Can Large Language Models Keep Up? Benchmarking Online Adaptation to Continual Knowledge Streams

Anonymous ACL submission

Abstract

Large language models operating in dynamic real-world contexts often encounter knowledge that evolves continuously or emerges incrementally. To remain accurate and effective, models must adapt to newly arriving information on the fly. We introduce ONLINE ADAPTATION TO CONTINUAL KNOWLEDGE STREAMS (OAKS) to evaluate this capability, establishing a benchmark for *online adaptation over streaming, continually updating knowledge*. Specifically, each model is evaluated at every interval using the same set of questions, allowing us to assess whether it can track and reason over such fine-grained knowledge dynamics across time. To support this setting, we present two datasets: OAKS-BABI and OAKS-Novel, where individual facts evolve multiple times across context chunks. These datasets include dense annotations to measure whether models track changes accurately. Evaluating 14 models with varied inference approaches, we observe significant limitations in current methodologies. Both state-of-the-art models and agentic memory systems fail to adapt robustly on OAKS, demonstrating delays in state-tracking and susceptibility to distraction within streaming environments.[1]

1 Introduction

In real-world settings, knowledge is inherently dynamic, evolving continuously and emerging incrementally. Consequently, LLM-based systems operating as conversational assistants or embodied agents must adapt to information that changes sequentially over time on the fly (Yu et al., 2025; Kim et al., 2024b; Zheng et al., 2025b). For example, assistants receive user context gradually during dialogue (Maharana et al., 2024; Wu et al., 2025), and robots encounter new properties of their environments during exploration (Majumder et al., 2023; Kim et al., 2024a). If the information updates are not integrated in real time, model predictions risk becoming outdated or even unsafe. However, current benchmarks primarily target static knowledge or offline tasks, failing to adequately evaluate online adaptation in dynamic settings.

To fill this void, we introduce ONLINE ADAPTATION TO CONTINUAL KNOWLEDGE STREAMS (OAKS), a benchmark designed to evaluate models in an *online adaptation* setting over *streaming, continually updating knowledge*.[2] OAKS synthesizes continual knowledge learning (Liska et al., 2022; Jang et al., 2022) and online adaptation (Lin et al., 2024; Hu et al., 2023); facts arrive sequentially and may supersede or contradict prior information, necessitating that models dynamically revise their knowledge state. Distinct from prior work, OAKS evaluates whether models can track factual updates, maintain consistency, and generalize when reasoning over long-horizon streams characterized by frequent, fine-grained state changes. To the best of our knowledge, OAKS is the first benchmark to unify these two paradigms, supporting both large-scale fine-grained knowledge adaptation and stepwise online evaluation over streaming knowledge.

To evaluate models on OAKS, we introduce new datasets: OAKS-BABI (OAKS-B), a synthetic dataset derived from the BABILong benchmark (Kuratov et al., 2024), and OAKS-Novel (OAKS-N), a human-curated dataset sourced from literary texts. As shown in Figure 1, the datasets consist of multiple context chunks c_t, where facts evolve dynamically over time intervals t. For each question, we annotate answers at each time interval based on all knowledge accumulated up to that point, capturing *answer transitions* triggered by new information. This setup explicitly evaluates a model's

[1] We will open-source the code and datasets.

[2] *Streaming* refers to a dataset characteristic where facts arrive sequentially over time. *Online adaptation* refers to an inference setting in which a model adapts to the most up-to-date information on the fly.

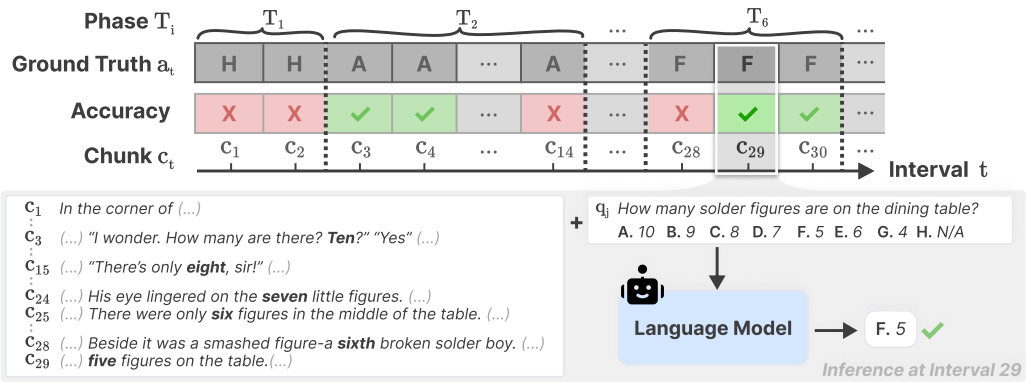

Figure 1: Overview of ONLINE ADAPTATION TO CONTINUAL KNOWLEDGE STREAMS. At each time interval t, a new context chunk c_t is streamed, and the model is queried with context accumulated up to t and a question q_j. Performance is calculated as average accuracy across all intervals by comparing predictions with the ground truth answer a_t, then averaging over all questions. A Phase T_i denotes a contiguous set of chunks sharing the same ground truth answers. Answer options are limited to OAKS-N, as OAKS-B uses an open-ended format.

ability to track when and how answers change in an online setting. During evaluation, the model is asked the same set of questions at each interval, with access to the context up to that point (e.g., chunks c_1 to c_{29} when evaluating at interval 29, as shown in the figure). Performance is measured by interval-level accuracy, reflecting whether the model maintains the correct state at each specific moment in time.

We conduct an extensive analysis of 14 state-of-the-art models on OAKS-B and OAKS-N, including strong closed-sourced model Gemini 3 (Google, 2025) and various sizes of Qwen3 (Yang et al., 2025), utilizing diverse strategies for constructing accumulated context. Our results show that models struggle with OAKS, achieving an average accuracy of 39.4% on OAKS-B and 57.5% on OAKS-N. We find that models particularly degrade with *frequently updating* knowledge, with accuracies dropping to 33.3% on OAKS-B and 53.0% on OAKS-N. While inference-time scaling enhances performance on complex reasoning tasks, its gains are limited when tracking states that undergo frequent updates. Across a range of context construction strategies, including agentic memory systems, naive retrieval augmented generation (RAG), and recency-based approaches, we observe that OAKS remains challenging across all settings. Among these methods, RAG tends to show robust performance, whereas agentic memory approaches outperform simple RAG when frequent knowledge updates exist.

Leveraging the online nature of OAKS, which queries the same question at multiple points within a streaming context, we conduct a fine-grained analysis of models' knowledge tracking behavior. We find that activating "thinking mode" enhances both adaptability and stability, leading to higher overall accuracy. However, distinct failure modes emerge across different models: some tend to over-update, changing predictions unnecessarily, while others under-update, displaying inertia even when the underlying state shifts. Our intra-phase analysis further reveals that even when models achieve comparable accuracy, the underlying causes of error diverge based on their state-transition behaviors: over-updating models tend to capture the correct phase but suffer from frequent distractions within it, whereas under-updating models are prone to missing entire phases. Finally, we observe that performance degrades on questions requiring reasoning over multiple context chunks and at later time intervals as context length increases.

2 Related Work

Continual Knowledge Learning and State Tracking Benchmarks Real-world knowledge is dynamic, motivating benchmarks that evaluate language models under settings where knowledge evolves continually (Kim et al., 2024d; Jang et al., 2022; Liska et al., 2022). However, existing benchmarks typically involve a small number of knowledge updates or focus on divergent fact updates rather than the same underlying fact. Our work addresses this by tracking fine-grained continual updates. OAKS is also related to state-tracking benchmarks, which study how models maintain and update evolving states (Kim and Schuster,

Dataset	Unit	#Steps	U/Q	Fully Annot.
EvolvingQA	Corpus	6+	2.0	✗
StreamingQA	Corpus	4	2.0	✗
StreamingBench	Frame	5	1.0	✗
FactTrack	Fact	39	2.0	✗
MultiWOZ	Turn	14	1.0	✓
BABILong	Full	1	1.0	✗
OAKS-B	Chunk	65	**4.7**	✓
OAKS-N	Chunk	78	**4.7**	✓

Table 1: Comparison between our dataset with prior datasets along four axes: *Unit* = size of each data unit, *#Steps* = number of evaluation steps, *U/Q* = average number of updates (*U*) per question (*Q*), *Fully Annotated (Fully Annot.)* = whether all questions are annotated for each unit.

2023; Niu et al., 2024). While prior work focuses on short-term, structured states such as dialogue slots (Budzianowski et al., 2018; Lee et al., 2019), OAKS instead focuses on open-ended knowledge states over long horizons, evaluating whether models can maintain temporal consistency over a continuous stream of updates in an online setting.[3]

Online Adaptation to Streaming Inputs Prior research on online or lifelong learning in language has focused on self-evolving, lifelong agents that acquire new capabilities or task-level skills over time (Zheng et al., 2025a; Wei et al., 2025), or on tracking synthetic facts with limited update complexity (Wu et al., 2023; Lyu et al., 2025). In contrast, we focus on fine-grained knowledge updates, a setting where minor failures can easily compound into significant errors over time. Inspired by real-time understanding frameworks in the video domain (Lin et al., 2024; Niu et al., 2025), we introduce OAKS to explicitly evaluate realistic streaming knowledge updates at the granularity of individual facts in an online setting.

3 Online Adaptation to Continual Knowledge Streams

In this paper, we present Online Adaptation to Continual Knowledge Streams (OAKS), a benchmark that simulates real-world scenarios in which language models encounter streaming knowledge and must adapt to incrementally revealed updates online. In Section 3.1, we introduce new datasets, and in Section 3.2, we describe the evaluation setup and metric.

[3]Due to length constraint, full list of related works is in Appendix A

3.1 Dataset

To evaluate model performance on OAKS, we present two datasets: OAKS-BABI (OAKS-B), derived from the BABILong benchmark (Kuratov et al., 2024), and OAKS-Novel (OAKS-N), sourced from full-length literary novels. Both datasets feature sequential context chunks containing knowledge updates, with questions explicitly curated to target facts that evolve over time. Formally, each data consists of a set of Q questions $\{q_j\}_{j=1}^{Q}$ and an ordered sequence of C context chunks $\{c_i\}_{i=1}^{C}$. At each time interval t, a new chunk c_t (spanning 2k tokens) is revealed. For each question q_j, we provide a ground-truth answer $a_{j,t}$ and supporting evidence $e_{j,t}$, representing the valid knowledge state conditioned on the cumulative history $\{c_i\}_{i=1}^{t}$.

As detailed in Table 1, unlike prior continual knowledge benchmarks such as EvolvingQA (Kim et al., 2024d) and StreamingQA (Liska et al., 2022), our dataset is composed of granular context increments distributed over a large number of time steps. This structure, coupled with questions that track multiple state transitions, enables stepwise online evaluation. Furthermore, compared to general online adaptation (StreamingBench (Lin et al., 2024)) or fact-tracking benchmarks (FactTrack (Lyu et al., 2025), MultiWOZ (Budzianowski et al., 2018)), we focus specifically on the evolution of semantic knowledge, facilitating a more fine-grained assessment of knowledge adaptation capabilities. Since each question targets a different aspect of knowledge from the context, the timing and frequency of phase transitions vary across questions. To evaluate model robustness under varying degrees of change, we stratify the dataset into three subsets, *Sparse, Moderate, Frequent*, based on the frequency of answer changes for each question.

Additional details on dataset construction, filtering, human annotation, refinement, and statistics on OAKS-B and OAKS-N can be found in Appendix B.1.

OAKS-B OAKS-B repurposes context from BABILong (Kuratov et al., 2024), shifting the focus from static fact retrieval to dynamic knowledge tracking and reasoning. To achieve this, we reformulated the dataset in two ways: (1) generated *new questions* that focus on state changes, ensuring that answers vary depending on the part of the context being considered and require synthesizing or reasoning over multiple facts; (2) annotated *an-*

swers at every time interval explicitly tracking how facts evolve over the stream. The resulting dataset includes four question types: tracking, counting, bridge, and comparison. Tracking questions focus on understanding frequent fact updates; counting, bridge, and comparison questions require aggregation and reasoning over multiple chunks. The dataset contains 1.2k questions in total, a context length of 128k tokens (65 chunks), and an average of 4.7 answer changes per question.

OAKS-N OAKS-N leverage full-length novels to provide contexts with natural narratives, rich storylines, dynamically interacting characters, and complex, concurrent plotlines. For each of the 39 novels included, we generated an initial set of question candidates using Gemini 2.5 Pro (Gemini Team et al., 2025), which were subsequently rigorously curated by human experts to ensure quality and consistency. We recruited experienced freelancers familiar with the books, investing a total of $17.4k to validate answers and supporting evidence at each interval. These annotators also refined or replaced low-quality questions, generated missing answer options, and removed ambiguous answer options, resulting in a final dataset filtered down to 55% of the initial question pool.

The final dataset contains 870 multiple-choice questions (avg. 5.5 options) in total, with an average book length of 150.6k tokens (77.6 chunks, ranging from 26 to 286), and an average of 4.7 answer changes per question.

3.2 Evaluation Setup and Metric

Setup As illustrated in Figure 1, we evaluate models over a sequence of time-interval chunks. At each interval, the model is asked the same set of questions based on all the context observed so far, testing its ability to update, incorporate, or retain knowledge on the fly as new information arrives. Formally, at interval t, a model M observes the set of chunks up to t, $\mathcal{S}_t = \{c_i\}_{i=1}^{t}$, and for each question $q_j \in Q$ predicts an answer $p_{j,t} = M(q_j, \mathcal{S}_t)$.[4]

Metric We evaluate model performance using interval-level accuracy. At each interval, we compare the model's prediction with the current ground-truth answer, assigning a score of 1 for a match and 0 otherwise. These scores are averaged across intervals to obtain a question-level accuracy, which is subsequently averaged over the entire dataset to compute the final benchmark score. Evaluation criteria and formal definition of metric in Appendix B.2.

4 Experimental Setup

In this section, we share details of the models, context representation, and inference setup. Additional details in Appendix C.

Base Models We evaluate 14 LLMs across open-source and proprietary families and a wide range of scales. We primarily use the Qwen family (Yang et al., 2025, 2024) as the baseline to utilize its multiple sizes (Qwen3-(4B, 8B, 235B), Qwen3-Next-80B, Qwen3-30B (+Thinking), and Qwen2.5-7B), and include comparable open-source models such as GPT-OSS (Agarwal et al., 2025) (20B, 120B), and Gemma 3 (Team et al., 2025) (4B, 27B). For proprietary models, we focus on Gemini 2.5 (Flash, Pro) (Gemini Team et al., 2025), and Gemini 3 (Google, 2025). Unless noted, all models but Gemini are non-thinking versions.

Context Representations (Base, RAG, Agentic Memory Systems) For a *Base* setting, we concatenate all preceding chunks up to the current interval t, truncating older chunks when the model's context limit is exceeded.[5] For *RAG*, we use Qwen3-Embedding-0.6B (Zhang et al., 2025) as a retrieval model. Retrieval is restricted to chunks from previous time intervals. Unless otherwise specified, we retrieve the top 30 most relevant memory chunks. For *agentic memory systems*, we evaluate HippoRAG-V2 (Gutiérrez et al., 2025), MemAgent (Yu et al., 2025), and A-Mem (Xu et al., 2025b), which maintain and update memory incrementally up to the current time interval.

Inference Setup Experiments were mostly conducted with 8 A100 80G GPUs using vLLM (Kwon et al., 2023). We use the same setup of temperature 0.7, TopP 0.8, TopK 20, and MinP 0, if not stated otherwise, as the best practice.

5 Evaluation Result

5.1 Overall results

In this section, we analyze key findings from Table 2, summarizing the overall performance of the evaluated models on OAKS-B and OAKS-N.

[4] Our dataset includes sentence-level evidence annotation for each answer, but this would be computationally heavy setup; we therefore use 2k token chunks as the evaluation unit.

[5] For Qwen3-235B, we cap the context to 131k tokens due to GPU memory constraints despite its nominal 1M token capacity.

			OAKS-BABI								OAKS-Novel							
			Base				RAG				Base				RAG			
Model	Size	Active	All	Sprs.	Mod.	Freq.	All	Sprs.	Mod.	Freq.	All	Sprs.	Mod.	Freq.	All	Sprs.	Mod.	Freq.
Qwen3	4B	-	26.4	34.4	23.1	18.5	29.3	37.2	25.9	21.6	51.3	59.1	49.7	47.9	55.8	68.1	56.0	48.3
	8B	-	33.1	36.8	34.0	25.8	35.1	39.9	34.5	28.4	52.7	62.5	51.7	47.6	47.7	60.0	47.8	40.4
	30B	3B	35.8	38.5	37.3	29.4	37.8	40.1	39.2	32.0	62.8	71.2	63.7	57.1	61.0	68.9	63.0	54.9
	80B	3B	41.1	44.7	42.0	34.1	43.5	47.4	44.2	36.2	64.6	74.5	63.3	59.9	65.3	74.4	65.3	59.9
	235B	22B	46.8	48.4	49.8	40.0	44.7	46.4	46.8	39.0	64.7	73.4	64.9	59.4	66.0	74.7	66.5	60.6
Qwen2.5	7B	-	24.7	27.3	23.6	22.4	32.4	35.5	32.5	27.5	33.9	41.2	32.9	30.6	40.2	49.6	41.0	34.2
GPT-OSS	20B	3.6B	22.4	21.9	23.8	21.3	22.2	22.2	22.9	21.2	45.3	54.3	45.4	39.9	44.2	54.7	44.3	38.1
	120B	5.1B	37.5	40.5	38.4	31.6	33.7	35.9	33.9	29.8	54.6	62.8	53.6	50.6	54.6	65.4	54.3	48.6
Gemma 3	4B	-	24.2	22.9	26.3	23.3	25.2	24.9	26.5	23.7	38.0	41.8	40.1	34.1	35.2	39.4	38.1	30.5
	27B	-	37.8	40.2	40.0	30.9	38.6	42.8	39.3	30.9	61.2	70.1	60.5	56.6	56.0	67.3	57.5	48.3
Gemini 2.5	Flash	-	56.2	58.3	58.6	49.3	56.6	60.3	58.4	47.9	65.6	75.2	65.6	60.0	64.5	74.2	64.3	59.0
	Pro	-	60.3	64.6	62.1	50.9	**62.2**	**67.2**	**63.3**	**52.9**	**76.7**	**83.7**	**76.4**	72.9	**75.8**	**82.2**	**75.4**	**72.4**
Gemini 3	Pro	-	**66.3**	**70.3**	**69.3**	**55.6**	51.3	51.0	55.5	45.8	75.5	80.3	75.3	**73.0**	74.0	79.9	74.0	70.6

Table 2: Accuracy (%) of models on OAKS-B and OAKS-N under Base and RAG settings. Gemini results are reported with the thinking mode enabled. *Active* denotes the number of active parameters in Mixture-of-Experts (MoE) models. *Sprs.*, *Mod.*, and *Freq.* denote Sparse, Moderate, and Frequent transition subsets; *All* is averaged over all subsets. The highest score within each column is in **bold**.

OAKS is difficult across various models The results indicate that OAKS task remains challenging, with substantial room for improvement across all evaluated systems. The average accuracy is 33.0% on OAKS-B and 52.9% on OAKS-N for open-sourced models; 60.9% on OAKS-B and 72.6% on OAKS-N for closed-sourced models. Even for the strongest closed-source model, Gemini 3 Pro, performance reaches only 66.3% on OAKS-B and 75.5% on OAKS-N.

Performance tends to improve with a larger, better base model Overall, performance generally scales with model size within the same model family. Among models of comparable size, the Qwen3 family achieves higher accuracy than alternatives. Further, comparing Qwen2.5 and Qwen3 shows that a stronger base model consistently leads to better performance. Proprietary models like Gemini also tend to consistently exhibit higher performance than open-source models. MoE models perform comparably to fully dense models of similar scale, even with a small number of active parameters (e.g., Qwen3 30B vs. Gemma 3 27B).

Questions with many transitions are difficult When comparing *Sparse*, *Moderate*, and *Frequent* subsets, which are defined by the number of answer changes for each question, we observe that the *Frequent* set, which contains questions with many transitions, poses greater challenges for timely updates. On OAKS-B, performance decreases from 42.2% for *Sparse* questions to 40.6% for *Moderate* and 33.3% for *Frequent*; on OAKS-N, the corresponding values are 65.4%, 57.2%, and 53.0% on average. We hypothesize that this degradation arises because frequent answer changes lead to more dynamic knowledge states, requiring models to repeatedly update multiple facts while retaining previously valid information, which exacerbates both tracking and retention difficulties.

Naive RAG shows limited effectiveness in OAKS When comparing RAG with the Base setting, we observe different patterns across the models and datasets, but modest drop in performance on average: an improvement of 0.4% on *Sparse* but a performance drop of 0.04% on *Moderate* and 0.8% on *Frequent* subsets. This suggests that a simple RAG approach is insufficient for OAKS, particularly in the *Frequent* subset, where knowledge evolves dynamically, and multiple interrelated facts are distributed across overlapping context chunks. We identify two main challenges: (1) retrieval itself becomes difficult when many semantically related chunks exist and questions often require reasoning over multiple chunks (Press et al., 2023; Shao et al., 2025); and (2) even with successful retrieval, prior work shows that models are sensitive to input context and often struggle to effectively process complex or irrelevant contexts, which can rather degrade performance (Lee et al., 2025b; Mallen et al.,

Model	Think	All	Tracking	Counting	Bridge	Comparison
Qwen3-30B	-	35.8	27.1	34.8	24.6	48.3
	✓	**43.6**	**43.8**	**37.7**	**37.3**	**53.9**
Gemini 2.5 Flash	-	43.2	54.4	42.6	30.0	53.1
	✓	**56.2**	**58.7**	**53.9**	**42.4**	**69.6**
Gemini 2.5 Pro	-	42.9	52.4	45.8	29.8	50.2
	✓	**60.3**	**55.3**	**57.5**	**51.0**	**71.7**

Table 3: Accuracy (%) of Qwen3-30B and Gemini 2.5 on OAKS-B by question type with and without thinking mode (Think). Best in **bold**.

Strategy	All	Sprs.	Mod.	Freq.
Base	24.7	27.3	23.6	22.4
RAG (30)	**32.4**	**35.5**	32.5	27.5
RW (30)	27.6	29.7	26.9	25.5
RAG (15) + R.W (15)	31.5	34.0	31.8	27.2
HippoRAG2	20.8	19.5	23.9	18.5
MemAgent	31.3	30.7	**33.6**	**29.1**
A-Mem	30.3	30.6	33.3	25.6

Table 4: Accuracy (%) of agentic memory system using Qwen2.5-7B instruct as base model on OAKS-B. The number in parentheses represents the number chunks prepended by each method. Best in **bold**.

2023). Thus, we further analyzed more advanced context representation strategies in Section 5.3. Retrieval performance in Appendix D.2.

5.2 Thinking improves performance, especially on complex reasoning questions

Inference-time scaling with additional intermediate *thinking* processes (Wei et al., 2022; OpenAI et al., 2024; Kojima et al., 2022; DeepSeek-AI et al., 2025) consistently improves performance on OAKS. Table 3 compares models with and without *thinking* mode for Qwen3-30B and Gemini 2.5. In OAKS-B, most question types other than tracking type require reasoning over multiple pieces of evidence distributed across non-contiguous chunks. Enabling thinking mode leads to consistent improvements in overall accuracy, where the most pronounced improvements are observed in bridge-type questions (15.4%). Bridge questions are inherently more challenging as they necessitate multi-hop reasoning, integrating multiple factual sentences while simultaneously tracking various states within the context. In contrast, counting-type questions require tracking only a single state with high precision; thus, the performance gain is relatively marginal (8.0%). These results suggest that the internal reasoning process of the thinking mode offers the most substantial benefit when task complexity is high, particularly requiring the simultaneous tracking of multiple independent states.

5.3 OAKS is difficult for even agentic memory systems

Table 4 shows the performance of agentic memory systems built on Qwen2.5-7B-Instruct[6] in OAKS-B. Overall, agentic memory methods underperform naive RAG in aggregated accuracy. However, on the *Moderate* and *Frequent* subsets, they show competitive or improved performance, with MemAgent achieving the best results. This is likely due to MemAgent's interval-based memory tracking training objective, which aligns with our evaluation setting that requires continual tracking of dynamic knowledge. Nonetheless, its performance remains limited because training is based on static question types with rewards computed only after processing all chunks, rather than at each interval. These results highlight OAKS as a challenging benchmark even for such agentic memory systems due to its frequent, fine-grained knowledge updates. More analysis in Appendix D.2.

6 Analysis

In this section, we present a detailed analysis of models' behavior on OAKS. We focus primarily on OAKS-B to control for models' prior knowledge (Appendix D.3) with Qwen3-30B unless otherwise specified. Appendix E provides additional analyses on OAKS-N, models' evidence reasoning, and the correlation between OAKS and long-context understanding ability.

6.1 Fine-grained analysis of predicted knowledge transition behavior

To analyze how models track evolving factual knowledge, we categorize their behavior under two ground truth (GT) scenarios: *GT Phase Transitions* and *No GT Transition*, where the phase changes or stays, respectively.[7] Table 5 shows the average frequency of different behavioral patterns across all intervals. Model's predicted transition behavior is defined by two factors: (1) whether the model's predicted answer **C**hanges or **S**tays relative to the previous prediction, and (2) whether the resulting

[6] We chose Qwen2.5-7B-Instruct as MemAgent was trained on top of it.

[7] Since Stay intervals are prevalent (94% of all intervals), rates are averaged within each GT scenarios to sum to 100%.

Models	Size	Think	GT Phase Transitions (Change)				No GT Transition (Stay)			
			Adaptability (C / ✓)	Maladaptation (C / ✗)	Prescience (S / ✓)	Stubbornness (S / ✗)	Lag (C / ✓)	Volatility (C / ✗)	Stability (S / ✓)	Obstinacy (S / ✗)
Gemma 3	27B	-	31.6	28.6	11.9	27.9	12.2	27.0	25.2	35.6
GPT-OSS	120B	-	39.1	40.8	5.5	14.6	12.8	46.5	24.2	16.5
Qwen3	30B	-	34.3	33.6	9.7	22.4	8.9	36.7	26.3	28.1
		✓	39.6	34.6	7.7	18.2	13.0	37.7	30.4	18.9
Gemini 2.5	Flash	-	36.3	17.0	17.2	29.5	7.5	16.9	35.0	40.7
		✓	47.5	23.8	15.0	13.7	12.4	27.5	43.3	16.8
Avg			38.1	29.7	11.2	21.1	11.1	32.1	30.7	26.1

Table 5: Analysis of knowledge tracking behavior on OAKS-B. The table shows the average occurrence rate of specific tracking behaviors across all time intervals. The second row shows the behavioral types (e.g., `Maladaptation`) along with the model's predicted action (whether predicted answer **C**hange or **S**tay from previous prediction) and the resulting prediction correctness (correct: ✓ vs. incorrect: ✗).

answer is correct (✓) or incorrect (✗). For descriptive purposes, we map these combinations to specific behavioral archetypes (e.g., `Adaptability`). We provide a schematic illustration of these behaviors, full results, and additional analysis in Appendix E.1.

Enabling explicit thinking alters transition behavior Across both Qwen3 and Gemini 2.5, enabling "thinking mode" consistently shifts model behavior. Non-thinking variants exhibit a higher rate of `Prescience`, `Stubbornness`, and `Obstinacy`, indicating that it tends to retain previous answers even when an update is required. In contrast, thinking-enabled models show a higher rate of `Adaptability` and `Stability`, suggesting explicit reasoning improves both transition timing and answer correctness.

Different models exhibit distinct transition behavior Comparison over different non-thinking models shows a distinct behavior in how often they change predictions. GPT-OSS and Qwen3 exhibit a higher overall change rate (**C**, avg 63.2%) than stay rate (**S**, avg. 36.8%). In contrast, Gemini 2.5 and Gemma 3 exhibit a higher stay rate (**S**, avg. 55.8%) compared to the change rate (**C**, avg 44.2%). This is also reflected in their dominant error modes: `Volatility` and `Obstinacy`, respectively. These results suggest model-specific biases toward either *over-updating* or *under-updating* knowledge.

Models often detect true transitions but frequently make unnecessary updates We analyzed which behavior is dominant under each GT scenarios. Under *GT Phase Transitions*, `Adaptability` is the most frequent behavior (38.1%), indicating that models often detect true transitions. In contrast, *No GT Transition* is dominated by `Volatility` (32.1%), showing that models frequently update even when the underlying fact remains unchanged. This suggests that models tend to be sensitive to true knowledge updates but also prone to unnecessary changes, likely due to interference from surrounding contextual information.

Model	Size	Think	ACC	AL(↓)	DS(↓)	PM(↓)
Gemma 3	27B	-	37.8	5.4	**26.5**	30.3
GPT-OSS	120B	-	37.5	8.6	38.8	15.1
Qwen3	30B	-	35.8	6.9	34.0	23.2
		✓	43.6	9.0	37.0	10.4
Gemini 2.5	Flash	-	43.2	6.2	28.3	22.3
		✓	**56.2**	**5.1**	31.8	**7.0**

Table 6: Analysis of intra-phase behavior (ACC, AL, DS, PM) on OAKS-B over models. Best in **bold**.

6.2 Intra-phase analysis shows trade-off between phase capture and stability

To evaluate the models' *intra-phase* behavior at finer granularity, we categorize error intervals into three cases and score separately: Acquisition Latency (AL), Distraction Susceptibility (DS), and Phase Miss rate (PM). AL quantifies the delay between a ground-truth state transition and the first correct prediction; DS measures the frequency of incorrect predictions after an initial correct prediction within a phase; PM denotes phases in which the model gets all wrong. Same as Accuracy, all metrics are normalized by the total number of intervals per question and averaged across questions.

Table 6 presents a comparative analysis of met-

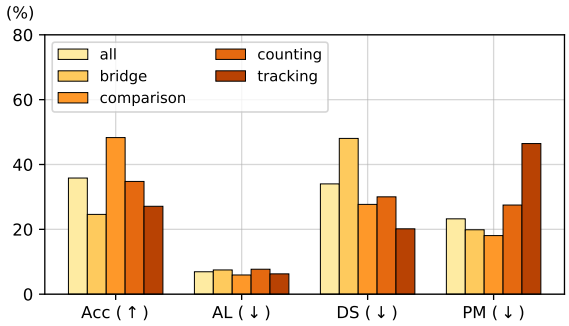

Figure 2: Accuracy (%) across different question types of OAKS-B.

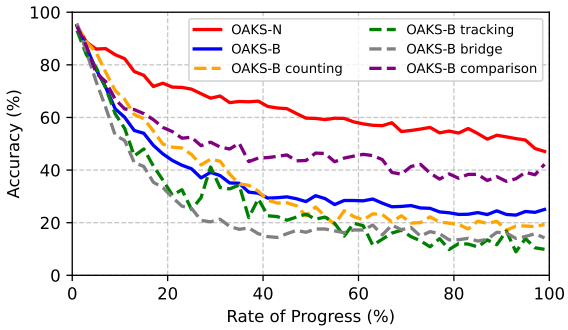

Figure 3: Accuracy (%) across the timestep where the question is asked.

rics across representative models with similar accuracy. Although Gemma 3 and GPT-OSS achieve comparable performance, Gemma 3 exhibits a substantially higher PM rate, indicating failure to capture some phase entirely, whereas GPT-OSS, with high `volatility`, successfully captures each phase at least once, resulting in lower PM but higher AL and DS. Comparing the thinking vs. non-thinking model of Qwen3-30B and Gemini 2.5 Flash, we find that accuracy gains are primarily driven by the models' improved ability to capture ground-truth states at least once per phase, as reflected by substantially reduced PM rate. However, higher DS of the thinking model implies that contextual distraction, where the model loses track of a previously identified state as the input length increases, remains a persistent challenge.

6.3 Failure modes vary across question types in OAKS-B

Figure 2 analyzes model performance across OAKS-B question types with varying reasoning demands and number of chunks to attend to. Bridge questions require simultaneous tracking and updating of multiple states, resulting in the highest DS due to increased interference from monitoring multiple entries. Comparison questions remain relatively high performance, likely not because they are easier, but because the candidate answers are embedded inside the question, similar to multiple choice questions, reducing the search space. Tracking-type questions, even though involving only a single piece of evidence, remain challenging with the highest PM rate, due to its frequent state transitions (8.8 vs. 3.7-5.7 in other types). Collectively, these results indicate that different question types induce distinct failure modes in streaming contexts, many of which stem from difficulties in fine-grained fact tracking.

6.4 Accuracy degrades at later intervals

Figure 3 shows that average accuracy at each interval degrades at later intervals. This degradation is more pronounced in OAKS-B, where the supporting evidence typically appears only once; if the model fails to capture it when it first appears, the error persists and accumulates in subsequent intervals. Within OAKS-B, bridge and tracking questions exhibit the largest degradation, as they are more vulnerable to missed evidence and compounded errors over time. In contrast, OAKS-N show more stable performance, likely because relevant information is often revisited across intervals, partially mitigating error accumulation.

7 Conclusion

In this paper, we introduce ONLINE ADAPTATION TO CONTINUAL KNOWLEDGE STREAMS (OAKS), a benchmark for evaluating language models in the setting of online adaptation to streaming, continually evolving knowledge. We present two datasets, OAKS-BABI and OAKS-Novel, which contain context incrementally revealed in small chunks over long sequences and fixed questions each with answers annotated for every interval to track evolving facts. Experiments on 14 models with varying inference strategies show that the task remains challenging even for state-of-the-art models and agentic memory systems, especially under frequent knowledge updates and in later intervals. We further find that models are easily distracted by surrounding contextual information and often lose track of previously identified states during the adaptation.

Limitations

Evaluation on OAKS entails extensive inference due to the incrementally accumulating nature of the context. While we have included 14 representative models in this study, due to limited computational costs and API expenses, we could not further test on broader range of architectures. Future work could extend this benchmark to a more diverse set of models to further generalize our findings.

Exploring even more complex natural texts with more frequent or complex fact transitions would further enable analysis of scalability and error accumulation. Also, datasets with contexts that are free from the model's parametric knowledge priors would provide a clearer lens on online adaptation behavior.

Our current analysis focuses on inference-time adaptation via incremental context accumulation. However, OAKS also serves as a valuable testbed for parametric online learning, and future work could explore how models update their internal weights to assimilate evolving knowledge.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *ArXiv preprint*, abs/2508.10925.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *ArXiv preprint*, abs/2412.15204.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Booookscore: A systematic exploration of book-length summarization in the era of llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Gemini Team, Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *ArXiv preprint*, abs/2507.06261.

Google. 2025. Gemini 3.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *ArXiv preprint*, abs/2502.14802.

Jiale Han, Austin Cheung, Yubai Wei, Zheng Yu, Xusheng Wang, Bing Zhu, and Yi Yang. 2025. Rag meets temporal graphs: Time-sensitive modeling and retrieval for evolving knowledge. *ArXiv preprint*, abs/2510.13590.

Nathan Hu, Eric Mitchell, Christopher Manning, and Chelsea Finn. 2023. Meta-learning online adaptation of language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4418–4432, Singapore. Association for Computational Linguistics.

Yuanzhe Hu, Yu Wang, and Julian McAuley. 2025. Evaluating memory in llm agents via incremental multi-turn interactions. *ArXiv preprint*, abs/2507.05257.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2022. Towards continual knowledge learning of language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Gregory Kamradt. 2023. Needle in a haystack - pressure testing llms.

Byeonghwi Kim, Minhyuk Seo, and Jonghyun Choi. 2024a. Online continual learning for interactive instruction following agents. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Chaeeun Kim, Soyoung Yoon, Hyunji Lee, Joel Jang, Sohee Yang, and Minjoon Seo. 2024b. Exploring the practicality of generative retrieval on dynamic

corpora. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13616–13633.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Yekyung Kim, Yapei Chang, Marzena Karpinska, Aparna Garimella, Varun Manjunatha, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024c. Fables: Evaluating faithfulness and content selection in book-length summarization. *ArXiv preprint*, abs/2404.01261.

Yujin Kim, Jaehong Yoon, Seonghyeon Ye, Sangmin Bae, Namgyu Ho, Sung Ju Hwang, and Se-Young Yun. 2024d. Carpe diem: On the evaluation of world knowledge in lifelong language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5401–5415, Mexico City, Mexico. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Y. Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Dong-Ho Lee, Adyasha Maharana, Jay Pujara, Xiang Ren, and Francesco Barbieri. 2025a. Realtalk: A 21-day real-world dataset for long-term conversation. *ArXiv preprint*, abs/2502.13270.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Hyunji Lee, Franck Dernoncourt, Trung Bui, and Seunghyun Yoon. 2025b. Corg: Generating answers from complex, interrelated contexts. *ArXiv preprint*, abs/2505.00023.

Hyunji Lee, Se June Joo, Chaeeun Kim, Joel Jang, Doyoung Kim, Kyoung-Woon On, and Minjoon Seo. 2024. How well do large language models truly ground? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2437–2465, Mexico City, Mexico. Association for Computational Linguistics.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. 2024. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *ArXiv preprint*, abs/2411.03628.

Adam Liska, Tomás Kociský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 13604–13622. PMLR.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Zhiheng Lyu, Kevin Yang, Lingpeng Kong, and Dan Klein. 2025. Facttrack: Time-aware world state tracking in story outlines. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2825–2848.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. 2023. Clin: A continually learning language agent for rapid task adaptation and generalization. *ArXiv preprint*, abs/2310.10134.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023.

When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Cheng Niu, Xingguang Wang, Xuxin Cheng, Juntong Song, and Tong Zhang. 2024. Enhancing dialogue state tracking models through llm-backed user-agents simulation. *ArXiv preprint*, abs/2405.13037.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, and 1 others. 2025. Ovobench: How far is your video-llms from real-world online video understanding? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18902–18913.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, and 1 others. 2025. Reasonir: Training retrievers for reasoning tasks. *ArXiv preprint*, abs/2504.20595.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *ArXiv preprint*, abs/2503.19786.

Luanbo Wan and Weizhi Ma. 2025. Storybench: A dynamic benchmark for evaluating long-term memory with multi turns. *ArXiv preprint*, abs/2506.13356.

Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Qian Wang, and Yue Zhang. 2024. Novelqa: A benchmark for long-range novel question answering. *Preprint*, arXiv:2403.12766.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. 2025. Mem-{\alpha}: Learning memory construction via reinforcement learning. *ArXiv preprint*, abs/2509.25911.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed H Chi, and 1 others. 2025. Evo-memory: Benchmarking llm agent test-time learning with self-evolving memory. *ArXiv preprint*, abs/2511.20857.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomás Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*.

Yuhao Wu, Tongjun Shi, Karthick Sharma, Chun Wei Seah, and Shuhao Zhang. 2023. Online continual knowledge learning for language models. *ArXiv preprint*, abs/2311.09632.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. 2025a. Streamingvlm: Real-time understanding for infinite video streams. *ArXiv preprint*, abs/2510.09608.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025b. A-mem: Agentic memory for llm agents. *ArXiv preprint*, abs/2502.12110.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *ArXiv preprint*, abs/2505.09388.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. Qwen2.5 technical report. *ArXiv preprint*, abs/2412.15115.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and 1 others. 2025. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *ArXiv preprint*, abs/2507.02259.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *ArXiv preprint*, abs/2506.05176.

Junhao Zheng, Xidi Cai, Qiuke Li, Duzhen Zhang, ZhongZhi Li, Yingying Zhang, Le Song, and Qianli Ma. 2025a. Lifelongagentbench: Evaluating llm agents as lifelong learners. *ArXiv preprint*, abs/2505.11942.

Junhao Zheng, Chengming Shi, Xidi Cai, Qiuke Li, Duzhen Zhang, Chenxing Li, Dong Yu, and Qianli Ma. 2025b. Lifelong learning of large language model based agents: A roadmap. *ArXiv preprint*, abs/2501.07278.

A Extended Related Work

Continual Knowledge Learning and State Tracking Benchmarks Real-world knowledge is inherently dynamic: facts can become outdated, remain invariant, or require the incorporation of entirely new information. Thus, there is growing interest in evaluating language models under settings where knowledge evolves continually (Kim et al., 2024d; Jang et al., 2022). However, existing benchmarks typically involve a limited number of knowledge updates and often expand the knowledge with divergent facts rather than repeatedly updating the same underlying fact. This makes it difficult to isolate and assess a model's ability to track continual updates to identical pieces of knowledge over time. Our work addresses this gap by providing a fine-grained evaluation of dynamic knowledge transitions at the individual fact for the same question set.

We also draw on research in state tracking, which studies how models maintain evolving states (Kim and Schuster, 2023; Niu et al., 2024). It focuses on temporal consistency and stepwise updates, which are similar to OAKS. However, previous studies typically address short-term, structured states, such as dialogue slots (Budzianowski et al., 2018; Lee et al., 2019), whereas OAKS focuses on open-ended, continually updating knowledge states over long horizons in an online setting, evaluating a model's ability to maintain temporal consistency across a long-term continuous stream of information without updating parameters and keeping in context.

Online Adaptation of Streaming Inputs In real-world scenarios, knowledge often arrives sequentially over time, requiring models to continuously update their internal knowledge based on streaming inputs, rather than being presented in a static, offline format where all relevant information is available upfront. Prior research on online learning in the text domain has focused on self-evolving, lifelong agents that acquire new capabilities or task-level skills over time (Zheng et al., 2025a; Wei et al., 2025) or synthetic facts with small updates (Wu et al., 2023; Lyu et al., 2025). These works emphasize agent competence or behavior, but largely operate at the granularity of *tasks* rather than individual pieces of knowledge. However, evaluating fine-grained knowledge updates (i.e., when and how specific facts change) is important, as failures to incorporate small but relevant updates

can propagate and accumulate into larger errors when such knowledge is reused or composed in downstream tasks. Inspired by streaming or online video benchmarks (Lin et al., 2024; Xu et al., 2025a), we introduce, to the best of our knowledge, the first benchmark in the text domain that explicitly evaluates streaming knowledge updates at the granularity of individual facts.

Long-Context Understanding Benchmarks With recent LLMs able to process increasingly long contexts, a variety of benchmarks have been proposed to measure long-context understanding. Synthetic benchmarks, e.g., Needle-in-a-Haystack (Kamradt, 2023; Liu et al., 2024), evaluate a model's ability to retrieve specific information embedded within lengthy inputs, where the target information is sparsely embedded and not strongly correlated with the surrounding context. Other works adopt more naturalistic or conversational settings. For example, long-context dialogue datasets (Maharana et al., 2024; Wu et al., 2025; Lee et al., 2025a; Wan and Ma, 2025) require models to answer questions based on extended conversations or narrative formats. There are also benchmarks using long-form documents or novels, where models must answer questions using the full text. Moreover, agent-based benchmarks (Hu et al., 2025; Wang et al., 2025) evaluate long-context reasoning through interaction with an external environment. Our task is similar in that it also involves long contexts, but differs by focusing on continually updating knowledge in an online setting. To capture this, we segment the long context into temporal chunks and annotate answers for each question at every chunk, enabling evaluation of a model's ability to track and update knowledge over time.

B Online Adaptation to Continual Knowledge Streams

B.1 Dataset

B.1.1 Data statistics

Partition based on frequency of answer changes per question We partition each dataset into three subsets based on the frequency of answer changes per question. We define the bins such that the number of samples in each group is approximately balanced, while ensuring that all questions with the same number of changes are assigned to the same group to maintain consistency in our analysis on how the number of answer changes affect on performance. The resulting distributions are in Table 7. Both of our datasets are in English. Figure 4 shows the frequency distribution of answer changes per question.

Dataset	Total	Sprs.	Mod.	Freq.
OAKS-B	1,224	40%	25%	35%
OAKS-N	870	25%	33%	42%

Table 7: Distribution of answer change frequency across datasets: Sparse (Sprs.), Moderate (Mod.), and Frequent (Freq.)

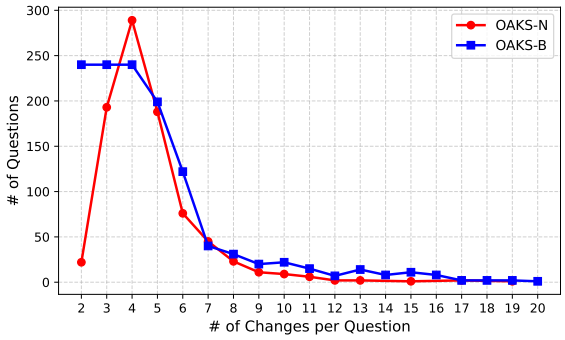

Figure 4: Frequency distribution of answer changes per question for OAKS-B and OAKS-N.

For OAKS-B, the subsets correspond to 2–3, 4–5, and 6–20 changes, respectively; for OAKS-N, the ranges are 2–3, 4, and 5–19 changes.

Since our dataset includes sentence-level evidence annotation for each answer, we could perform sentence-wise fine-grained evaluation using sentence as the data unit. However, this would be computationally expensive, as it would require running inference for every sentence. Therefore, we adopt 2k token chunks as the dataset unit for evaluation.

B.1.2 OAKS-BABI

BABILong OAKS-B uses a sample of context from BABILong (Kuratov et al., 2024), which builds on the bAbI benchmark (Weston et al., 2016) fact-based world state (e.g., "Mary moves from kitchen to hallway"), which is interleaved within long-form novels to evaluate retrieval and reasoning under high distractor density. BABILong is under Apache License, Version 2.0, and the license of the dataset used when constructing BABILong is PG-19 corpora (Rae et al., 2020) under Apache 2.0 License7 and bAbI dataset (Weston et al., 2016) under BSD License8.

Algorithm 1 OAKS-BABI dataset construction

Require: Narrative facts $\mathcal{F}$, Question Templates $\mathcal{P}$, Canonical Verb Groups $\mathcal{V}$
Ensure: Dataset $\mathcal{D} = \{(Question, Timeline_Answers)\}$

```
S ← Empty entity state dictionary
{1. Knowledge Extraction and State Logging}
for each fact_i in F do
    (s, v, o, [r]) ← ParseSentence(fact_i)
    v_canon ← Normalize(v, V)
    S[s][v_canon] ← S[s][v_canon] ∪ {(o, t)}
    S[o][v_canon] ← S[o][v_canon] ∪ {(s, t)}
    if v_canon = VERB_P (Transfer) then
        S[r][get] ← S[r][get] ∪ {(o, s, t)}
    end if
end for
{2. Temporal Question Synthesis}
D ← ∅
for each entity in S do
    for each template in P do
        Q ← FillTemplate(template, entity)
        Actions ← RetrieveActions(S, entity, template)
        T ← GenerateTimeline(Actions, length = |F|)
        if CountChanges(T) > 0 then
            // Save only the ones with more than one change
            D ← D ∪ {(Q, T)}
        end if
    end for
end for
return D
```

Dataset Construction Process We sampled 12 examples from the original BABILong dataset (Kuratov et al., 2024), each containing a sufficient number of facts (average 87) to generate questions that capture knowledge transitions, with an average of 4.7 answer changes per question (ranging from 2 to 20). Algorithm 1 illustrates how we construct the four question types, where the question templates $\mathcal{P}$ are listed in Table 8 and the canonical verb groups $\mathcal{V} \in$ {Move, Acquire, Discard, Transfer}. The algorithm can be easily applied to generate additional questions whenever a sufficient number of transitional facts are available. To avoid potential confusion between the novel content and original bAbI facts, we modified location and character names. Texts are split into chunks of 2k tokens using the GPT-NeoX tokenizer.

Data Statistics and Question Types of OAKS-B Tracking questions focus on a single state but involve frequent fact updates, experiencing 9 state updates per question, while others have 4.6 on average. Comparison and bridge questions require reasoning over multiple context chunks, while counting questions require tallying actions or occurrences. Representative examples and the distribution of question types are provided in Table 9. Overall, the dataset contains a total 1.2k questions, context split into 65 chunks of 2k tokens and including 87 facts on average, and the average number of answer changes per question is 4.7.

B.1.3 OAKS-Novel

Book Selection We utilize novels as the foundational context for generating questions in OAKS-Novel, as they present multiple entities whose states evolve dynamically alongside the narrative. Moreover, literary narratives frequently incorporate flashbacks, temporal jumps, and speculative future scenarios, all of which make it difficult for models to accurately track and integrate temporal dynamics. The evidence for each question is subtly woven into the text, mirroring the nuanced, context-rich situations language models face in real-world use.

Category	Question Template
Tracking Questions	Where is [SUB]? Who is holding [OBJ]? Who gave the [OBJ] to someone else?
Counting Questions	How many times has [SUB] moved to [PLACE]? How many times has [SUB] moved? How many people have visited [PLACE]? How many times has [SUB] picked up [OBJ]? How many unique objects has [SUB] picked up? How many times has [SUB] dropped [OBJ]? How many unique objects has [SUB] dropped? How many total times has [OBJ] been picked up? How many unique people have held [OBJ]? How many total times has [OBJ] been dropped? How many unique people have dropped [OBJ]? How many times has [RECIPIENT] received anything from anyone? How many different people have given something to [RECIPIENT]? How many times has [SUB] given any object to [RECIPIENT]?
Bridge Questions	Who most recently traveled directly from [PLACE1] to [PLACE2]? Where was [SUB1] the last time [SUB2] moved to the [PLACE]? Where is the most recent location that [SUB] moved to after acquiring [OBJ]? Where is the most recent location that [SUB] moved to after dropping [OBJ]? Where is the most recent location that [SUB] acquire [OBJ]? Where is the most recent location that [SUB] drop [OBJ]?
Comparison Questions	Has [SUB1] or [SUB2] visited more distinct places? Which location did [SUB] visit more often, [PLACE1] or [PLACE2]? Who picked up a greater number of distinct objects, [SUB1] or [SUB2]? Who dropped a greater number of distinct objects, [SUB1] or [SUB2]? Who picked up a greater number of objects, [SUB1] or [SUB2]? Who dropped more objects, [SUB1] or [SUB2]?

Table 8: Question templates used for OAKS-B dataset construction. The slots [SUB], [OBJ], [PLACE], and [RECIPIENT] are dynamically filled based on the entities present in the narrative facts.

To ensure a sufficient density of state transitions in a story, we select novels from adventure, mystery, and science-fiction genres, which are plot-driven or narratively rich. Full list of the curated books and the statistics of question and context are provided in Table 10.

We used artifacts from Project Gutenburg for public domain datasets, which have copyright under *Public domain in the USA*. For copyright, we purchase the electronic copies of books and convert epubs to text files preserving all information including front and back matter following previous works using novels (Kim et al., 2024c; Chang et al., 2024).

Dataset Construction Process After obtaining raw electronic copies of the books and extracting the main narrative text[8], we segmented the text into chunks of approximately 2,000 tokens using the gpt-neox tokenizer. To preserve narrative coherence, sentences separated by newline characters in the original text were kept within the same chunk whenever possible, preventing semantically related content from being split across chunks.

After annotation, each question included between 5 and 15 answer options (5.5 on average), comprising both correct and distracting options. By default, all questions included the option "We cannot answer this question at this point," which was intended to be selected only before the relevant information appears in the narrative. Once a valid answer option becomes available, this option never becomes an answer. The order of answer options was randomized when finalizing the dataset.

Representative examples of the annotated questions and answer transitions are provided in Table 11.

QA draft generation Initial drafts of questions and answer options for each book were generated using Gemini 2.5 Pro (Gemini Team et al., 2025). To ensure broad coverage of the narrative and to capture diverse aspects of the story, we adopted a two-step generation procedure. First, we identified

[8]We removed non-content elements such as titles, author names, tables of contents, and other front or back matter, as these could interfere with model evaluation by hinting prior knowledge acquired during pretraining.

Type	Percentage	Description	Example
Tracking	7%	Questions requiring only a single piece of evidence.	*Where is Daniel?*
Counting	28%	Questions requiring multiple pieces of evidence and simple counting.	*How many times has Sandra moved?*
Bridge	30%	Questions requiring multiple pieces of evidence.	*Who most recently traveled directly from office to kitchen?*
Comparison	35%	Questions requiring multiple pieces of evidence and simple comparative reasoning.	*Who dropped more objects, Sandra or Mary?*

Table 9: Overview of Question Types, Statistics, and Examples for OAKS-BABI dataset. This table describes the four main categories of questions found in the dataset.

Copy. Status	Title	Author	Cont. Len(k)	Num. of Cs	Num. Qs	Avg. Num. options	Avg. Num. State Trans.
Copy	And Then There Were None	Agatha Christie	77	40	27	6.4	4.9
	Murder on the Orient Express	Agatha Christie	84	44	37	5.7	3.8
	Never Let Me Go	Kazuo Ishiguro	126	66	18	5.5	4.4
	The Book Thief	Markus Zusak	167	86	20	5.6	5.0
	Flowers for Algernon	Daniel Keyes	117	61	26	5.8	4.8
	The Kite Runner	Khaled Hosseini	149	78	23	5.3	4.4
	Eragon	Christopher Paolini	220	113	25	5.0	4.3
	Lord of The Flies	William Golding	86	45	18	5.1	4.0
	Tomorrow, and Tomorrow, and Tomorrow	Gabrielle Zevin	185	95	27	5.4	4.6
	Magpie Murders	Anthony Horowitz	197	103	21	5.1	3.3
	Before I Go to Sleep	S. J. Watson	146	76	17	5.3	5.5
	The Plot	Jean Hanff Korelitz	124	65	12	5.4	4.3
	The Silent Patient	Alex Michaelides	110	57	11	5.6	3.5
	Dark Matter	Blake Crouch	125	65	18	5.6	5.7
	The Chronicles of Narnia: The Lion, the Witch and the Wardrobe	C. S. Lewis	48	26	25	5.1	4.2
	The Maze Runner	James Dashner	140	73	23	5.0	4.4
	1Q84	Haruki Murakami	553	286	17	5.1	4.6
	Harry Potter and the Sorcerer's Stone	J. K. Rowling	115	59	23	5.5	4.0
	Harry Potter and the Prisoner of Azkaban	J. K. Rowling	166	85	26	5.1	3.7
	Harry Potter and the Goblet of Fire	J. K. Rowling	288	148	21	5.6	4.1
Public	Frankenstein; Or, The Modern Prometheus	Mary Shelley	100	51	29	6.6	6.7
	Pride and Prejudice	Jane Austen	171	87	33	5.0	4.1
	Little Women; Or, Meg, Jo, Beth, and Amy	Louisa May Alcott	273	139	30	6.3	5.8
	Adventures of Huckleberry Finn	Mark Twain	164	83	21	6.0	4.9
	Anne of Green Gables	L. M. Montgomery	148	76	25	5.8	5.7
	Peter Pan	J. M. Barrie	68	35	23	5.1	4.2
	The Scarlet Letter	Nathaniel Hawthorne	100	52	21	5.1	4.1
	The Adventures of Tom Sawyer, Complete	Mark Twain	105	54	21	5.3	4.9
	Treasure Island	Robert Stevenson	98	51	19	5.8	4.9
	Gulliver's Travels into Several Remote Nations of the World	Jonathan Swift	142	72	12	6.6	6.4
	The Life and Adventures of Robinson Crusoe	Daniel Defoe	157	80	26	5.6	5.1
	Around the World in Eighty Days	Jules Verne	95	49	19	7.1	6.8
	The Phantom of the Opera	Gaston Leroux	126	64	21	5.3	4.7
	The murder of Roger Ackroyd	Agatha Christie	108	56	24	5.1	4.2
	A Journey to the Centre of the Earth	Jules Verne	121	62	10	5.8	5.2
	Oliver Twist	Charles Dickens	236	119	31	5.5	4.6
	Dracula	Bram Stoker	217	110	26	5.9	4.6
	The Picture of Dorian Gray	Oscar Wilde	111	57	24	5.1	4.0
	The Secret Garden	Frances Burnett	111	57	20	5.1	4.3

Table 10: Dataset statistics for selected books, including metadata and annotation details. Each column represents Copyright Status, Title, Author, Context Length, Number of Context Chunks, Number of Questions, Average Number of Options, and Average Number of State Transitions.

Example Question 1: What mode of transportation are Phileas Fogg and Passepartout currently utilizing? *Source: Around the World in Eighty Days*	(A) Train from London to Paris (B) The steamer 'Mongolia' (C) Train from Bombay to Calcutta (D) An elephant (E) Train through India (F) The steamer 'Rangoon' to Hong Kong (G) The question cannot be answered at this point in the story.

Chunk	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	···
Answers	G	A	A	B	B	B	B	C	C	D	D	D	D	E	E	F	F	F	F	···

Example Question 2: What is Elizabeth's opinion of Mr. Darcy? *Source: Pride and Prejudice*	(A) He is proud and disagreeable. (B) She is ashamed of her prejudice and feels respect for him. (C) She loves him. (D) He is a dishonest man who does not keep his word. (E) She is offended by his perception of her. (F) The question cannot be answered at this point in the story.

Chunk	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	···
Answers	F	F	F	A	A	A	A	A	A	D	D	D	D	E	E	E	A	B	B	···

Example Question 3: What is Victor Frankenstein's primary goal or motivation at this point in the story? *Source: Frankenstein*	(A) To learn the secrets of nature and, specifically, the principle of life. (B) To escape the memory and consequences of his work. (C) To destroy the Monster as an act of vengeance. (D) Recovering from his illness (E) Living in regret and remorse because his creation murdered his brother William. (F) To create a companion for the Monster. (G) The question cannot be answered at this point in the story.

Chunk	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	···
Answers	G	G	A	A	A	A	D	D	D	D	E	E	E	C	B	B	B	B	B	···

Table 11: OAKS-Novel Example Questions, Answer Choices, and Answer Evolution over Book Chunks. *Note: The Answers shown in the diagram are for demonstration purposes only and may not accurately reflect the ground truth answers for the listed chunks.*

the main entities in each novel—such as central characters or key objects—by prompting the model with the full text of the book (see Prompt 1). Second, for each identified entity, we generated questions that synthesize information distributed across multiple chunks of the narrative. These questions were designed to either require multi-hop reasoning or track states that evolve over the course of the story (see Prompt 2).

Reason of choosing MC While answers in OAKS-B are typically stated explicitly, open-ended generation for OAKS-N often yields many valid surface forms for the same underlying answer. This makes automatic evaluation less reliable (Bai et al., 2024; Wang et al., 2024). To enable consistent scoring, we therefore cast OAKS-N as a multiple-choice QA task. Each question has at least five options, including distractors, to keep the difficulty consistent and to challenge models that have not closely tracked the narrative details.

Manual Curation and Quality Control To ensure that each question is objectively answerable at any point in the narrative, annotators filtered or refined low-quality questions, created new questions, aligned answer options with the source text, and annotated explicit supporting evidence for each state transition. Through this rigorous process, only 55% of the initial questions were retained or reformulated as high-quality.

Discarded questions mostly resembled conventional reading-comprehension questions that are answerable only after reading the entire book. The removed questions primarily fell into three categories: (i) questions that merely stitched together information from multiple chunks thus reduced to evidence searching rather than state tracking; (ii) questions that exhibited only a single state change but did not require multi-chunk reasoning or multi-hop inference; and (iii) questions whose answer options were unsupported by the surrounding context or appeared simultaneously within the same chunk.

Beyond filtering out low-quality questions, the annotators and authors extensively revised the remaining questions to ensure that each question was

Prompt 1: Identifying Main Entity

You are an AI assistant specializing in narrative analysis and entity extraction. Your task is to analyze the novel and identify the most important characters and objects, central to the narrative.

Your response must only contain the names of the entities, separated by a | character. Do not include labels, explanations, or any other text.

Example Output:
Jason Hill | Maria | The Sapphire | Captain Eva | The Northern Village

objectively answerable at any point in the narrative and that exactly one answer option was correct at each chunk. For example, questions that asked about a state's evolvement were revised to ask the status at specific point in the story. (e.g., *How does Mr. Darcy's assessment of Elizabeth Bennet's appearance evolve?* → *What is Mr. Darcy's latest assessment of Elizabeth Bennet's appearance?*) Similarly, when a state was revealed cumulatively over time , we reformulated questions to target the most recent revelation(e.g., *What are the different names or identities Erik is known by?* → *What term has been most recently introduced to describe the man who lives at the opera?*), rather than the entire accumulated state, ensuring that exactly one answer option is correct at each point in the narrative.

Answer options were also carefully curated: unsupported options were removed, contextually necessary options were added, and cases where multiple options appeared within the same chunk were consolidated. This process ensured that the answers remain well-defined and that only a single option was valid for each chunk. Throughout this process, the validity of the question or answer options and the correctness of the selected answers were jointly reviewed and discussed by at least two annotators or authors.

Human Annotation Via Upwork, we hired 18 experienced freelancers who were native English speakers from USA and had already read the selected books. Compensation per book ranged from $50 to $850, depending on book length and the number of initial draft questions, with a total annotation cost of $17,400. Beyond the core question-answer annotation task, compensated at approximately $20 per hour with an average workload of 14 hours per book, we additionally provided payment for high-quality question generation and substantial question–answer revisions.

Annotators were provided with (i) the novel text segmented into chunks and (ii) a spreadsheet containing the questions and answer options. For each question, annotators labeled the correct answer at every chunk, and revised or removed questions and answer options. Whenever an answer changed from previous answer, annotators were required to copy and paste the exact evidence sentence from the corresponding chunk that justified the answer transition.

The authors subsequently conducted a thorough verification of all annotated files to ensure consistency and correctness. This included checking that (i) each question had exactly one valid answer per chunk, (ii) the cited evidence explicitly appeared in the specified chunk and correctly supported the selected answer, (iii) the cited evidence aligns with the chunk the answer was chosen, and (iv) each question contained at least five answer options, including distracting options.

The full annotation instructions provided to the annotators are included in Instruction 1 to 4. To ensure high-quality annotations and alignment with our research goals, we provided the annotators with an explanation on academic purpose of the study. The annotation process was conducted transparently through the Upwork platform, ensuring fair labor practices, after internal ethics review.

B.1.4 Dataset Safety

To ensure data privacy and ethical standards, for OAKS-B, we modified all location and character names using randomized, non-existent strings to ensure they do not refer to real-world individuals or locations. The factual sentences in the widely used original bAbI benchmark (Weston et al., 2016) consist of mundane descriptions of daily activities

and contain no offensive content. For OAKS-N, the foundational contexts are derived from existing, publicly available novels.

B.2 Evaluation Setups and Metrics

B.2.1 Evaluation Criteria

After extracting the final answer from the model's prediction, we compare it against the ground-truth answer using normalized text. To ensure a robust benchmark, correctness is assessed using an exact-match criterion. For OAKS-N, where models are prompted to respond with a single character corresponding to an answer option, we extract and evaluate that character directly.

For OAKS-B, to mitigate the ambiguity of open-ended generation, we conditionally allow multiple equivalent answer forms only under the following cases: (i) Counting-type questions: we accept both numeric and textual representations (e.g., once, twice, three times). (ii) Counting-type questions before the relevant information is revealed: we accept both Unknown and 0. (iii) Comparison-type questions before valid evidence is encountered: we accept both Unknown and Same.

B.2.2 Metrics

For OAKS-BABI and OAKS-Novel, we evaluate a model with Accuracy and analyze fine-grained behavior over three additional metrics: Acquisition Latency (AL), Distraction Susceptibility (DS), and Phase Miss rate (PM).

Accuracy Accuracy measures the proportion of time intervals where the model predicts the correct answer for a given question. We then average the score over all questions. We provide a formal definition below.

Given our dataset consisting of C chunks $\mathcal{C} = \{c_i\}_{i=1}^{C}$ and a question set $\mathcal{Q} = \{q_j\}_{j=1}^{Q}$ with corresponding ground truth answers $\mathcal{A} = \{a_{i,j}\}_{j=1,i=1}^{Q,C}$, a model M is asked to predict an answer given a set of chunks $\mathcal{S}_t = \{c_i\}_{i=1}^{t}$ up to a certain interval t. Let the model's prediction be $p_{j,t} = M(q_j, \mathcal{S}_t)$.

Then the Accuracy of model M is defined as:

$$\textbf{Accuracy} = \frac{1}{Q}\sum_{j=1}^{Q}\frac{1}{C}\sum_{i=1}^{C}\mathbb{1}\left[p_{i,j} = a_{i,j}\right] \quad (1)$$

Acquisition Latency (AL) AL describes how quickly or slowly it takes the model to adapt to a new state relative to the length of all sequence. For each phase, we count the number of incorrect predictions before a correct prediction (*lag* intervals) and obtain the proportion of the whole intervals. Then, we compute the average over all the questions.

Mathematically,

- let N_j be the total number of distinct phases for question q_j,
- let $T_{k,j}$ be the k-th phase for q_j and $|T_{k,j}|$ its duration,
- let $a_{t,j}$ be the ground truth answer at interval t for question q_j.

Define

$$\tau_{k,j} = \min\{t \in \{1, \ldots, |T_{k,j}|\} \mid p_{t,j} = a_{t,j}\}$$

as the first time step in phase $T_{k,j}$ where the models answers q_j correctly. If the model is never correct, we define $\tau_{k,j} = 0$.

AL metric is defined as:

$$\textbf{AL} = \frac{1}{Q}\sum_{j=1}^{Q}\frac{1}{C}\sum_{k=1}^{N_j}(\tau_{k,j} - 1)\cdot\mathbb{1}[\tau_{k,j} > 0] \quad (2)$$

Distraction Susceptibility (DS) DS measures the distraction rate of a model. For each phase, we quantify DS by counting the number of incorrect predictions after its first correct prediction (number of *lapses*) and average the ratio of all intervals and over all questions.

DS metric is defined as:

$$\textbf{DS} = \frac{1}{Q}\sum_{j=1}^{Q}\frac{1}{C}\sum_{k=1}^{N_j}\sum_{t=\tau_{k,j}+1}^{|T_{k,j}|} d_{j,k,t}, \quad (3)$$
$$d_{j,k,t} = \mathbb{1}[p_{t,j} \neq a_{t,j}]\cdot\mathbb{1}[\tau_{k,j} > 0]\,.$$

Phase Miss rate (PM) PM measures the rate at which a model completely fails to capture the correct state throughout an entire phase. We quantify PM by summing up the duration of each phase where the model missed completely and averaging the ratio of all intervals over all questions.

PM is defined as:

$$\textbf{PM} = \frac{1}{Q}\sum_{j=1}^{Q}\frac{1}{C}\sum_{k=1}^{N_j}|T_{k,j}|\cdot\mathbb{1}[\tau_{k,j} = 0] \quad (4)$$

B.2.3 Statistical Reliability

Due to the incremental nature of our evaluation, where models must perform inference at every interval, the total number of required inferences is

Avail-ability	Model		Architecture			Context Length		Exp. Length
	Model Family	Full-name	Size (Active)	Type	Attention	Default	Expanded	
Open	Qwen	Qwen3-4B-Instruct-2507	4B	Dense	GQA	262k	-	262k
		Qwen2.5-7B-Instruct	7B	Dense	GQA	32k	131k	131k
		Qwen3-8B	8B	Dense	GQA	32k	131k	131k
		Qwen3-30B-A3B-Instruct-2507	30B (3B)	MoE	GQA	262k	-	262k
		Qwen3-30B-A3B-Thinking-2507	30B (3B)	MoE	GQA	262k	-	262k
		Qwen3-Next-80B-A3B-Instruct	80B (3B)	MoE	Hybrid	262k	1M	262k
		Qwen3-235B-A22B-Instruct-2507	235B (22B)	MoE	GQA	262k	1M	133k
	GPT-OSS	gpt-oss-20b	20.9B (3.6B)	MoE	GQA	131k	-	131k
		gpt-oss-120b	116.8B (5.1B)	MoE	GQA	131k	-	131k
	Gemma	Gemma 3-4b-it	4B	Dense	GQA	131k	-	131k
		Gemma 3-27b-it	27B	Dense	GQA	131k	-	131k
Propr-ietary	Gemini	Gemini 2.5 Flash	-	-	-	1M	-	1M
		Gemini 2.5 Pro	-	-	-	1M	-	1M
		Gemini 3.0 Pro	-	-	-	1M	-	1M

Table 12: Model Architecture and Performance Settings. Attention architecture of Qwen3-80B is Hybrid of Gated DeltaNet and Gated Attention)

exceptionally high. Specifically, for each model, the total inference count is calculated as $D \times C \times Q$, where D represents the number of documents(books), C the number of chunks per document, and Q the number of questions per interval. This results in approximately 78k inferences for OAKS-BABI and 67k for OAKS-Novel per model. The baseline experiment on Qwen3-30B model required approximately 125 GPU hours on Nvidia A100 GPUs. Given the substantial computational cost, we report results from a single comprehensive run for each model rather than multiple trials with different seeds. However, the sheer volume of evaluation points across thousands of unique context-question pairs provides a high degree of consistency and statistical significance for our comparative analysis.

C Experimental Setup

C.1 Base Models

We assess the proposed dataset on a total of 14 language models, including both open-source and proprietary systems, across a wide spectrum of model scales. Because long-context capacity and architectural design vary substantially across models, we summarize these characteristics in Table 12.

C.1.1 RAG

Retrieval is restricted to chunks from previous time intervals, with chunk indices added to preserve order (i.e., no access to future information). Unless otherwise specified, we retrieve the total 30 relevant memory chunks from top, including both directly matched chunks and those linked through the framework's internal structure.

C.1.2 Agentic Memory Systems

HippoRAG-v2 (Gutiérrez et al., 2025) is a graph-based retrieval framework built on Personalized PageRank. MemAgent (Yu et al., 2025) is designed for long-context processing with linear computational complexity, partitioning inputs into chunks and incrementally updating memory by combining prior memory with newly observed chunks at each timestamp, trained using GRPO. A-Mem (Xu et al., 2025b) is an agentic memory system inspired by the Zettelkasten method and organizes memory as an interconnected knowledge network through dynamic indexing and bidirectional linking.

C.1.3 Context Representation

Baseline When evaluating on OAKS-N, for models whose maximum sequence length is shorter than 1M tokens, we adopt a rolling-window strategy that always fills the model's full context length, discarding the earliest chunks as new chunks are appended. For ultra-scale models such as Qwen3-235B, we cap the maximum model length at 133k tokens, since inference with the full length cannot be executed on a single 8 H100 node. The maximum context length used for each model in the baseline concatenation setting is reported in the final column of Table 12.

RAG To evaluate the models' tracking performance under a Retrieval Augmented Generation(RAG) setting, we structured the retrieval pro-

cess and prompt formatting as follows:

- **Indexing and Retrieval**: We build a vector index where each entry is formatted as "# chunk index : {chunk_idx}, context: {chunk_text}". For a given question q_j at interval t, we perform a similarity search using a query that incorporates the chunk index to maintain structural consistency with "# chunk index : {chunk_idx}, question: {question}"
- **Prompt Construction**: The top k retrieved chunks are then prepended to the model's input prompt. To ensure the model can distinguish between different retrieved fragments, we use a structured list format as "- Retrieved context:\n# chunk index : {chunk_idx}, context: {context}\n# chunk index : {chunk_idx}, context: {context}"

For RAG+R.W, the RAG component is only activated when the total available context exceeds the current window size ($t > w$), where w denotes the window size, utilizing the historical chunks beyond the window for retrieval. The context is distinguished from each other with the format: "- Retrieved context:\n# chunk index : {chunk_idx}, context: {context}\n# chunk index : {chunk_idx}, context: {context}\n\n # Recent Chunks: {context}\n{context} " and the interval information is embedded into the question as well "Current Head Index : {chunk_idx}, question: {question}"

C.1.4 Inference Parameters

For inference, we generally follow the best practices recommended by the model providers when available. As a default configuration, we adopt the settings used for the Qwen3 series: a temperature of 0.7, top-p of 0.8, and top-k of 20. Some models use provider-specified configurations that differ from this baseline, including Qwen3-Next-80B (temperature 0.6, top-p 0.95, top-k 20) and Qwen3-30B-Thinking (temperature 0.6, top-p 0.95, top-k 20). For Gemini 2.5 Flash and Gemini 2.5 Pro, we use the default setup of temperature 0.0, top-p 0.95, and top-k 40. We generate up to 4096 tokens for all models. When querying Gemini with the thinking process enabled, we use the model's default mode, where it automatically budgets up to 8192 thinking tokens. For Gemini 3.0 Pro, we extended the maximum generation tokens to 32,768 in cases where the model exhausted the initial 4k-token limit in thinking mode without returning a final answer.

When the concatenated context exceeds a model's default context length and the model supports context extension via YaRN, we expand the maximum sequence length accordingly, following the procedures specified in the official documentation.

Experiments were conducted using 4 or 8 A100 80G GPUs, 8 H100 80G GPUs, or 4 H200 140G GPUs based on the model size and availability.

C.1.5 Inference Prompt

The prompts used to evaluate models on OAKS-B and OAKS-N are provided in Prompt 3 and Prompt 4, respectively. The task description is shared across the two benchmarks; however, the only difference lies in (i) how the "not answerable" option is defined and (ii) the output format. Specifically, OAKS-B requires open-ended generation, whereas OAKS-N requires selecting from predefined answer options. Our prompts are adapted from the original prompt used in the BABILong benchmark (Kuratov et al., 2024) and follow the best practice described in official document of Qwen3. To ensure deterministic evaluation, especially for models prone to divergent outputs, we appended a strict formatting constraint to the system prompt, where the model was instructed to conclude every response with a standardized template: "## Answer: [CHAR]".

D Evaluation Result

D.1 Relationship with reasoning/thinking ability

Table 14 presents the performance results on OAKS-N. Unlike the results observed in OAKS-B, the activation of Thinking mode in OAKS-N leads to a performance degradation. While OAKS-B is a synthetic dataset specifically designed to require explicit multi-evidence integration, OAKS-N necessitate implicit multi-hop reasoning woven into a naturalistic narrative. We hypothesize that inference-time scaling is most effective for tasks with high structural complexity, such as those requiring the simultaneous tracking of multiple states as seen in OAKS-B, rather than the nuanced linguistic extraction required by OAKS-N.

D.2 Analysis on Performance of RAG and RW

No clear winner on optimal context construction strategies. Table 13 compares different con-

	OAKS-B				OAKS-N			
Strategy	All	Sprs.	Mod.	Freq.	All	Sprs.	Mod.	Freq.
Base	35.8	38.5	37.3	29.4	62.8	71.2	63.7	57.1
RAG	**37.8**	**40.1**	**39.2**	**32.0**	61.0	68.9	63.0	54.9
R.W.	36.4	38.8	38.3	29.9	62.6	69.2	63.6	58.0
RAG+R.W	36.7	39.3	38.1	30.7	**64.6**	**72.1**	**64.7**	**60.1**

Table 13: Accuracy (%) of Qwen3-30B on OAKS-B and OAKS-N under different context construction strategies. We compare baseline concatenation, RAG with top-30 retrieved chunks, a rolling window of the most recent 30 chunks (RW), and their combination (RAG+RW; top-15 retrieved and 15 most recent).

Model	Think	All	Sparse	Moderate	Frequent
Qwen3	-	**62.8**	**71.2**	**63.7**	**57.1**
30B	✓	61.8	70.6	61.1	57.1
Gemini 2.5	-	**67.3**	**77.3**	**68.2**	**60.9**
Flash	✓	65.6	75.2	65.6	60.0

Table 14: Accuracy (%) of Qwen3-30B and Gemini 2.5 with RAG on OAKS-N by answer change frequency with and without Thinking Mode.

Models	Size	LongBench
Qwen2.5	7B	30.0
Gemma 3	4B	29.4
GPT-OSS	20B	14.8
Gemma 3	27B	33.6
Qwen3	30B	32.6

Table 15: Performance on LongBench-V2.

text construction strategies under long-context constraints, including the Base setting, RAG, Rolling Window (RW), which retains only the most recent 30 chunks, and their combination (RAG+RW). While each strategy improves performance over the Base setting in most cases, no single approach consistently dominates across datasets. Specifically, RAG achieves the best performance on OAKS-B (+2.0%), suggesting that relevance-based evidence localization is effective when salient evidence is clearly distinguishable from background context. In contrast, on OAKS-N, RAG degrades performance (-1.8%), whereas the combined RAG+RW strategy performs best (+1.8%). We hypothesize that this is because novels are narrative-heavy contexts with complex temporal structure. Thus, even when retrieval is successful, models may struggle to effectively integrate retrieved passages, while the additional RW component preserves temporally aligned context, facilitating better understanding and utilization of accumulated long-term history (Han et al., 2025; Gutierrez et al., 2024; Lee et al., 2024).

Performance by number of context chunks in RAG and Rolling Window Figure 5 illustrates the performance of the retriever, Qwen3-Embedding-0.6B (Zhang et al., 2025), reporting the Pass@k metric as a function of the number of retrieved chunks. Given that our dataset necessitates the synthesis of multiple pieces of evidence, a single chunk rarely contains the complete ground truth. For the purpose of this analysis, however, we proxy the gold truth as the chunk of the state transition within the current phase. We observe that retriever performance increases monotonically, approaching near-perfect recall as k reaches 60.

Figure 6 and Figure 7 present the performance as a function of the number of chunks in RAG and RW on OAKS-B and OAKS-N, respectively. In both settings, increasing the number of chunks initially improves the performance but peaks around 30 chunks.

D.3 Difference by dataset

We observe that the average score for OAKS-B (39.4%) is lower than that of OAKS-N (57.5%). We conjecture that the multiple-choice format of OAKS-N introduces a restricted answer space,[9] and that the narratives within OAKS-N may still be susceptible to the influence of prior knowledge bias. The lower performance on OAKS-B may be attributed to its synthetic design, which allows for the creation of intrinsically more complex and difficult tasks.

E Analysis

E.1 Fine-grained Analysis of Predicted Knowledge Transition Behavior

In this section, we provide an additional detailed analysis based on Table 5 in Section 6.1, which summarizes the results for representative models. Table 16 and Table 17 show the analysis over OAKS-B and OAKS-N, respectively. Figure 8 shows a schematic illustration of model behaviors.

Correct transitions are easier than correct answers: Aggregating across models, we observe that models tend to correctly predict whether a transition occurs compared to producing the correct

[9] A random baseline can achieve an expected accuracy of 18.6%.

answer itself. On average, correct transition behaviors occur more often (31.2%) than incorrect ones (18.8%), whereas answer correctness is lower overall (22.8% correct vs. 27.2% incorrect). This indicates that identifying when to update knowledge is an easier subproblem than determining what the updated answer should be.

Dominant behaviors when answer is correct? We analyze the model behavior among the cases where the answer is correct (✓). When the transition prediction is correct, `Adaptability` occurs more frequently than `Stability`, indicating that maintaining stable knowledge is more challenging than updating knowledge at the correct time. We hypothesize that this difficulty arises due to long and complex, multiple interacting facts within the given context, which increases the likelihood of spurious updates.

Dominant behaviors when answer is incorrect? We analyze the model behavior among the cases where the answer is incorrect (✗). In these cases, `Volatility` and `Maladaptation` are the more frequent behaviors when the transition is incorrect or correct, respectively, indicating that erroneous answers often coincide with unnecessary or poorly timed changes. Compared to the correctly answered cases in the previous paragraph, the gap between changing and staying behaviors is smaller when the answer is incorrect (7.5 for incorrect vs. 14.6 for correct), suggesting that errors arise from a broader range of failure modes.

E.2 Correct evidence prediction does not ensure correct answer prediction

We conduct a detailed error analysis of the relation between answer and evidence prediction for OAKS-N to evaluate how well the model uses the source from the book. We use an LLM as a judge to assess whether the predicted reasoning aligns with the annotated evidence and to check whether the model relies on its own knowledge.[10] The largest error category is when both answer and evidence are incorrect (47.3%), indicating that failures frequently co-occur. Fully correct predictions, where both answer and evidence are correct, account for 18.8% of cases. In 14.2% of cases, the model produces the correct answer despite incorrect evidence, typically by relying on its own parametric knowledge or focusing on an incorrectly related passage. When evidence is correct, the model still produces an incorrect answer in 19.7% of cases, often due to confusion among multiple answer options, highlighting that even correct evidence does not guarantee a correct answer and is consistent with the overall lower accuracy of answer prediction compared to evidence prediction. Taken together, these observations indicate that answer reasoning is a critical ability for solving this task. We used Gemini 2.5 Pro as Judge. Prompt 4 shows the prompt used for LLM Judge.

[10]Cases in which the model produced an answer directly without explicit reasoning were removed from this analysis

E.3 OAKS is not solvable with simple long context understanding ability

We analyze the correlation between performance on long context tasks and on OAKS. To measure long-context ability, we use LongBench-v2 (Bai et al., 2024), a widely adopted benchmark for evaluating models' performance on long context inputs (Table 15). Across five models of similar size, the Pearson correlation between LongBench-v2 performance and performance on OAKS-B and OAKS-N is 0.69 and 0.34, respectively. When analysis over the *Frequent* subset of OAKS-B and OAKS-N, the correlation drops to 0.45 and 0.30. This suggests that while OAKS benefits from a model's long context ability, as contexts of OAKS are usually long, it also requires additional capabilities: adapting to dynamic knowledge online and accurately tracking evolving information. Similarly, the analysis in Section 6.4 shows that performance tends to degrade as the time interval increases, further highlighting that simple long-context understanding is insufficient to solve OAKS.

F Ethical Considerations / Potential Risks

Our study is primarily restricted to English narratives, which may limit the generalizability of our findings to other languages. OAKS incurs substantial computational cost, which would indicate increased energy consumption. We hired human annotators when constructing OAKS-N; although we applied extensive filtering and verification procedures, potential annotation errors may remain due to the inherent subjectivity of human judgment.

G Usage of LLM

In our research, we employed Large Language Models (LLMs) for initial data generation and writing assistance. In OAKS-N, as explained in Section 3.1 and Appendix B.1.3, initial drafts of

Think	Models	Size	GT Phase Transitions (Change)				No GT Transition (Stay)			
			Adaptability (C / ✓)	Maladaptation (C / ✗)	Prescience (S / ✓)	Stubbornness (S / ✗)	Lag (C / ✓)	Volatility (C / ✗)	Stability (S / ✓)	Obstinacy (S / ✗)
✗	Qwen3	4B	28.4	26.2	2.5	42.9	4.6	23.6	21.4	50.4
		8B	36.5	48.2	3.7	11.6	11.0	57.9	21.5	9.6
		30B	34.3	33.6	9.7	22.4	8.9	36.7	26.3	28.1
		80B	36.0	29.5	9.2	25.3	9.5	31.3	31.3	28.0
		235B	39.8	27.9	11.3	21.0	12.0	31.6	34.5	21.9
	Qwen2.5	7B	28.6	37.2	5.9	28.3	9.2	46.0	14.8	30.0
	GPT-OSS	20B	33.3	54.6	3.8	8.2	9.3	68.5	12.0	10.2
		120B	39.1	40.8	5.5	14.6	12.8	46.5	24.2	16.5
	Gemma 3	4B	26.5	44.0	7.4	22.1	7.9	48.7	15.5	27.9
		27B	31.6	28.6	11.9	27.9	12.2	27.0	25.2	35.6
	Gemini 2.5	Flash	36.3	17.0	17.2	29.5	7.5	16.9	35.0	40.7
		Pro	37.2	20.2	15.4	27.2	8.3	20.6	33.8	37.2
✓	Qwen3	30B	39.6	34.6	7.7	18.2	13.0	37.7	30.4	18.9
	Gemini 2.5	Flash	47.5	23.8	15.0	13.7	12.4	27.5	43.3	16.8
		Pro	49.3	25.3	14.1	11.3	15.6	27.3	44.4	12.6

Table 16: Analysis of model knowledge tracking behavior on OAKS-B. The table shows the average occurrence rate of specific tracking behaviors across all time intervals. Results are partitioned by the ground truth (GT) state: whether the answer **Change** from the previous interval or **Stay** the same. Since Stay intervals are more frequent(94% of all intervals), rates are averaged within each GT category to sum to 100%. The second row shows the behavioral labels (e.g., Maladaptation) and the model's action in the sub-header (e.g., (C / ✓)): whether the predicted answer **C**hanged from the previous prediction or **S**tay, and whether the resulting answer is correct (✓) or wrong (✗).

Think	Models	Size	GT Phase Transitions (Change)				No GT Transition (Stay)			
			Adaptability (C / ✓)	Maladaptation (C / ✗)	Prescience (S / ✓)	Stubbornness (S / ✗)	Lag (C / ✓)	Volatility (C / ✗)	Stability (S / ✓)	Obstinacy (S / ✗)
✗	Qwen3	4B	51.8	15.2	7.5	25.5	8.2	20.4	38.6	32.8
		8B	53.3	18.4	6.7	21.7	8.2	23.1	43.1	25.7
		30B	54.8	9.7	11.9	23.7	5.4	10.5	57.9	26.2
		80B	55.5	8.5	12.5	23.5	4.6	8.7	60.3	26.4
		235B	55.2	9.0	11.3	24.5	5.0	9.5	60.3	25.2
	Qwen2.5	7B	27.9	39.1	4.2	28.9	12.2	31.1	19.7	37.0
	GPT-OSS	20B	48.2	23.0	5.8	23.0	10.7	33.5	33.6	22.2
		120B	46.8	23.2	8.3	21.7	11.0	23.4	43.4	22.3
	Gemma 3	4B	37.2	14.5	16.0	32.2	4.5	14.0	32.7	48.9
		27B	57.2	8.0	12.3	22.5	5.0	9.4	56.0	29.6
	Gemini 2.5	Flash	59.9	10.4	9.8	19.9	6.3	12.8	61.4	19.6
		Pro	25.4	35.2	1.2	38.2	15.7	27.0	9.4	47.8
✓	Gemini 2.5	Flash	60.8	15.7	9.7	13.8	8.5	19.5	56.7	15.4
		Pro	67.3	6.2	13.9	12.7	4.1	7.5	72.6	15.8

Table 17: Analysis of model knowledge tracking behavior on OAKS-N.

question-answer pairs were generated by Gemini 2.5 Pro, which were subject to rigorous human annotation process to filter out low-quality questions. During writing, LLMs were utilized for sentence-level refinement and grammatical polishing. All AI-generated suggestions were carefully reviewed and edited by the authors to maintain the coherency and accuracy.

Prompt 2: Generating Question about a Specific Entity

You are given a novel. Your task is to generate a set of Question, Answer, and Evidence pairs that fulfill the following criteria:

Question Criteria
- **Synthesizing Information**: Each question must require synthesizing information from at least *two* separate parts of the novel. A question that can be answered with a single local quote should not be included.
- **Types of questions**: Example topics include:
    - Tracking the evolution of a character's feelings, understanding, or situation over time.
    - Connecting a character's backstory from one chapter to their actions in another.
    - Solving a mystery for which clues are scattered across different locations.
    - Comparison questions where the relevant information appears in different parts (e.g., "Who is older, A or B?").
- **Avoid single-point questions**: Avoid questions that can be answered by quoting just one part of the novel.

Answer & Evidence Format
- **Answer**: Provide a *concise* answer in *a few words*.
- **Evidence**: Then, *support* the answer with quoted pieces of evidence from the text. These evidences should come from different chunks of the novel (not from the same chunk).

Input Schema
You will receive two inputs:
1. A JSON object: This object contains the Novel text.
2. An Entity: A string representing the main character or object to be tracked (e.g., "Jason Hill").

Output Schema
The output must be a single JSON object matching this structure.

```
[{
   "question": "What is Harry's primary Quidditch broomstick?",
   "answers": [{
      "answer": "Nimbus Two Thousand",
      "source": "One of Harry's most prized possessions was his Nimbus Two
        Thousand racing broom."
     },
     {
      "answer": "A school-owned Shooting Star",
      "source": "He had been riding one of the school brooms at team practice,
        an ancient Shooting Star, which was very slow and jerky; he definitely
        needed a new broom of his own."
     },
     ...
    ]
 },
]
```

ENTITY TO TRACK

```
{entity}
```

Instruction 1: Overall Instruction to the Annotators

Task Instruction:
We have divided the assigned book into multiple chunks. The goal of this project is to ‘collect and refine questions and answers that reflect “dynamic changes throughout the storyline”.‘ In other words, tracking the changing answers in each chunk. For example, if the main character moves frequently, a question like ‘where is the main character living?’ will have different answers at different points in the story, depending on the events that have occurred so far.

Your responsibilities:
You will be working with an Excel file that contains text from a book, along with questions and answer options. Please fill in the Excel file according to the provided guidelines. See [Guidelines on how to fill in Excel] for details.

- Tasks:
1. Read the given chunk carefully.
2. After reading each chunk:
   – Answer the given multiple-choice questions based on **the content in the current chunk and all previous chunks.** If none of the existing options are appropriate, add a new one. (Read [Guidelines for Answering] for more details)
   – Copy and paste the relevant evidence snippet from the chunk to ‘source‘ tab.
3. Generate at least five new questions that fit the same dynamic-tracking criteria. (Read [Guidelines for Generating Questions] for more details)
4. Generate **distracting** answer options. (Read [Guidelines for Generating Distracting Answer Options] for more details)

Instruction 2: Guidelines for Answering

1. Use only information from the current and previous chunks. Do NOT use prior knowledge about later parts of the story.
2. Choose the most accurate answer based on the information available so far. Each question should have only **one** appropriate answer.
   - If a question appears to have multiple correct answers, and both seem valid:
     - Write "remove" for the option that has never been chosen as an answer before in Column E. Make sure you do not choose the option later on.
     - If removing an option affects later chunks, please contact us for clarification.
     - Examples
       - **[Example 1]** Question is *"Where is character A living?"* with options "Paris" and "LA". Earlier answer is *Paris* and New information states that *A lives in LA*
         → Choose LA as the correct answer and provide evidence.
       - **[Example 2]** Question is *"What is the role of A?"* with a option "nurse". In Chunk 5, it mentions A is a *nurse* and in chunk 7, it adds that *A also started working as a writer on weekends*
         → Add a new option *"nurse and writer"* and choose this as an answer.
   - Along with several other answer options, there are always two default options:
     - "Prev" : Previous answer still holds
       - **[Example]** If in previous chunks the character moved to home A, and the current chunk doesn't mention about move, then the answer to "Where is the character living" remains A and you should choose "Previous answer still holds"
     - "NA" : We cannot answer to this question at this point
       - If the answer cannot be found because the relevant information has not appeared in the story yet, choose "NA."
       - **[Example]** If the question is "Where is the character living?" and the location has not yet been revealed, the answer should be "NA".
       - Once you select any answer other than "NA", you should not choose "NA" afterwards. If none of the existing options fit, add a new answer option instead.
3. Except for "NA" or "Prev", copy and paste the relevant evidence snippet from the chunk.
4. If answering a question requires information from multiple chunks, you may select the answer only when all necessary evidence is present in the current or previous chunks. In this case, the evidence should be taken from the latest chunk—the one that contains the final piece of information needed to answer the question.
5. If the current chunk includes a flashback (a reflection on a past location), do NOT treat that as the current state.
6. If all the answer options are inappropriate, create and add new option.
7. When a question asks about a character's understanding of something, include evidence showing *where the character actually understands* the concept or situation. However, sometimes a character *hears* or *learns* about a situation but becomes confused because it contradicts their prior beliefs. In those cases, only select the answer and evidence that **explicitly show the character has reached understanding**.
   - **Valid example**: "Character A nodded in agreement after the explanation." (This shows clear understanding.)
   - **Invalid example**: "Character A listened to the explanation." (This only shows the character hearing the explanation, not understanding it yet.)

Instruction 3: Guidelines for Generating Questions

Valid question set satisfies these criteria:

1. There could be two types of questions:
   - The question tracks the evolution of a character's feelings, understanding, or situation as the story progresses. The appropriate answer to the question is different according to the relative position in the story.
   - A question that you can answer ONLY when you have information from two or more chunks.
2. For each chunk, there should be only one answer for each question.
3. Do not ask anything trivial.
   - For example, what is the color of the character A wearing?
4. The question shouldn't be subjective or ambiguous
   - For example, "who is character A's best friend?" since the definition of "best friend" is ambiguous unless explicitly mentioned in the book.
5. You have to generate at least 5 new questions over the whole book. Make sure you do the same answer, evidence annotation for these new questions as you did for original questions.

Instruction 4: Guidelines for Generating Distracting Answer Options

1. The distracting options should be plausible but incorrect, designed to challenge readers who haven't paid close attention to the details of the story.
2. Each question must have at least six answer choices in total. These two default options ("NA" and "Prev") are always included, and the additional four options should always exist. If the answer options are fewer than six, add the incorrect distracting answers to reach a minimum of six total choices.

Prompt 3: Prompt for evaluation on OAKS-B

I will give you context and a question. You need to answer the question based only on the information from the text.

Follow these rules carefully:

1. Strict Contextual Grounding: You must base your reasoning exclusively on the text provided. Do not use any external or prior knowledge of this story (even if you recognize it).
2. State Persistence: When a state is established (e.g., "The box is on the table"), you must assume that state persists unless a later part of the text explicitly changes or contradicts it (e.g., "...he moved the box into the basement.").
3. Final State Priority: Your answer must always reflect the state of the subject at the end of the provided text. If a state changes multiple times, your answer must be based only on the most recent, final version.
4. If the answer cannot be found in the context, respond with 'Unknown'. If the question is a comparison and the values are same, respond with 'Same'. Otherwise, give a concise response in 1–2 words (not a complete sentence).
5. You may do reasoning, but in the end, output your final response in the format: '## Answer: short answer'. The short answer will be parsed for an exact match.
   Example: If the question is 'Where is Sandra?' and the answer is 'kitchen', your response must end with '## Answer: kitchen'. (Do not include extraneous text like 'Sandra is in the kitchen').

Context:

`{context}`

Question:

`{question}`

Prompt 4: Prompt for evaluation on OAKS-N

I will give you context and a question. You need to answer the question based only on the information from the text.

Follow these rules carefully:

1. Strict Contextual Grounding: You must base your reasoning exclusively on the text provided. Do not use any external or prior knowledge of this story (even if you recognize it).
2. State Persistence: When a state is established (e.g., "The box is on the table"), you must assume that state persists unless a later part of the text explicitly changes or contradicts it (e.g., "...he moved the box into the basement.").
3. Final State Priority: Your answer must always reflect the state of the subject at the end of the provided text. If a state changes multiple times, your answer must be based only on the most recent, final version.
4. Answer Selection: You must choose your answer from the provided options. Do not generate your own answer. If the text does not contain the information to answer the question, you must select the option that says "We cannot answer this question at this point".
5. Output Instructions: Please show your choice in the end of the answer field with only the choice letter, e.g., "answer": "C".

Context:

`{context}`

Question:

`{question}`

Options:

`{options}`

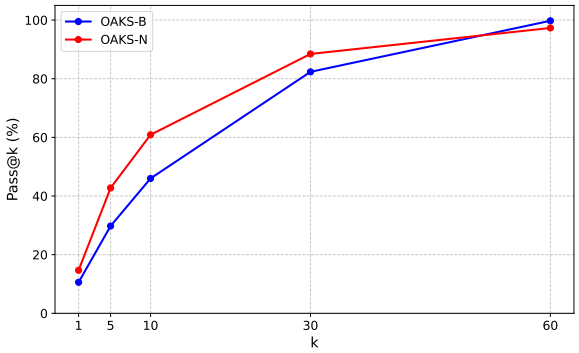

Figure 5: Pass@k performance as number of context increases for retrieval in RAG on OAKS-B and OAKS-N

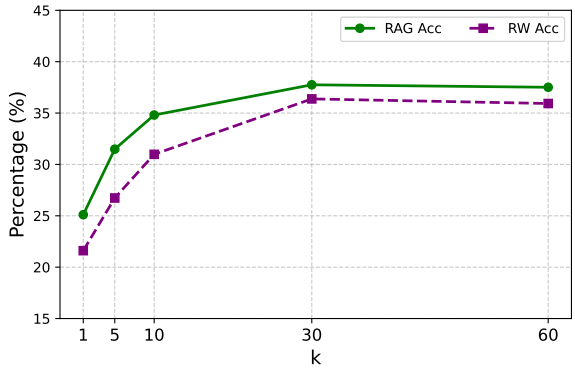

Figure 6: Accuracy (%) on OAKS-B across different numbers of context chunks in RAG and RW.

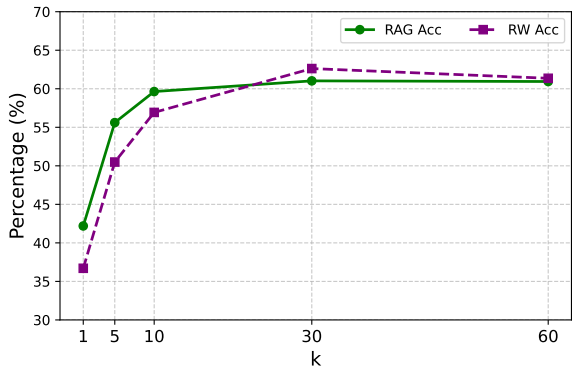

Figure 7: Accuracy (%) on OAKS-N across different numbers of context chunks in RAG and RW.

Prompt 5: Prompt for LLM Judge over evidence

You are an expert Evidence Quality Auditor. Your task is to verify if the Model's reasoning path matches the specific evidence logic found in the Ground Truth (GT) Evidence.

Evaluation Data
- **Question:** [INSERT QUESTION]
- **GT Evidence:** [INSERT GT EVIDENCE]
- **Model Output:** [INSERT MODEL OUTPUT]

Evaluation Criteria
- The "Core Anchor" Alignment: Identify the specific factual anchor in the GT Evidence (e.g., a specific location or name). The model must use this anchor as its primary basis for the answer.
- Narrative Context Allowance: The model MAY use external knowledge of the source material (e.g., "In Chapter 1," "the inciting incident," "the parchment") to frame or organize its reasoning. This is seen as helpful context.
- The "Source Primacy" Rule (Strict): The model MUST NOT use its own knowledge to contradict, correct, or bypass the GT Evidence. Phrases like "Based on my own knowledge, the text is wrong" or reaching a conclusion that ignores the GT facts in favor of external ones must be marked as FAIL.
- Derivation Integrity: The model must explicitly link the location/fact back to the specific phrases found in the GT (e.g., "Konigstrasse in Hamburg"). It fails only if it reaches the answer using logic that ignores or replaces the GT Evidence entirely.

Instructions
- Extract GT Logic: Summarize the core reason the GT Evidence supports the answer.
- Extract Model Logic: Summarize the core reason the Model provides in its reasoning.
- Compare:
- If the Model's logic is a direct reflection of the GT Evidence: PASS.
- If the Model's logic is different, uses outside facts, or misses the "Why" provided in the GT: FAIL.

Response Format
GT Logic Summary: <What is the specific 'Why' in the GT evidence?> Model Logic Summary: <What is the specific 'Why' in the model's reasoning?> Comparison: <Do they align? Does the model rely on the same sentences/logic?>

Verdict: [PASS or FAIL]

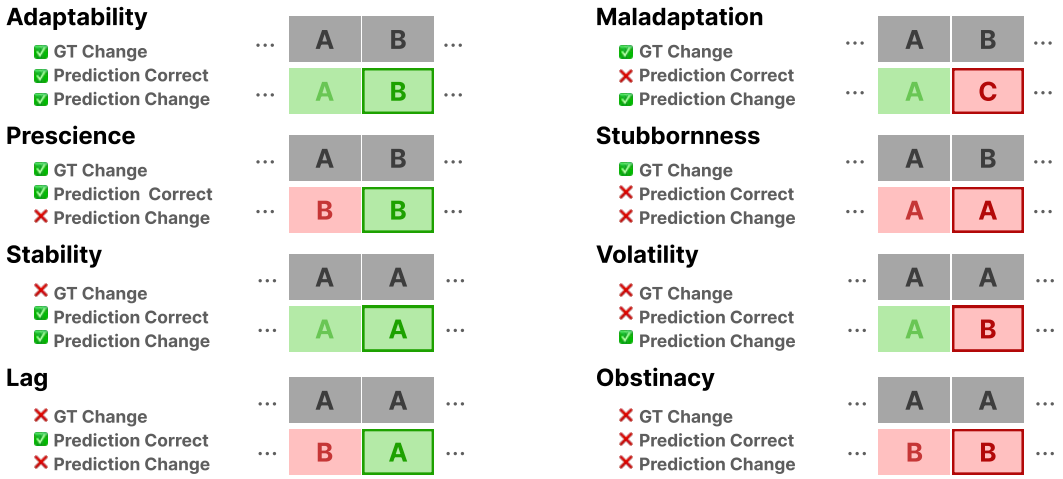

Figure 8: Extended visual explanations for model's behavior on OAKS described in Table 6.