contexts can be counterproductive.

Selective $c = \begin{cases} R(x, D), & \text{if retrieval criteria met} \\ \emptyset, & \text{otherwise} \end{cases}$

Selective Retrieval: The Starting Point

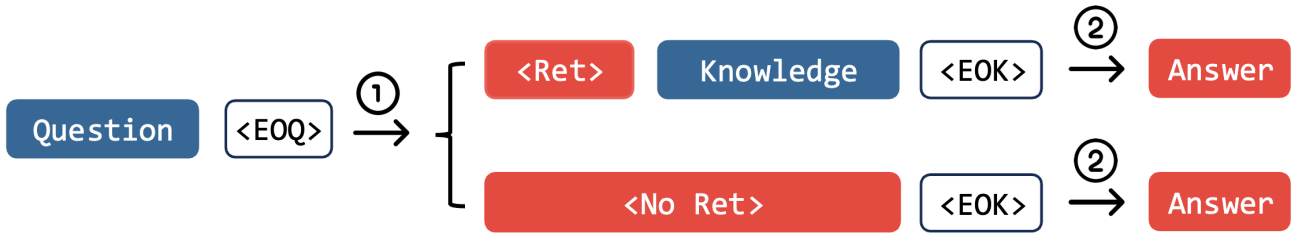
- The idea originated in 2021, shortly after the kNN-LM paper.
- Learning to turn off nonparametric contributions via language modeling.

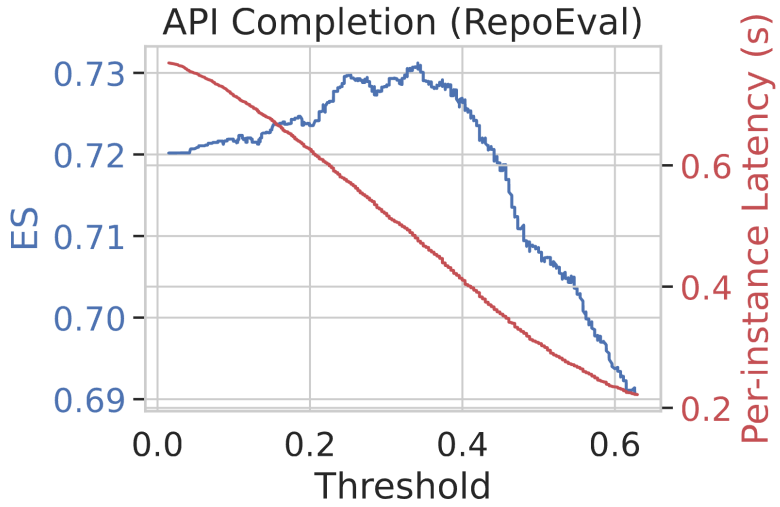


Efficient Nearest Neighbor Language Models. He et al. 2021.

Selective Retrieval: a Potent Baseline

- From 2022 to 2024: different attempts to define when to retrieve.
- In my 2024 work, we introduced performance-based self-reflection for selective retrieval, applied to repository-level code generation.



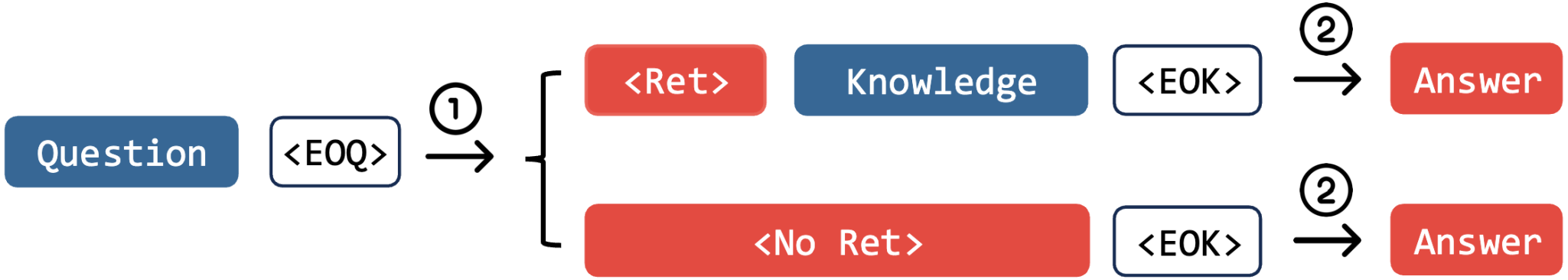


First self-selective retrieval framework to improve performance + latency.

Repoformer. Wu et al. 2024.

Selective Retrieval: Challenges

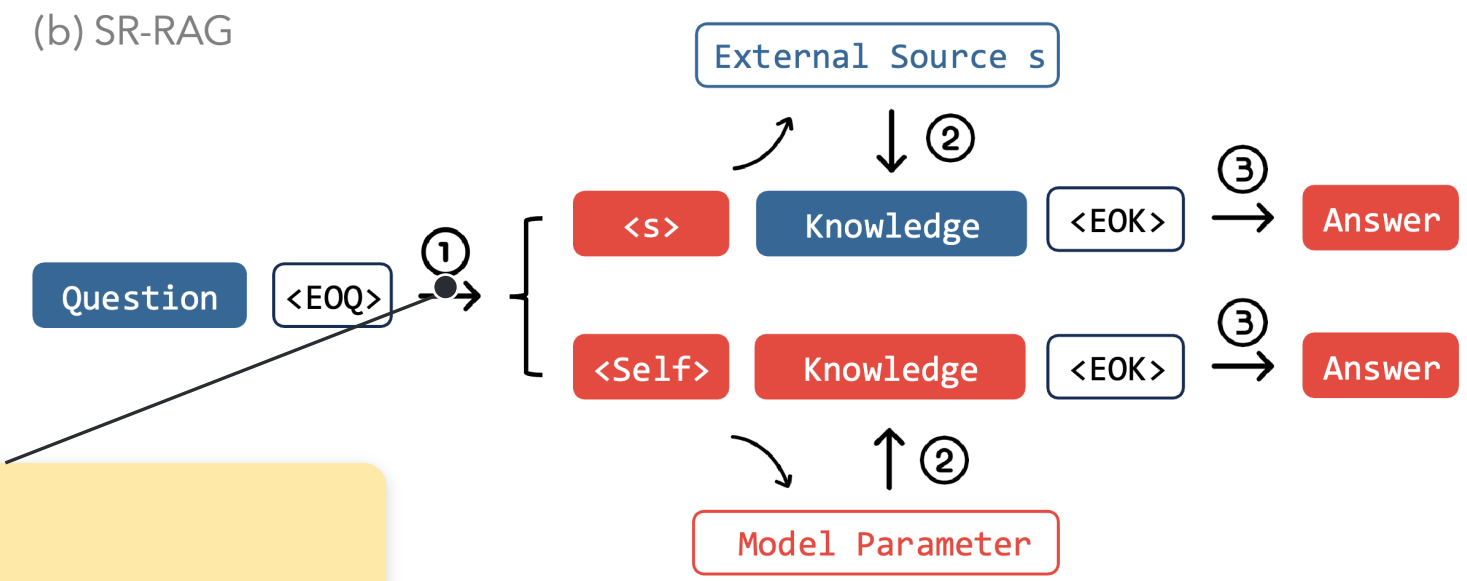
- A few questions emerge when reflecting on Repoformer's recipe…



- How to model routing across **multiple knowledge sources**?
- How to inject **domain-specific or task-specific** knowledge preferences?

Selective Retrieval, Reformulated

- We reformulate selective retrieval as knowledge source routing between LLM (~internal knowledge) and one or more external knowledge sources.



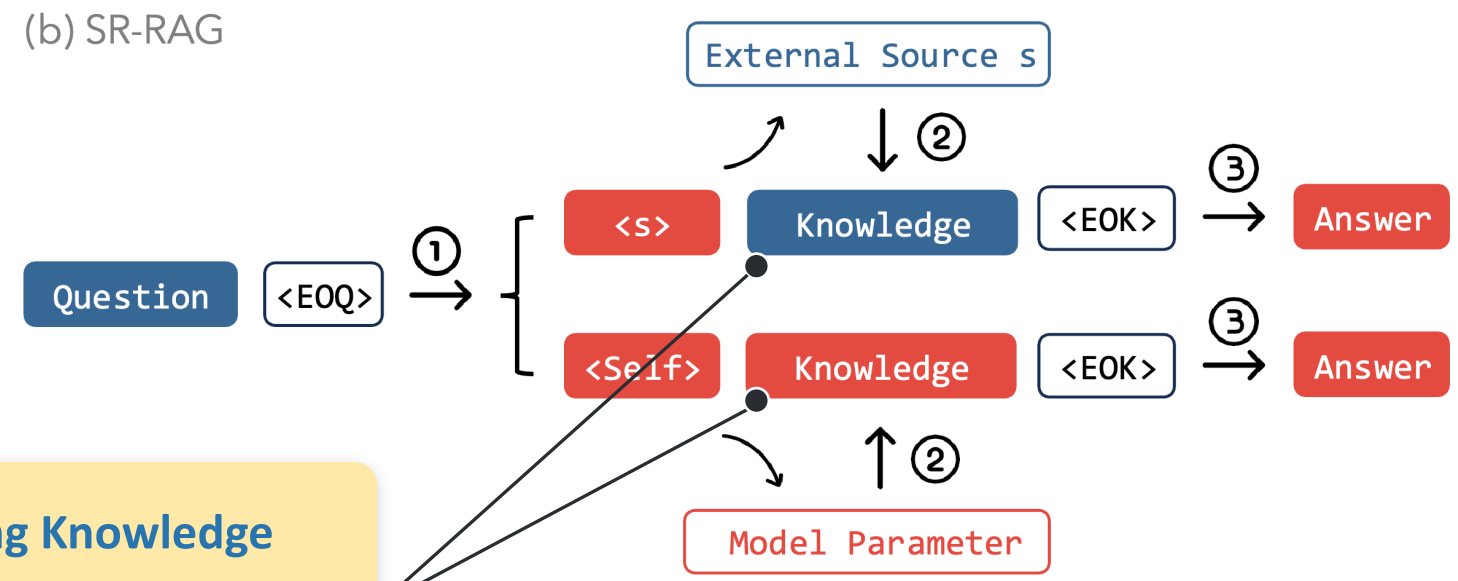
Step 1: Routing

$$\hat{s} = \underset{s \in \{\text{self}, D_1, \ldots, D_k\}}{\arg\max} \; p_\theta(\texttt{<s>} \mid q)$$

Self-Routing RAG. Wu et al. 2025.

Selective Retrieval, Reformulated

- We reformulate selective retrieval as knowledge source routing between LLM (~internal knowledge) and one or more external knowledge sources.



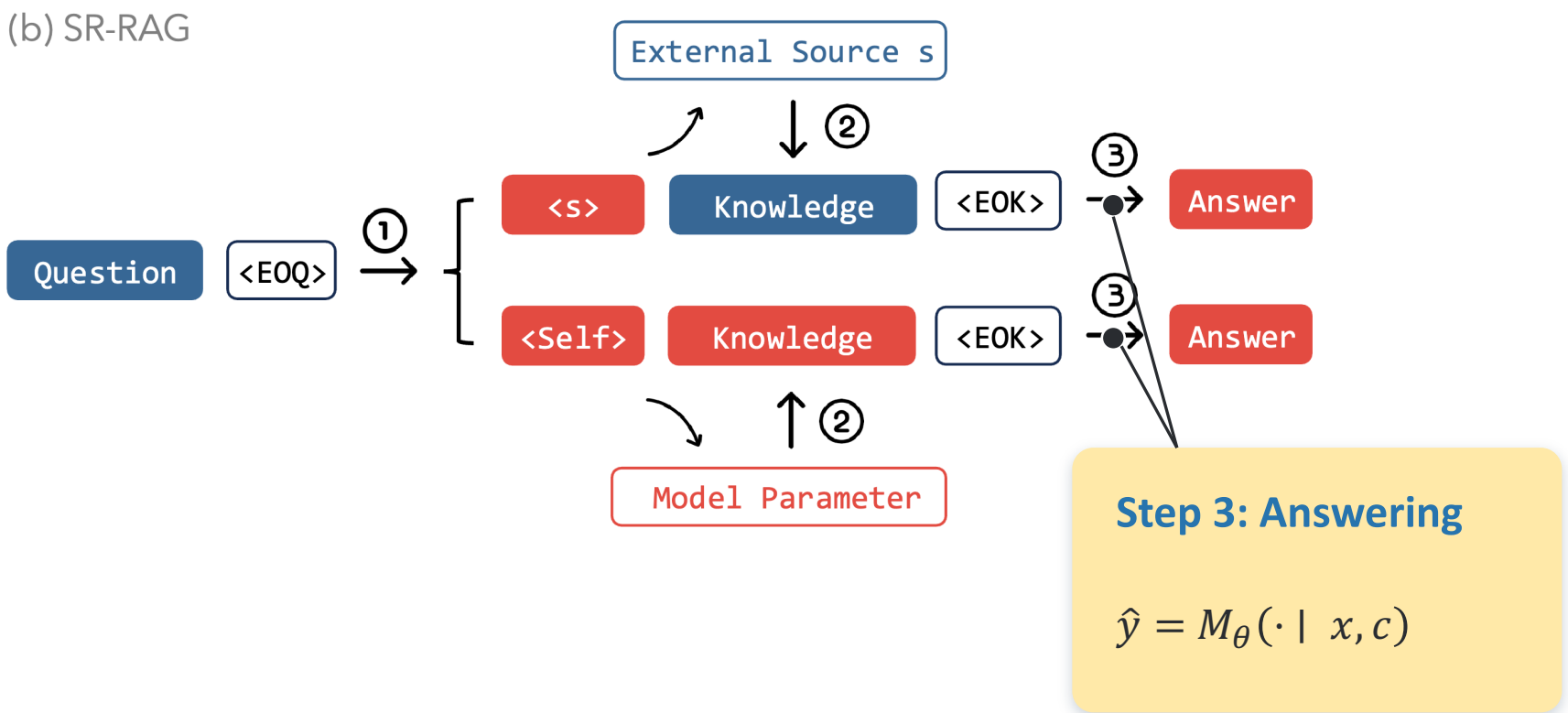
Step 2: Gathering Knowledge

$$c = \begin{cases} M_\theta(\cdot \mid q), & \text{if } \hat{s} = \text{self} \\ R(q, D_{\hat{s}}), & \text{otherwise} \end{cases}$$

Self-Routing RAG. Wu et al. 2025.

Selective Retrieval, Reformulated

- We reformulate selective retrieval as knowledge source routing between LLM (~internal knowledge) and one or more external knowledge sources.



Self-Routing RAG. Wu et al. 2025.

SR-RAG: Multi-task Training Objective

- We label each question by which source makes the gold answer most likely, then train routing, verbalization, and answering together.

Rollout	Five verbalizations from each knowledge source, scored by the gold answer's likelihood
Label	$s^\star$ = the source whose contexts rank highest by $\log p_\theta(a \mid q, c_j)$
Stage 1	$\mathcal{L} = \underbrace{-\log p_\theta(<s^\star> \mid q)}_{\text{route}} \quad \underbrace{-\log p_\theta(c_{i+} \mid q)}_{\text{verbalize, if } s^\star = \text{self}} \quad \underbrace{-\log p_\theta(a \mid q, c_{s^\star +})}_{\text{answer}}$
Stage 2	replace the verbalize term with DPO on $(c_{i+} \succ c_{i-})$

SR-RAG: Inference-Time Control

An interesting property of the reflection token emerges…

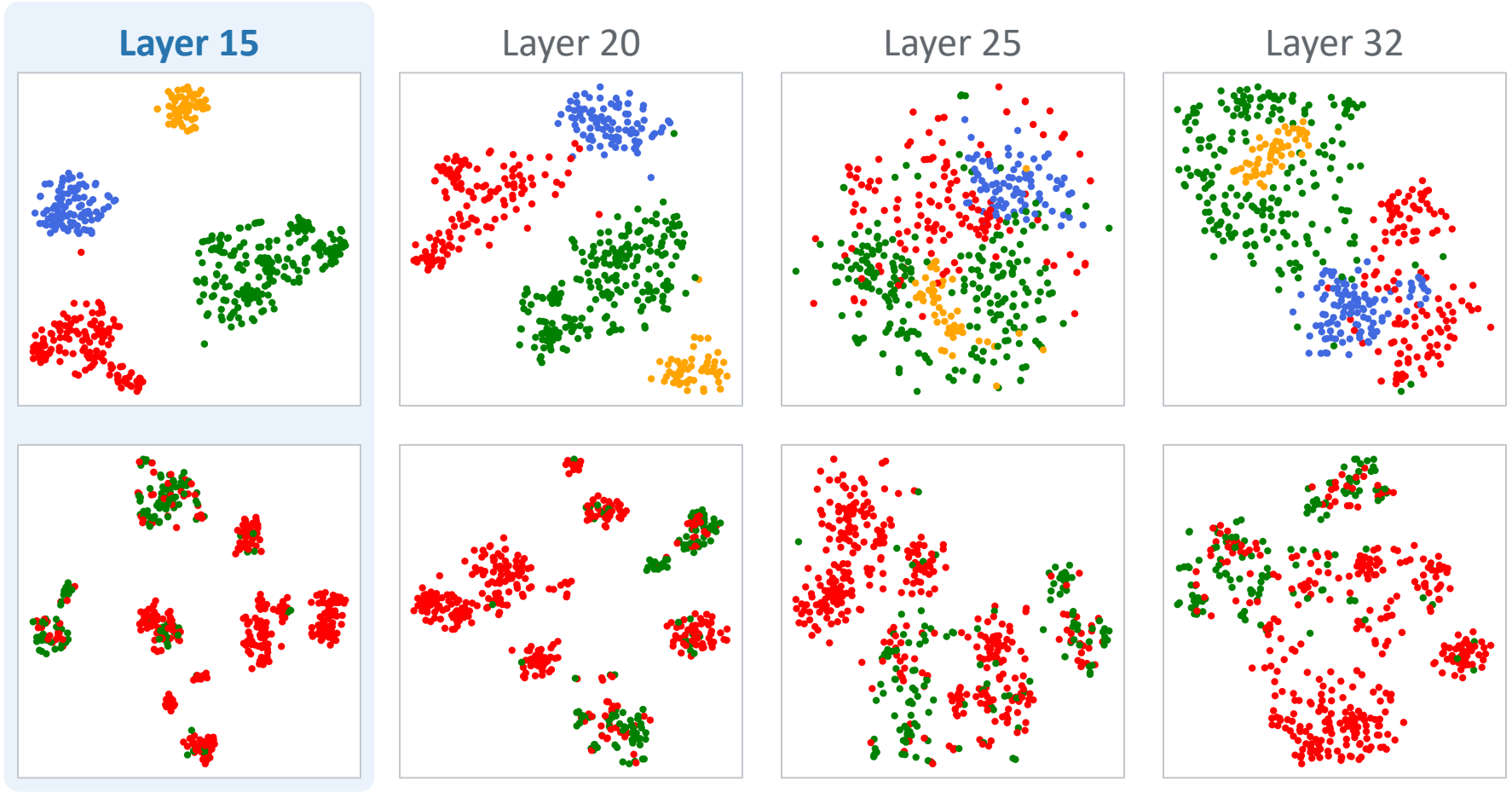
Llama-2-7B

Task type

- Fact checking
- Closed set
- Short form
- Long form

Verbalized knowledge

- Helpful
- Unhelpful



SR-RAG: Inference-Time Control

- At inference time, we further calibrate knowledge source selection by combining the model's source probability with votes from its neighbors in the hidden representation space.

$$\text{Retrieve from s} \sim p_\theta(\text{<s>} \mid q) \times p_D(\text{<s>} \mid q)$$

$$p_D(\text{<s>} \mid q) = \frac{1}{k} \sum_{j \in \text{kNN}(h_q)} \mathbf{1}[s_j = s]$$

💡 Policy Recalibration & Easy Domain Adaptation

SR-RAG: Experiments

- SR-RAG: better performance with less retrieval budget.

Training	Inference	PopQA		TriviaQA		PubHealth		ARC		Average	
		ACC	%RAG	ACC	%RAG	ACC	%RAG	ACC	%RAG	ACC	%RAG
No Fine-tuning	Always RAG	0.529	100%	0.641	100%	0.457	100%	0.546	100%	0.543	100%
	GenRead	0.247	0%	0.616	0%	0.515	0%	0.605	0%	0.496	0%
Selective RAG	Always RAG	0.567	100%	0.640	100%	0.588	100%	0.588	100%	0.596	100%
	Selective RAG	0.565	98%	0.638	100%	0.589	100%	0.594	65%	0.597	86%
SR-RAG	Always RAG	**0.568**	100%	**0.669**	100%	0.689	100%	0.608	100%	0.634	100%
	SR-RAG	**0.568**	97%	0.654	75%	**0.720**	45%	**0.625**	45%	**0.642**	66%
Ablation Studies											
SR-RAG	SR-RAG w/o. kNN	0.558	94%	0.658	77%	0.694	72%	0.627	56%	0.634	75%
SR-RAG w/o. kv. label	SR-RAG	0.568	100%	0.644	100%	0.598	100%	0.629	84%	0.610	96%
SR-RAG w/o. $\mathcal{L}_{verb}^{DPO}$	SR-RAG	0.564	98%	0.645	100%	0.674	100%	0.581	66%	0.616	86%

Section Summary

- LLMs need to know their knowledge boundary to decide when and where to retrieve.

- In Repoformer and Self-Routing RAG, we learn **selective retrieval** from the model's own performance, generalize it to routing among knowledge sources including the model itself, and steer the routing at inference time with a nonparametric policy datastore.

- As a result, we are able to improve the **accuracy and efficiency** of knowledge-augmented LLM systems at the same time.

Part III: Continually Curating Datastores

Evaluation

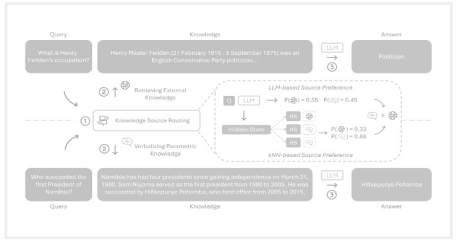
What must memory deliver?



LongMemEval-V2

Model

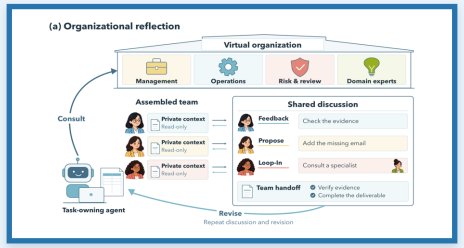
When and where to look?



Self-Routing RAG

Datastore

What is worth keeping?



ReOrg

1



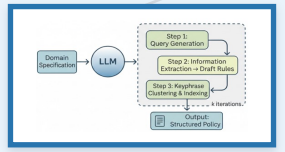
LongMemEval

2

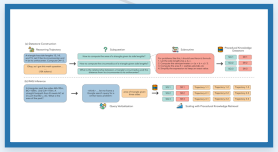


Repoformer

3



Deep Policy Research



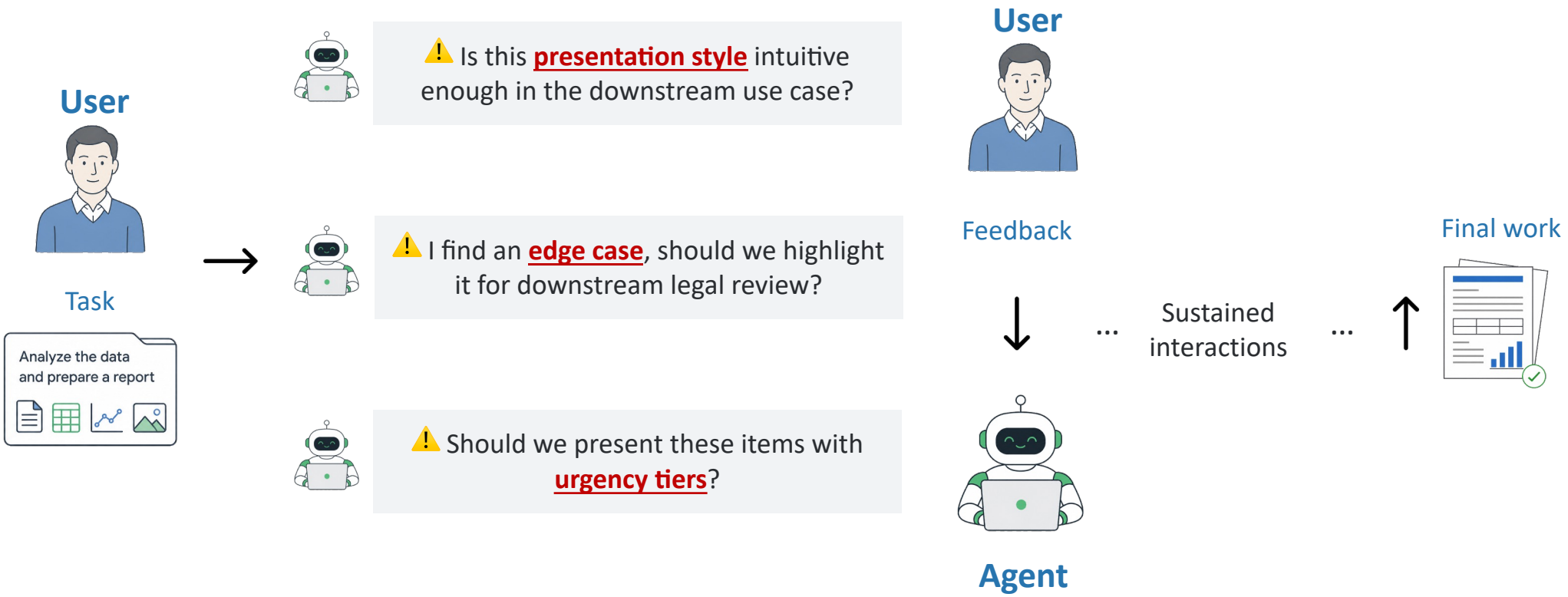
Reasoning Memory

Datastore Construction

- The last part of my thesis focuses on the **memory** itself.
- As LLM capabilities evolve, facts, procedures, and skills are gradually absorbed into the model and external datastore designs converged to simpler forms.
- **What is the next type of knowledge**, and how should we represent it?
- My latest work explores a higher order form: **organizational knowledge**.

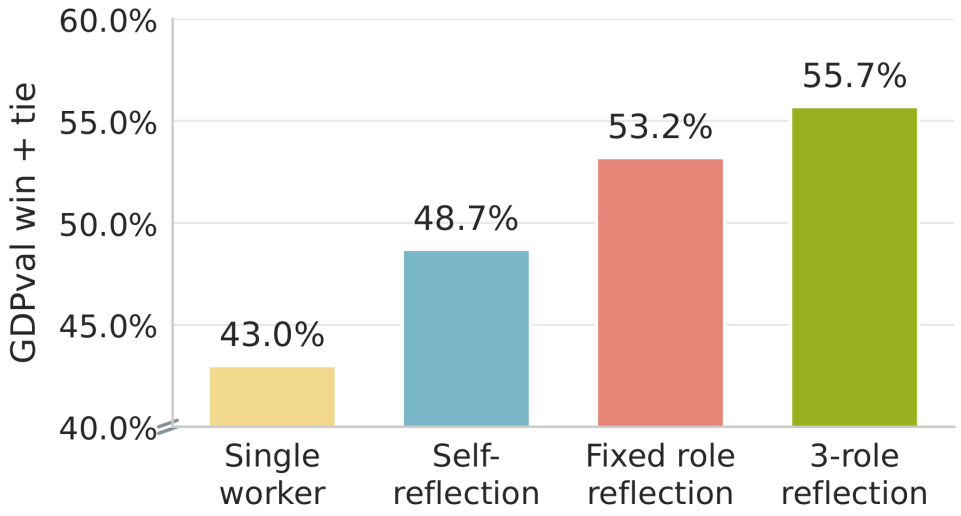
ReOrg: Motivation

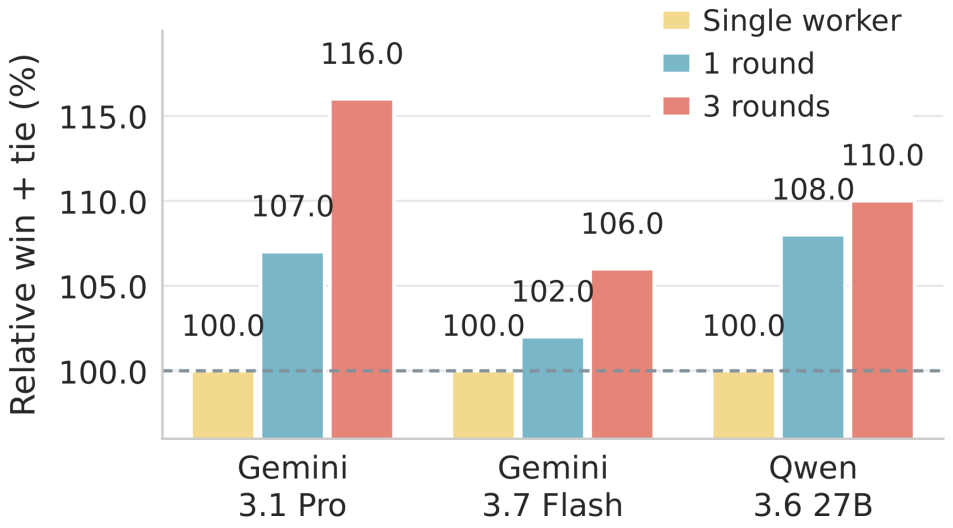
- Coding agents can carry out long-horizon knowledge work, yet they still rely on user input to fill the gaps that emerge on the way to a usable deliverable.



ReOrg: Pilot Study

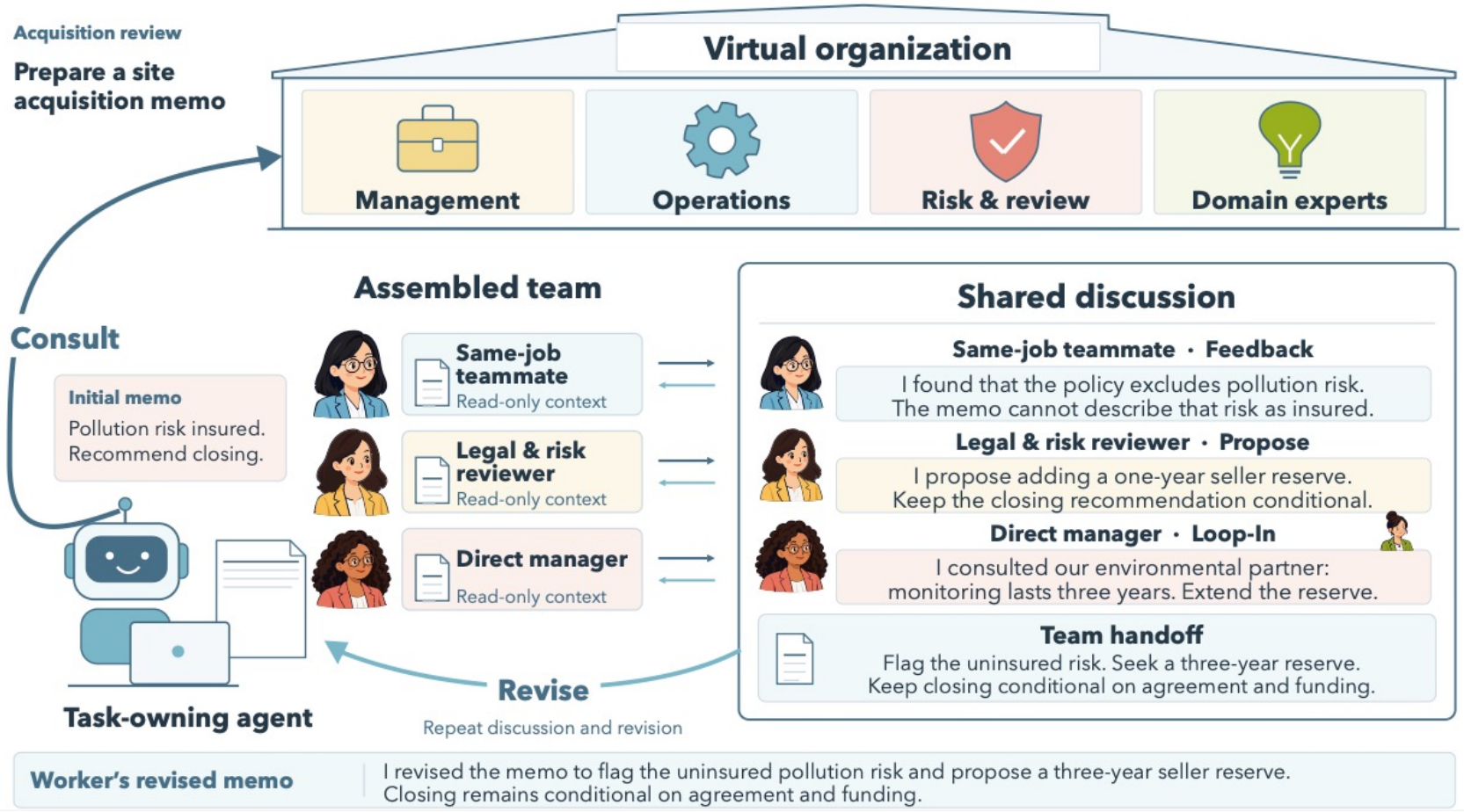
- Agents do not need to wait for the human user to give feedback.
  - In a pilot on GDPval, feedback from simulated stakeholders improves artifact quality.
  - **More perspectives** and **more rounds of feedback** lead to larger gains.





ReOrg: Method Overview

- In ReOrg, we place long-horizon agents in virtual organizations, where we simulate a team of stakeholders to collectively reflect and feedback over multiple rounds.



ReOrg: Representing Organizational Knowledge

- We package collaboration patterns into **team reflection skills**, which defines contribution type, contribution order, and how the contexts are propagated.

`role`

Who contributes?

Reviewer, requester, manager

`schema`

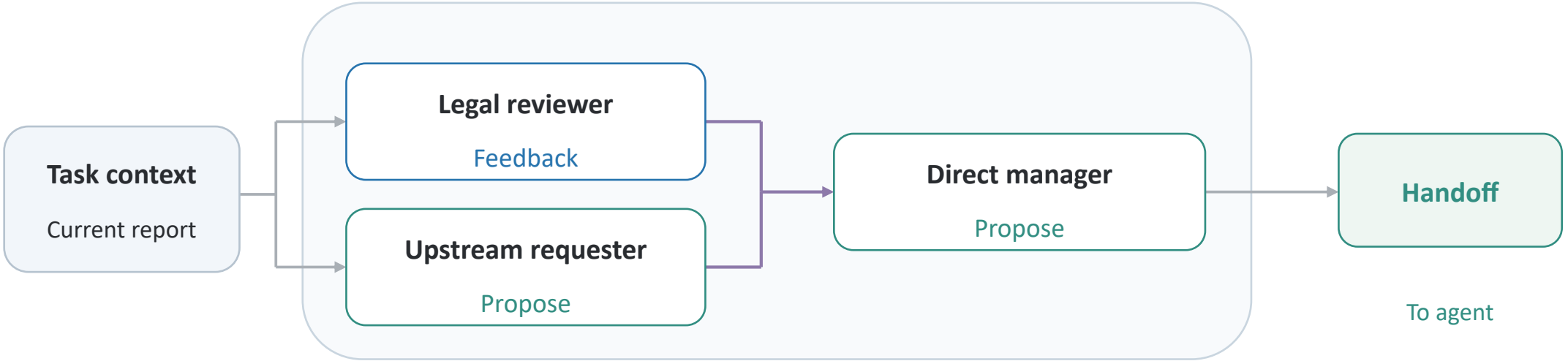
How do they communicate?

Feedback · Propose · Loop-In

`directed edge`

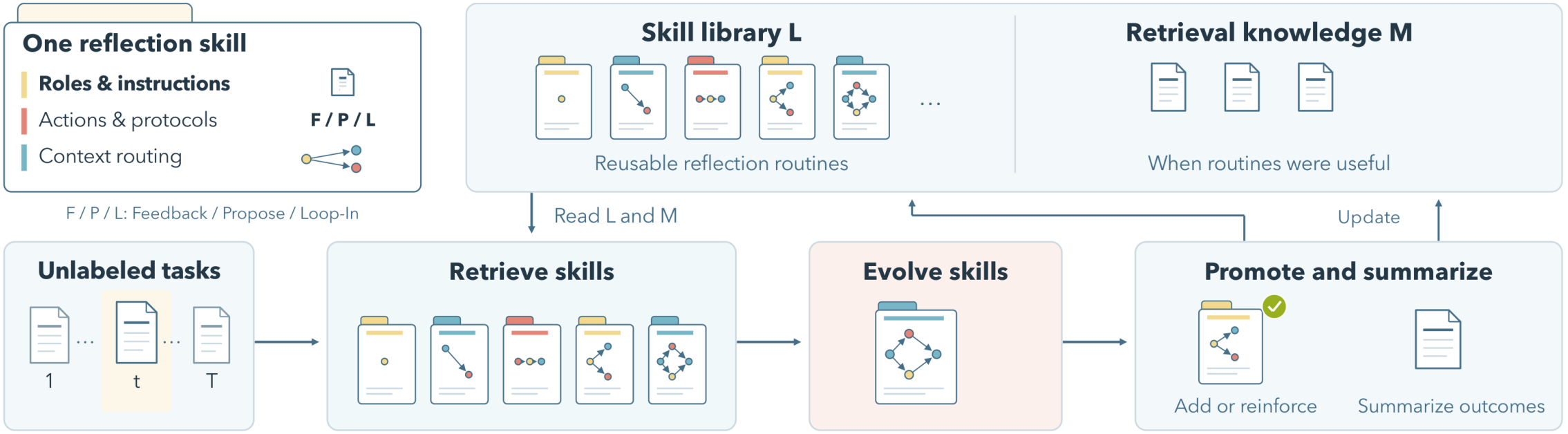
Whose output is passed on?

A → B passes A's output to B



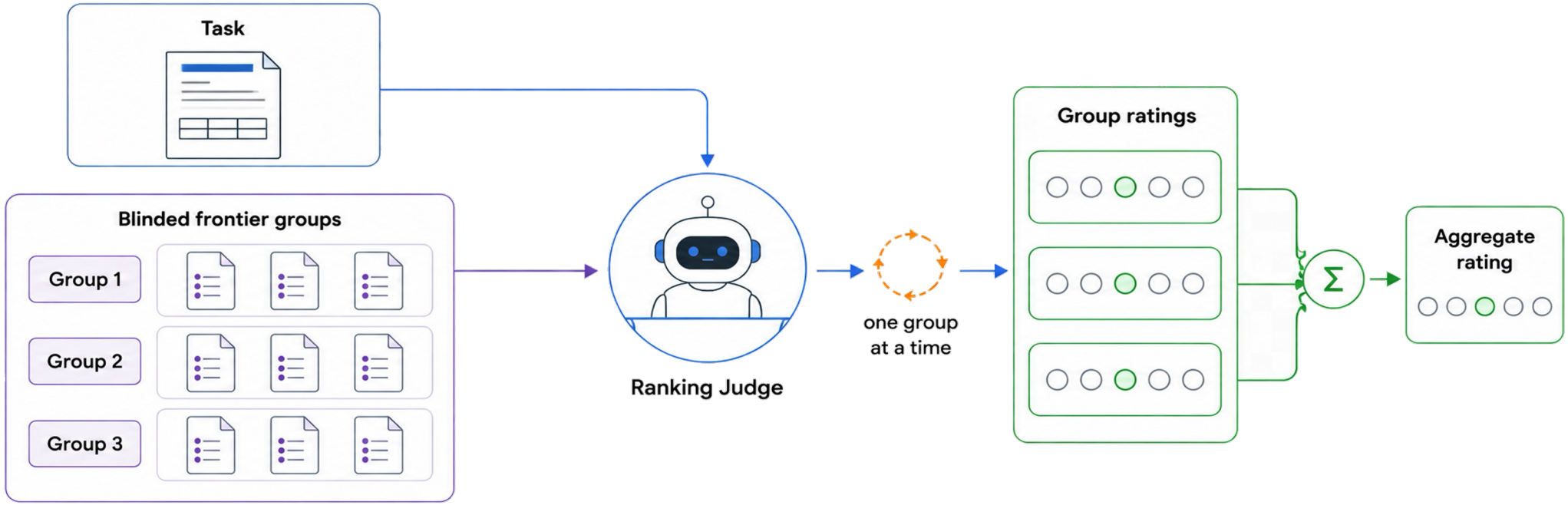
ReOrg: Accumulating Organizational Knowledge

- We design an evolutionary algorithm to evolve a skill library with a task stream.
- We **mutate** existing skills, **run** them on each task, and let an **unsupervised** judge **rank** the resulting artifacts.



ReOrg: Accumulating Organizational Knowledge

- We design an evolutionary algorithm to evolve a skill library with a task stream.
- An **unsupervised ranking judge** is the core of our skill discovery algorithm.



ReOrg: Accumulating Organizational Knowledge

- We design an evolutionary algorithm to evolve a skill library with a task stream.
- Algorithm overview

Algorithm 8.1 Continual learning of organizational reflection skills

Require: adaptation tasks $x_{1:T}$, initial library $\mathcal{L}_0$, planner knowledge M_0

for $t = 1, \ldots, T$ **do**
  $B \leftarrow \text{Retrieve}(x_t, \mathcal{L}_{t-1}, M_{t-1})$
  Execute skills in B and rank their resulting artifacts
  for each search step **do**
    $C \leftarrow \text{Mutate}(B)$
    Execute C and rank artifacts produced by $B \cup C$
    $B \leftarrow$ highest-ranked skills
  end for
  $\mathcal{L}_t \leftarrow \text{Promote}(\mathcal{L}_{t-1}, B)$
  $M_t \leftarrow M_{t-1} \cup \text{Summarize}(x_t, \text{search outcomes})$
end for
return frozen $(\mathcal{L}_T, M_T)$

ReOrg: Main Results

- We adapt our system using only 20 GDPval tasks without label supervision.
- ReOrg gives the best in-domain artifact quality and also improves work in two other knowledge-work domains.

	Gemini 3.1 Pro			Qwen 3.6 27B		
Method	GDPval	JobBench	LAB	GDPval	JobBench	LAB
Single worker	51.4	28.7	50.9	57.3	45.5	68.7
Self-reflection	53.5	29.8	59.3	64.3	45.5	74.9
Fixed manager	52.4	30.0	59.6	61.6	46.5	76.0
Random organization	53.5	30.0	**62.0**	64.3	45.7	75.2
ReOrg, 1 round	60.0	30.6	61.9	68.7	46.2	76.6
ReOrg, up to 3 rounds	**60.5**	**31.0**	**62.0**	**70.3**	**47.7**	**76.9**

Percent, by each benchmark's own metric. Skills are learned on 20 unlabeled GDPval tasks only.

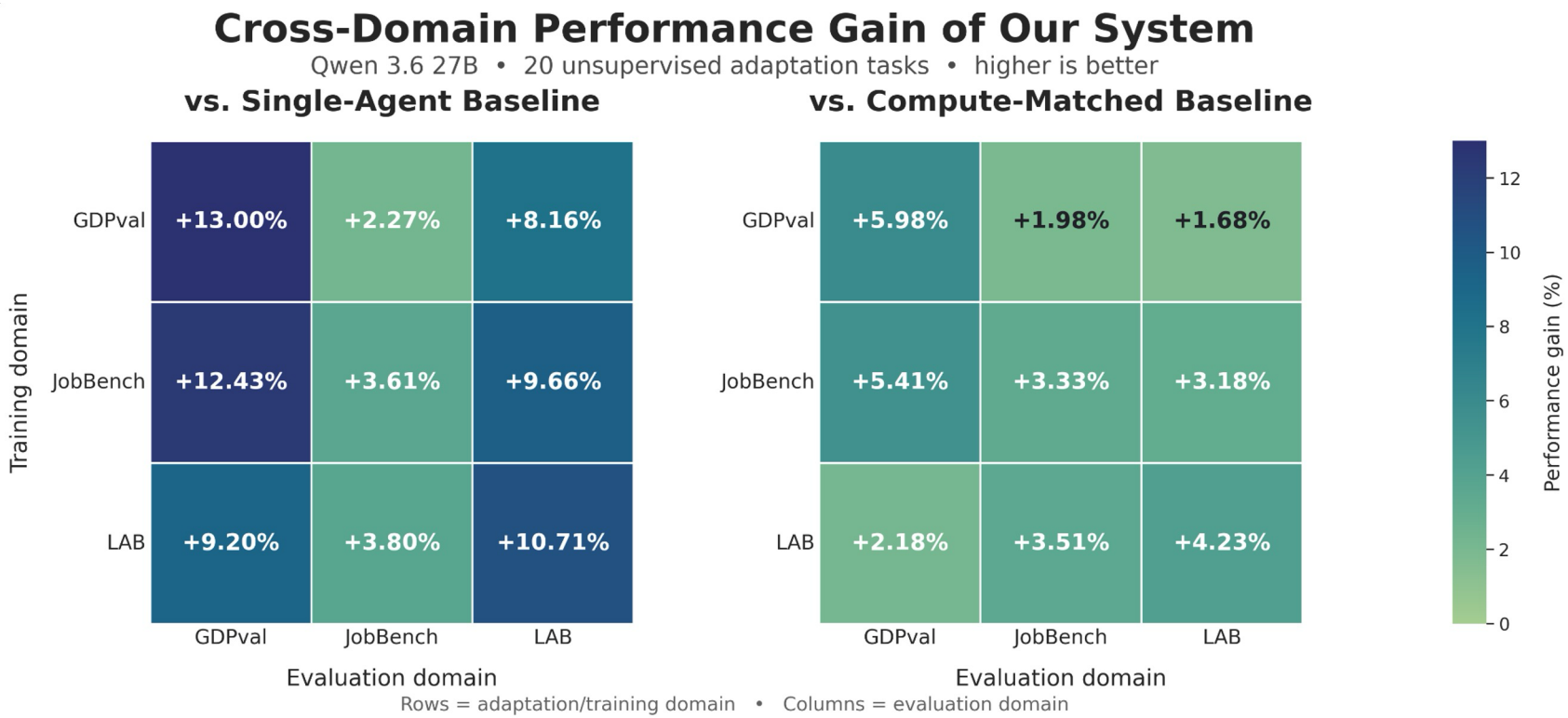
ReOrg: Main Results

- Compared to inducing lower-order procedural knowledge, ReOrg learns from the adaptation tasks more effectively.

	Gemini 3.1 Pro			Qwen 3.6 27B		
Method	GDPval	JobBench	LAB	GDPval	JobBench	LAB
Single worker	51.4	28.7	50.9	57.3	45.5	68.7
ReasoningBank	47.0	28.6	56.7	57.3	43.3	72.7
ReasoningBank + MaTTS	50.3	30.5	58.6	60.0	**46.2**	73.0
AWM	48.1	30.6	57.2	60.5	43.7	72.1
MemP	47.6	**31.2**	57.5	59.5	43.6	76.0
ReOrg, 1 round	**60.0**	30.6	**61.9**	**68.7**	**46.2**	**76.6**

ReOrg: Cross-Domain Transfer

- Skills learned in one professional domain also improve work in the other two, beyond a random organization of reviewers.

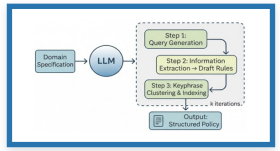


Zooming Out…

- During my PhD study, as LLM capabilities and their deployments improve, I have been rethinking the definition of datastores: **what** knowledge to include and **how** to represent it.

Evolving Datastores with Deployment: 2023

- In Deep Policy Research, we built a proof-of-concept **compute-intensive** datastore which proactively researches knowledge from the web to build a content moderation expert.



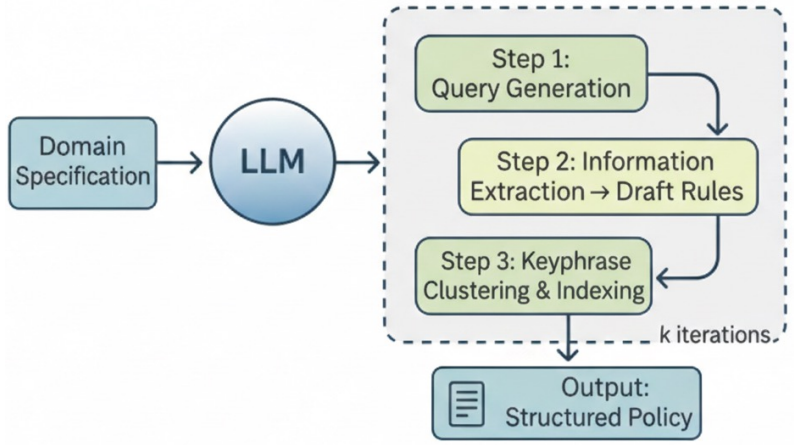
Deep Policy Research (2024)

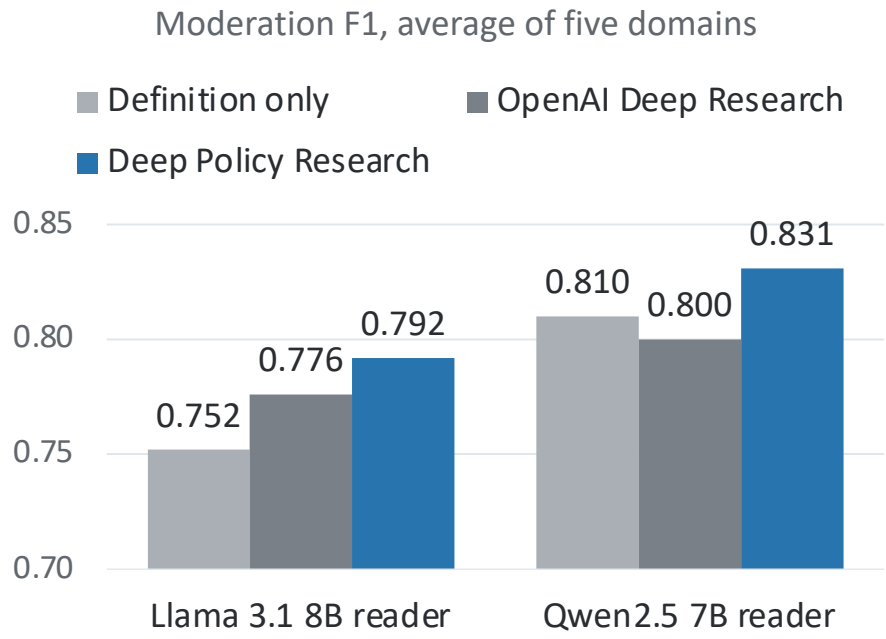
10K doc

↓

A fact

Fact QA

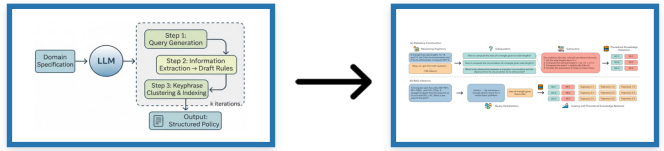




[Open-Domain Safety Policy Construction (EACL Findings 2026)]

Evolving Datastores with Deployment: 2024

- In Reasoning Memory, we scaled up a **general compute-intensive datastore** to benefit a wide range of science, math, and coding reasoning problems.



Deep Policy Research (2024)

10K doc

A fact

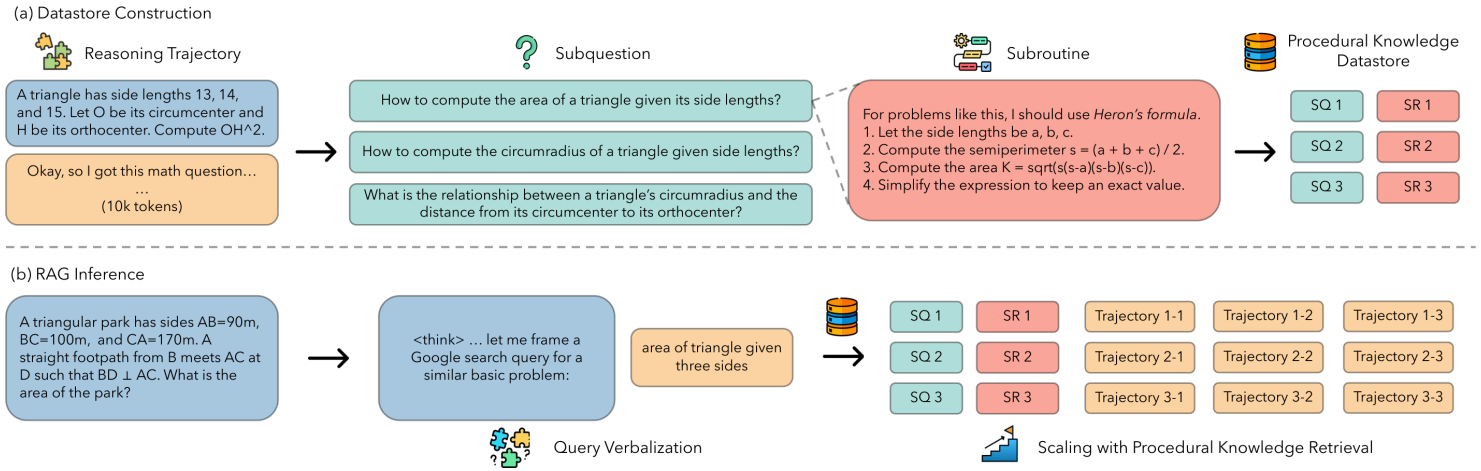
Reasoning Memory (2025)

100k tokens

An insight

Fact QA

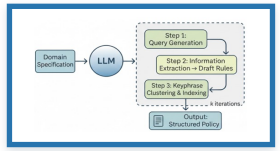
Reasoning



[Procedural Knowledge at Scale Improves Reasoning (COLM 2026)]

Evolving Datastores with Deployment: 2025

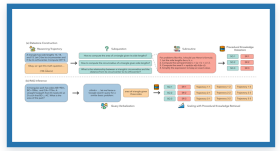
- In LongMemEval V1 & V2, we investigated history compression methods for chatbot and agent personalization after deployment.



Deep Policy Research (2024)

10K doc

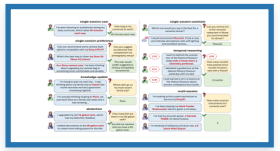
A fact



Reasoning Memory (2025)

100k tokens

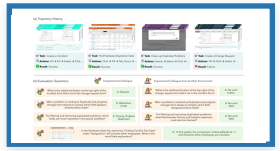
An insight



LongMemEval (2025)

1M chat tokens

A preference



LME-V2 (2026)

100M trajectory tokens

A workflow

Fact QA

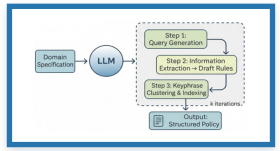
Reasoning

Chatbot

Agent

Evolving Datastores with Deployment: 2026

- In ReOrg, we explore the higher order **organizational knowledge** as the next step, showing the advantage of reusing collaboration structures among agents.

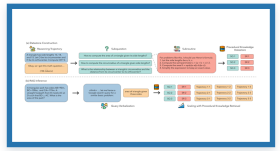


Deep Policy Research (2024)

10K doc

A fact

Fact QA

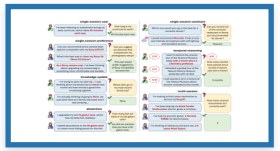


Reasoning Memory (2025)

100k tokens

An insight

Reasoning



LongMemEval (2025)

1M chat tokens

A preference

Chatbot

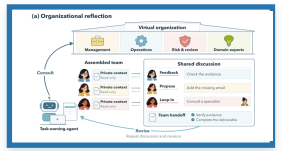


LME-V2 (2026)

100M trajectory tokens

A workflow

Agent



ReOrg (2026)

1B trajectory tokens

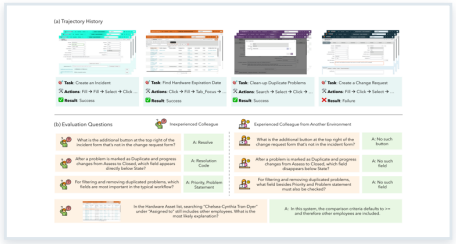
An organization

Long-Horizon Agent

Summary

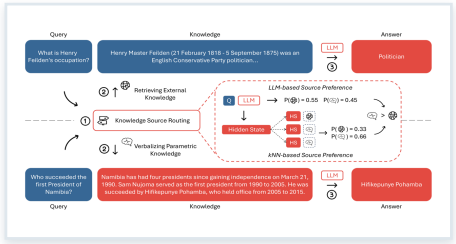
- To bridge the gap between LLMs and an evolving world, my work advances memory mechanisms that are **evaluated faithfully**, **accessed selectively**, and **curated continually**.

Evaluation



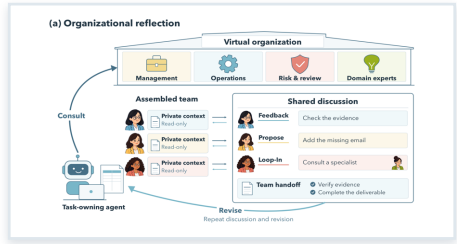
LongMemEval-V2

Model



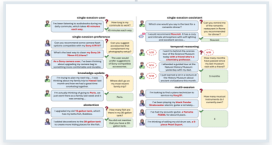
Self-Routing RAG

Datastore



ReOrg

1



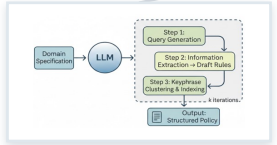
LongMemEval

2



Repoformer

3



Deep Policy Research



Reasoning Memory

Future work

- One important direction is making LLMs leverage memory of different **forms** together.
  - Text memory
    - Model-agnostic
    - Simple to construct
    - Easy to update/debug
    - Time-consuming at inference time
  - Parametric memory
    - Hard to share across models
    - Expensive to train
    - Hard to interpret
    - Cheap at inference time
- Advancing hybrid memory systems allows us to get the best of both worlds.