# PROACTIVE MEMORY FOR AD-HOC RECALL OVER STREAMING DIALOGUES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Real-world dialogue usually unfolds as an infinite stream. It thus requires bounded-state memory mechanisms to operate within an infinite horizon. However, existing read-then-think memory is fundamentally misaligned with this setting, as it cannot support ad-hoc memory recall while streams unfold. To explore this challenge, we introduce **STEM-Bench**, the first benchmark for **ST**reaming **E**valuation of **M**emory. It comprises over 14K QA pairs in dialogue streams that assess perception fidelity, temporal reasoning, and global awareness under infinite-horizon constraints. The preliminary analysis on STEM-Bench indicates a critical *fidelity-efficiency dilemma*: retrieval-based methods use fragment context, while full-context models incur unbounded latency. To resolve this, we propose **ProStream**, a proactive memory framework for streaming dialogues built on a hierarchical structure. It enables ad-hoc memory recall on demand by reasoning over continuous streams with multi-granular distillation. Moreover, it employs Adaptive Spatiotemporal Optimization to dynamically optimize retention based on expected utility. It enables a bounded knowledge state for lower inference latency without sacrificing reasoning fidelity. Experiments show ProStream delivers higher reasoning fidelity than prior baselines while maintaining substantially lower latency than full-context alternatives.

## 1 INTRODUCTION

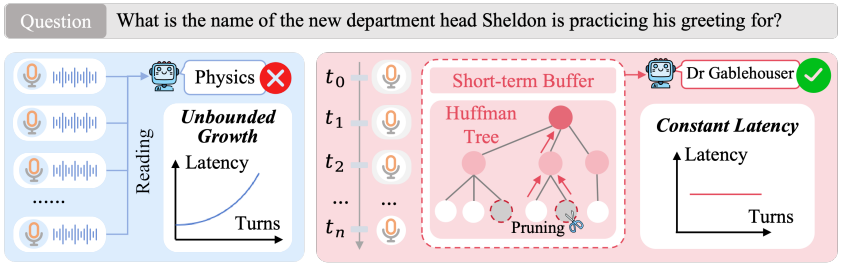
Large language models (LLMs) have largely advanced dialogue systems in various applications, e.g., customer service (Li et al., 2025) and personalized education (Kasneci et al., 2023). They usually adopt a long-term memory mechanism that enables systems to maintain continuity and reason across longitudinal sessions (Zhang et al., 2024; Packer et al., 2023). It has drawn significant attention, with many studies on memory retention and contextual reasoning (Maharana et al., 2024; Wu et al., 2024a; Pakhomov et al., 2025).

Despite this progress, most memory mechanisms continue to operate under a *read-then-think* paradigm, which presupposes a static, fully accessible context. This paradigm is fundamentally misaligned with the practical system for *streaming* dialogues, where *ad-hoc memory recall* may occur anytime as the dialogue unfolds. As shown in Figure 1 (Left), full-context attention incurs prohibitive computational costs and unbounded latency growth; moreover, noise from unorganized contexts can lead to reasoning failures. Consequently, we explore the *streaming memory paradigm* illustrated in Figure 1 (Right). It requires memory to operate as a bounded state within an infinite horizon, reflecting the reality that dialogue unfolds as an infinite stream.

To the best of our knowledge, our study is the first to *explore ad-hoc memory recall in streaming dialogues, which defines the ability to retrieve specific historical context on-demand to satisfy immediate reasoning needs.* To benchmark this pilot study, we introduce **STEM-Bench**, the first benchmark for **ST**reaming **E**valuation of **M**emory. Based on LongDialQA (Kim et al., 2024), we instantiate a continuously arriving dialogue stream using synthesized audio as the input interface. Our focus is on memory organization and ad-hoc recall under streaming constraints rather than on speech-recognition performance. STEM-Bench comprises 14k QA pairs to probe three core memory capabilities for ad-hoc recall: high-fidelity perception, structural logical reasoning, and dynamic global awareness. Our preliminary analysis of STEM-Bench uncovers a critical *fidelity-efficiency dilemma* challenging current methods. Standard memory-based on retrieval approaches fragment context, leading to reasoning degradation. Conversely, models attempting to maintain full context



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.




(a) Read-then-Think Paradigm (b) Streaming Memory Paradigm

Figure 1: Comparison of the *read-then-think* paradigm (left) and the *streaming memory paradigm* (right) based on The Big Bang Theory (TBBT) dialogues.

suffer from unbounded latency. These findings underscore that high performance in infinite streams cannot be achieved by *reactive* context scanning; it necessitates a paradigm shift toward *proactive* memory organization.

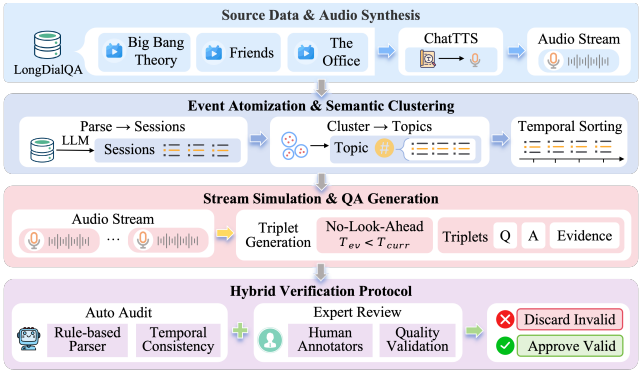
To resolve this, we propose **ProStream**, a proactive memory framework for streaming dialogues with a hierarchical organization. As depicted in Figure 1 (Right), ProStream transforms memory into an active, bounded state machine. By synergizing proactive semantic buffering with hierarchical, multi-granular distillation, it facilitates ad-hoc recall whenever demanded by reasoning. Moreover, it employs Adaptive Spatiotemporal Optimization to dynamically optimize information retention based on expected utility. Consequently, it maintains a bounded knowledge state for lower inference latency without sacrificing the reasoning fidelity. To evaluate ProStream, we first conduct a comprehensive comparison on STEM-Bench against non-trivial baseline memory mechanisms. The results demonstrate that ProStream achieves state-of-the-art performance in both accuracy and efficiency, with ablation studies confirming that all modules contribute positively. Furthermore, our quantitative analysis indicates that ProStream scales consistently well with LLMs of varying sizes, though it may render higher complexity in certain contexts. Finally, we provide a qualitative analysis to offer insights into the model's underlying strengths and remaining limitations.

In summary, our main contributions are as follows: 1) We present the pilot study of ad-hoc memory recall in dialogue streams with **STEM-Bench**, the first benchmark to evaluate perception, reasoning, and global awareness under strict infinite-horizon constraints. 2) We propose **ProStream**[1], a novel framework implementing proactive hierarchical memory for streaming dialogues. Leveraging adaptive spatiotemporal optimization to prioritize information with expected utility, it constructs a bounded knowledge topology for ad-hoc, reasoning-demanding recall in continuous streams. 3) Extensive evaluations show ProStream outperforms state-of-the-art baselines in accuracy and efficiency by large margins. Crucially, it avoids repeatedly processing the full dialogue history of existing methods with much lower inference latency, enabling scalable real-time deployment.

## 2 Related Work

**Dialogue Benchmarks for Long-Term Memory.** STEM-Bench falls within this category. This line of research reflects a progressive shift from traditional single-turn modeling (Budzianowski et al., 2018; Wei et al., 2018) toward long-term dialogue understanding. Early efforts evaluated personalized response generation from extended chat histories (Xu et al., 2022a;b). To quantify memory accuracy, subsequent benchmarks adopted the Question Answering (QA) paradigm. Representative studies include MemoryBank (Zhong et al., 2024) (multi-day histories with 194 probing questions), LoCoMo (Maharana et al., 2024) (single-hop to adversarial reasoning over 50 long-term dialogues), and PerLTQA (Du et al., 2024) (over 8k questions spanning world knowledge, social relationships, and events). Recently, LongDialQA (Kim et al., 2024) introduced response latency constraints to penalize slow systems, while LongMemEval (Wu et al., 2024a) and ConvoMem (Pakhomov et al., 2025) evaluated reasoning, knowledge updates, and abstention capabilities across tens of thousands of QA pairs. Despite this, existing benchmarks adopt a static, *read-and-think* paradigm diverging from real-world dialogue streams with dynamic, ever-growing contexts. In contrast, STEM-Bench explores *streaming* memory to facilitate ad-hoc recall whenever reasoning demands it within dialogue streams.

[1]Code is available at https://anonymous.4open.science/r/ProStream-8827




(a) Data Construction Pipeline

| Category | Big Bang | Friends | The Office | Total |
|---|---|---|---|---|
| High-Fidelity Perception (HFP) | | | | |
| Single-hop | 1,480 | 1,583 | 3,229 | 6,292 |
| Adversarial | 549 | 547 | 767 | 1,863 |
| Open-domain | 769 | 808 | 1,170 | 2,747 |
| Structural Logical Reasoning (SLR) | | | | |
| Multi-hop | 475 | 476 | 367 | 1,318 |
| Comparative | 408 | 408 | 298 | 1,114 |
| Dynamic Global Awareness (DGA) | | | | |
| Aggregative | 282 | 330 | 219 | 831 |
| Temporal | 284 | 301 | 188 | 773 |
| Total QAs | 4,247 | 4,453 | 6,238 | 14,938 |
| Meta-Data | | | | |
| Total Topics | 1,370 | 1,400 | 3,283 | 6,053 |
| Total Utterances | 18,617 | 18,164 | 18,892 | 55,673 |
| Avg. Utts/Topic | 13.59 | 12.97 | 5.75 | 9.20 |

(b) Data Statistics

Figure 2: Overview of STEM-Bench benchmark curation. (a) Data construction pipeline. (b) Dataset statistics and example instances for HFP, SLR, and DGA. TBBT: The Big Bang Theory.

**Methods of Long-Term Memory.** ProStream aligns with recent efforts on equipping agents with long-term memory. Early approaches expand the context window to process extensive histories directly (Fu et al., 2024), but suffer from prohibitive computation and the "lost-in-the-middle" issue (Liu et al., 2024b). Other methods introduce differentiable memory modules (Wang et al., 2023) or treat memory as context compression, representing histories as internal states (Chevalier et al., 2023), discrete tokens (Xu et al., 2023), or retrievable segments via Retrieval-Augmented Generation (RAG) (Jimenez Gutierrez et al., 2024). Recent work explores more structured memory mechanisms. LiCoMemory (Huang et al., 2025) and SGMem (Wu et al., 2025) model hierarchical or graph-based associations, while RMM (Tan et al., 2025) and MemGAS (Xu et al., 2025a) employ adaptive retrieval and online reinforcement learning for memory refinement. Event-centric approaches (Zhou, 2025) further ground history in structured semantic representations. However, these methods remain largely retrieval-centric and batch-oriented, making them less suitable for streaming scenarios requiring ad-hoc memory recall. In contrast, ProStream formulates memory as online hierarchical state maintenance with proactive spatiotemporal compression, enabling efficient long-term retention under low-latency constraints.

## 3 Our STEM-Bench

**Problem Formulation.** STEM-Bench formulates dialogue evaluation as an incremental state-evolution problem over an unbounded input stream. Let $\mathcal{S} = \{I_1, I_2, \dots\}$ denote the input sequence, where each $I_t$ is either an interaction unit $U_t = \{s_t, l_t\}$ consisting of speaker identity $s_t$ and audio content $l_t$, or a question probe $P_t = \{q_t, y_t\}$, where $q_t$ is the query issued at time $t$ and $y_t$ is the corresponding ground-truth answer. Under streaming constraints, the model maintains a bounded memory state $\mathcal{M}_t$, which is updated online from $\mathcal{M}_{t-1}$ and $U_t$. Upon receiving a probe, the model generates an answer $a_t$ based on $\mathcal{M}_t$ and $q_t$, with the goal of maximizing alignment between $a_t$ and $y_t$ without access to future context.

**STEM-Bench Construction.** Grounded in cognitive and computational theories, STEM-Bench evaluates three core capabilities: **(1) High-Fidelity Perception (HFP)** measures atomic detail retention under noise and weakly constrained queries through **Single-hop**, **Open-domain**, and **Adversarial** tasks, targeting the *Lost-in-the-Middle* effect (Liu et al., 2024a) and compression-induced hallucinations. **(2) Structural Logical Reasoning (SLR)** assesses the ability to connect fragmented events across timestamps via **Multi-hop** and **Comparative** tasks, emphasizing hierarchical evidence composition over shallow retrieval. **(3) Dynamic Global Awareness (DGA)** evaluates the online maintenance of evolving statistical and temporal states through **Aggregative** and **Temporal** tasks, requiring continual state updates without backtracking.

**Data Construction Pipeline.** We construct STEM-Bench by transforming textual dialogue data into a streaming memory evaluation setting, as illustrated in Figure 2(a). Synthesized audio is used only to instantiate the continuously arriving input interface, after which the dialogue is transcribed and processed by the memory system. The pipeline begins with audio synthesis, converting LongDialQA

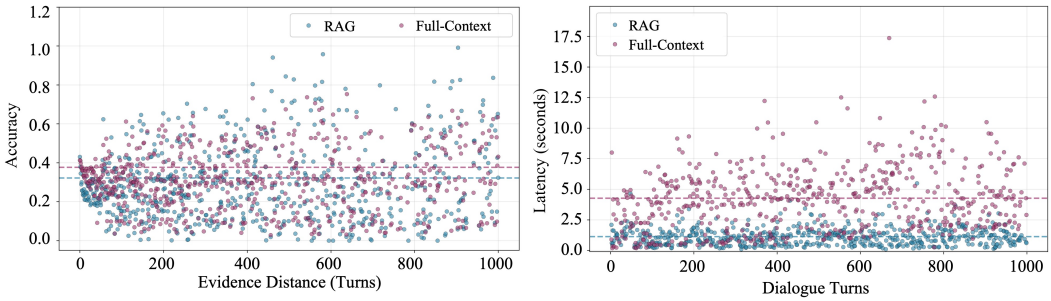



Figure 3: Preliminary analysis of RAG and full-context performance. (Right) The average score of all performance metrics $Q$ across evidence distances. (Left) End-to-end response latency of dialogue turns. The dashed lines indicate the overall mean values.

(Kim et al., 2024) into audio using ChatTTS[2]. We then reorganize the resulting dialogue content into a causally valid streaming sequence across sessions. Next, the QA generation stage produces QA pairs with evidence annotations under a strict *No-Look-Ahead Constraint*, ensuring that for any trigger turn $T_{\text{curr}}$, all supporting evidence $T_{\text{ev}}$ satisfies $T_{\text{ev}} < T_{\text{curr}}$. Finally, we apply a hybrid verification protocol to ensure data quality. After automated rule-based auditing, three domain experts manually validated logical consistency on a random subsample. This review yielded substantial agreement among annotators.

**Dataset Statistics.** Figure 2(b) summarizes the statistics of STEM-Bench, which comprises 14,938 QA pairs grounded in 55,673 utterances and 6,053 semantic topics from three source corpora. The benchmark is organized around three complementary memory capabilities. High-Fidelity Perception forms the largest portion, including over 6,200 **Single-hop** queries, 2,747 **Open-domain** queries, and 1,863 **Adversarial** instances. Structural Logical Reasoning further includes 2,432 **Multi-hop** and **Comparative** questions that require integrating evidence across temporally separated events. Dynamic Global Awareness is evaluated through over 1,600 **Aggregative** and **Temporal** questions, which require continual state updates for frequency tracking and temporal ordering.

**Evaluation Metrics.** We adopt a holistic evaluation protocol covering response quality and efficiency. BLEU-4 (**B-4**) and ROUGE-L (**R-L**) measure lexical overlap and sequence-level similarity, while the standard BERTScore F1 (**B-S**) measures contextual semantic similarity between the generated answer and the reference answer. Key Entity Matching (**KEM**) measures whether all required entities extracted from the ground-truth evidence are correctly mentioned in the generated answer. Evidence Similarity (**Evid.**) measures the cosine similarity between the retrieved context and the ground-truth evidence. We further use Gemini-2.5-Pro (**Gem**) as a reference-guided LLM judge to evaluate answer correctness. For aggregate comparisons, we use an average quality score $Q$ computed as the unweighted mean of the six higher-is-better metrics: B-4, R-L, B-S, Evid., KEM, and Gem. Latency is excluded from this score and is reported separately. The individual metrics are retained in the main tables to provide a more detailed view of model performance. For efficiency, we report the average end-to-end response latency (**Time**, in seconds), measured from probe arrival to answer completion and including all synchronous operations on the response-critical path.

## 4 Preliminary Analysis

To better motivate the design of a streaming memory architecture, we compare two representative paradigms on STEM-Bench: a **Full-Context baseline**, which accesses the complete dialogue history, and a **Standard RAG** baseline based on vector-similarity retrieval. As shown in Figure 3 (Right), the Full-Context baseline consistently achieves higher reasoning fidelity, with its average score across evaluation metrics (dashed purple line) remaining above that of RAG (dashed blue line). The concentration of low-accuracy cases for RAG across different evidence distances suggests that retrieval-based methods often rely on fragmented evidence and therefore struggle to preserve the global dependencies required for complete reasoning chains. In contrast, Figure 3 (Left) reveals a clear *fidelity–efficiency trade-off*. While the Full-Context baseline provides stronger accuracy, its latency

[2]https://github.com/2noise/ChatTTS

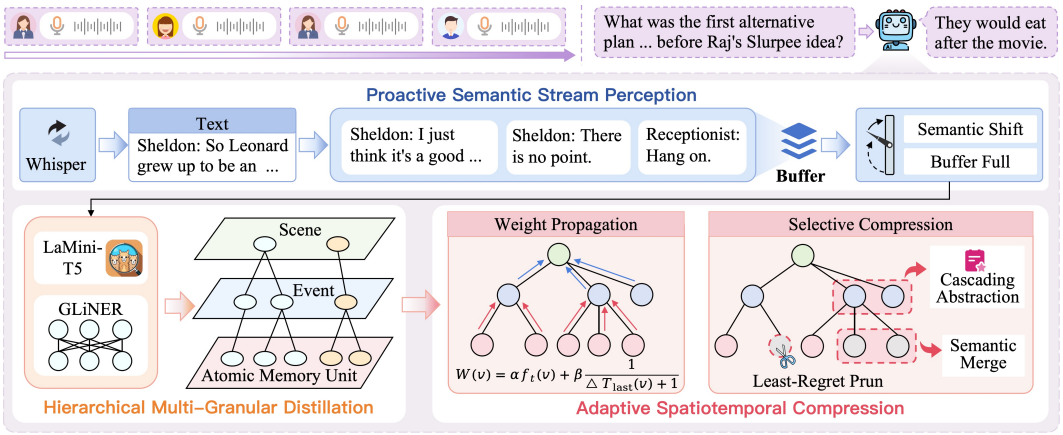



Figure 4: Overview of our ProStream with four components discussed in turn from §5.1 to §5.4.

increases substantially as the dialogue history grows, limiting its practicality in latency-sensitive settings. RAG, by comparison, maintains relatively low and stable latency, but does so at the cost of weaker global context integration and reduced reasoning fidelity. Together, these observations suggest that an effective streaming memory mechanism should preserve as much reasoning fidelity as possible under a bounded computational budget. Motivated by this objective, we introduce **ProStream**, a framework designed to address this trade-off through proactive state maintenance as below.

## 5 The ProStream Framework

As illustrated in Figure 4, **ProStream** reformulates memory maintenance as a bounded state evolution process, transforming an unbounded input stream into a compact, structured memory state. This pipeline operates in four stages: *Proactive Semantic Stream Perception* segments incoming dialogue into discrete semantic blocks; *Hierarchical Multi-Granular Distillation* structures these blocks into a multi-layered tree topology; *Adaptive Spatiotemporal Optimization* dynamically prioritizes information under strict capacity constraints; and *Probabilistic Evidence-Grounded Reasoning* synthesizes the maintained memory states with immediate context for response generation.

### 5.1 Proactive Semantic Stream Perception

Updating global memory $\mathcal{M}$ at every single turn incurs unnecessary computational overhead and risks fragmenting coherent contexts. To address this, we introduce the Short-Term Sensing Buffer (STSB) $\mathcal{B}$. Functionally, $\mathcal{B}$ accumulates an incoming stream of interaction units $U_t = \{s_t, l_t\}$. To capture both linguistic content and speaker information, we first transcribe the audio component $l_t$ via Whisper ASR $f_{\text{ASR}}$ (Radford et al., 2023) and concatenate it with the speaker identity $s_t$ to form a composite textual unit $x_t = [s_t; f_{\text{ASR}}(l_t)]$. We formulate stream segmentation as an online boundary detection problem. Specifically, for each incoming utterance $x_t$, we compute its dense representation $\mathbf{v}_t = f_{\text{Enc}}(x_t)$ with an encoder $f_{\text{Enc}}(\cdot)$ and monitor semantic continuity via cosine similarity $\psi_t = \cos(\mathbf{v}_t, \mathbf{v}_{\text{prev}})$ with the preceding unit's embedding. Once this local coherence drops below a drift threshold $\tau_{\text{drift}}$ or $\mathcal{B}$ reaches capacity, the accumulated sequence is consolidated into a semantic block $\mathcal{T}_{\text{block}}$, ready for the following hierarchical distillation (§5.2). To preserve boundary-spanning dependencies, we retain a context window of size $W_{\text{STSB}}$ from the buffered content for the subsequent cycle. The buffer capacity controls when a block is flushed, whereas $W_{\text{STSB}}$ controls the amount of context carried across the detected boundary.

### 5.2 Hierarchical Multi-Granular Distillation

Motivated by the analysis results from Figure 3, the unstructured semantic block $\mathcal{T}_{\text{block}}$ is distilled into a hierarchical tree structure $\mathcal{H}$. This topology relies on three semantic layers: 1) **Scene** $c$ for coarse-grained thematic clustering; 2) **Event** $e$ for temporal context segmentation; and 3) **Atomic Memory Unit** $o$ for fine-grained entity-level retention.

Specifically, the distillation process transforms the input semantic block $\mathcal{T}_{\text{block}}$ into this structure through a dual-path mechanism. On the generative front, we employ a unified instruction-tuned model $f_\theta(\cdot,\cdot)$ (Wu et al., 2024b) to perform recursive summarization. Conditioned on a summarization instruction $\rho_{\text{evt}}$, the model first distills the block into a textual event summary $e = f_\theta(\mathcal{T}_{\text{block}}, \rho_{\text{evt}})$. This summary is subsequently abstracted into a higher-level scene category $c = f_\theta(e, \rho_{\text{scn}})$ using a classification prompt $\rho_{\text{scn}}$. In parallel, to populate the atomic layer, we utilize GLiNER (Zaratiana et al., 2024) to identify candidate subject and object entities. An LLM-based relation induction step then infers the relation $\mathcal{R}$ between them, yielding a relational tuple $\epsilon = (\mathcal{E}_{\text{sub}}, \mathcal{R}, \mathcal{E}_{\text{obj}})$. We treat the extracted subject and object entities as candidate Atomic Memory Unit (AMU) nodes $o$. Each candidate $o$ is encoded via $f_{\text{Enc}}(\cdot)$ into $\mathbf{v}_o$. The candidate is attached as a child node under the corresponding Event $e_j$ only if it satisfies a novelty check against the existing atomic node set $\mathcal{O}_{\text{amu}}$: $\max_{o' \in \mathcal{O}_{\text{amu}}} \cos(\mathbf{v}_o, \mathbf{v}_{o'}) < \theta_{\text{concept}}$. Finally, the relation $\mathcal{R}$ is stored as a contextual attribute of the corresponding entity nodes, preserving the semantic dependencies of the original tuple for evidence reconstruction. Distilled but not-yet-mounted representations are temporarily maintained in the pending buffer $\mathcal{M}_{\text{pend}}$. These representations are not included in the long-term retention budget before mounting; they are integrated into the hierarchical tree at the subsequent memory-update step and become eligible for budgeted retention thereafter.

### 5.3 Adaptive Spatiotemporal Optimization

Streaming dialogues present a challenge of retaining useful information under a bounded memory budget. As utterances arrive, the memory topology $\mathcal{H}$ must self-evolve to balance retention quality and computational efficiency. We therefore formulate memory maintenance as a *Recall Probability Maximization* problem, aiming to reduce the expected loss associated with discarding retrieval-relevant information.

**Probabilistic Utility Formulation.** Drawing on the Rational Analysis of Memory (Anderson, 2013), we use access frequency and temporal recency as two signals for estimating the future utility of a memory node. For a node $v$ in the memory tree, we define the utility score:

$$u_{v,t} = \underbrace{\alpha \cdot \log(f_{v,t} + 1)}_{\text{Frequency Prior}} + \underbrace{\beta \cdot \exp\left(-\Delta t_v / \tau\right)}_{\text{Temporal Drift}}, \tag{1}$$

where $f_{v,t}$ denotes the cumulative access frequency, $\Delta t_v$ is the recency of the node, $\tau$ is the characteristic decay scale, and $\alpha, \beta$ are hyperparameters balancing the two terms.

**Optimization Objective.** Let $s_t \in \{0,1\}^{|\mathcal{H}_t|}$ be the state vector indicating retention at time $t$. Since the value of information may change as the stream evolves, we define the objective as maximizing cumulative discounted utility subject to a dynamic capacity constraint:

$$\max_{\{\mathbf{s}_t\}_{t=1}^{\infty}} \sum_{t=1}^{\infty} \gamma^t \sum_{v \in \mathcal{H}_t} s_{v,t} \cdot u_{v,t}, \text{s.t.} \quad \sum_{v \in \mathcal{H}_t} s_{v,t} \cdot c_v \le \mathcal{T}_{\max}, \quad \forall t, \tag{2}$$

where $\gamma \in (0,1)$ is a discount factor and $c_v$ is the token cost. This formulation is related to online knapsack-style retention under time-varying utilities.

**The ProStream Policy.** We approximate the objective via a *Greedy Marginal-Utility Policy* $\pi$. When the budget constraint is violated, $\pi$ restores feasibility through a three-stage process. First, it performs *Least-Regret Pruning* by iteratively discarding the node $v^* = \arg\min_{v \in \mathcal{H}_t}(u_{v,t}/c_v)$ with the lowest utility density. Second, to reduce redundant representations, the policy applies *Semantic Merging*, collapsing node pairs $\{v_i, v_j\}$ into a single centroid $\mathbf{v}_{\text{new}}$ whenever their latent distance $d(\mathbf{v}_i, \mathbf{v}_j) < \theta_{\text{sim}}$. Finally, to maintain topological consistency, *Cascading Abstraction* recursively prunes a parent node if it loses all supporting leaf nodes. These operations provide a practical bounded-state retention policy without requiring the system to reconstruct the entire memory topology at every update.

### 5.4 Probabilistic Evidence-Grounded Generation

To generate accurate responses, ProStream synthesizes a unified context $\mathcal{K} = \{\mathcal{B}, \mathcal{M}_{\text{pend}}, \mathcal{M}_{\text{tree}}\}$, where $\mathcal{B}$ is the STSB, $\mathcal{M}_{\text{pend}}$ is an intermediate buffer for distilled but not-yet-mounted representations, and $\mathcal{M}_{\text{tree}}$ comprises the top-$k$ semantic paths retrieved from the hierarchical tree $\mathcal{H}$. To

Table 1: Main comparison results on our STEM-Bench. Best results are **bolded**, second best are underlined. Metrics marked with ↑ indicate higher is better, while ↓ denotes lower is better. For ablations (bottom), w/o indicates without.

| Method | The Big Bang Theory | | | | | | | Friends | | | | | | | The Office | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Time↓ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | Gem↑ | Time↓ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | Gem↑ | Time↓ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | Gem↑ |
| RAG | 0.957 | 0.068 | 0.247 | 0.781 | 0.355 | 0.508 | 0.331 | 0.923 | 0.061 | 0.231 | 0.768 | 0.345 | 0.502 | 0.309 | 0.918 | 0.074 | 0.271 | 0.778 | 0.322 | 0.524 | 0.385 |
| RQ-RAG | 5.149 | 0.051 | 0.201 | 0.770 | 0.275 | 0.460 | 0.329 | 5.098 | 0.043 | 0.184 | 0.758 | 0.372 | 0.439 | 0.313 | 1.058 | 0.072 | 0.270 | 0.780 | 0.466 | 0.535 | 0.376 |
| RAPTOR | 0.519 | 0.041 | 0.165 | 0.761 | 0.302 | 0.375 | 0.214 | 0.513 | 0.035 | 0.157 | 0.746 | 0.396 | 0.367 | 0.214 | 0.443 | 0.032 | 0.142 | 0.745 | 0.324 | 0.334 | 0.201 |
| Full Context | 7.333 | 0.065 | 0.249 | 0.737 | 0.250 | 0.710 | **0.521** | 4.870 | 0.057 | 0.241 | 0.732 | 0.360 | 0.706 | **0.505** | 4.776 | 0.066 | 0.266 | 0.742 | 0.292 | **0.731** | **0.574** |
| Summary | 2.001 | 0.049 | 0.200 | 0.783 | 0.296 | 0.407 | 0.278 | 2.352 | 0.042 | 0.180 | 0.762 | 0.334 | 0.385 | 0.246 | 1.641 | 0.048 | 0.201 | 0.770 | 0.365 | 0.397 | 0.272 |
| GraphRAG | 0.997 | 0.069 | 0.246 | 0.788 | 0.417 | 0.479 | 0.341 | 0.950 | 0.055 | 0.227 | 0.775 | 0.460 | 0.465 | 0.309 | 1.231 | 0.055 | 0.223 | 0.779 | 0.420 | 0.433 | 0.311 |
| HippoRAG2 | 4.231 | 0.046 | 0.176 | 0.742 | 0.481 | 0.480 | 0.263 | 6.539 | 0.041 | 0.165 | 0.742 | 0.529 | 0.479 | 0.249 | 4.404 | 0.040 | 0.169 | 0.740 | 0.499 | 0.458 | 0.270 |
| MemoRAG | 7.498 | 0.049 | 0.193 | 0.791 | **0.551** | 0.366 | 0.265 | 4.146 | 0.043 | 0.178 | 0.771 | **0.575** | 0.355 | 0.242 | 5.922 | 0.036 | 0.171 | 0.773 | **0.568** | 0.345 | 0.225 |
| A-Mem | 1.301 | 0.057 | 0.204 | 0.755 | 0.370 | 0.371 | 0.272 | 1.290 | 0.053 | 0.200 | 0.742 | 0.386 | 0.376 | 0.266 | 1.272 | 0.059 | 0.220 | 0.753 | 0.353 | 0.377 | 0.295 |
| MemGAS | 5.711 | 0.013 | 0.070 | 0.528 | 0.402 | **0.768** | 0.508 | 7.982 | 0.012 | 0.072 | 0.530 | 0.433 | **0.749** | 0.466 | 7.694 | 0.006 | 0.039 | 0.592 | 0.547 | 0.417 | 0.285 |
| **ProStream** | **0.408** | **0.118** | **0.390** | **0.817** | 0.485 | 0.612 | 0.509 | **0.373** | **0.099** | **0.360** | **0.801** | 0.491 | 0.588 | 0.476 | **0.316** | **0.114** | **0.408** | **0.819** | 0.491 | 0.626 | 0.534 |
| -w/o $\mathcal{B}$ | 0.409 | 0.049 | 0.182 | 0.783 | 0.071 | 0.369 | 0.255 | 0.469 | 0.034 | 0.149 | 0.750 | 0.276 | 0.364 | 0.202 | 0.452 | 0.030 | 0.138 | 0.758 | 0.191 | 0.336 | 0.193 |
| -w/o $\mathcal{M}_{\text{tree}}$ | 0.400 | 0.116 | 0.382 | 0.810 | 0.463 | 0.607 | 0.500 | 0.318 | 0.097 | 0.304 | 0.703 | 0.480 | 0.581 | 0.466 | 0.319 | 0.110 | 0.402 | 0.817 | 0.484 | 0.620 | 0.526 |
| -w/o $\mathcal{M}_{\text{pend}}$ | 0.401 | 0.115 | 0.380 | 0.810 | 0.475 | 0.602 | 0.489 | 0.320 | 0.090 | 0.356 | 0.704 | 0.482 | 0.580 | 0.473 | 0.304 | 0.111 | 0.401 | 0.817 | 0.487 | 0.617 | 0.528 |

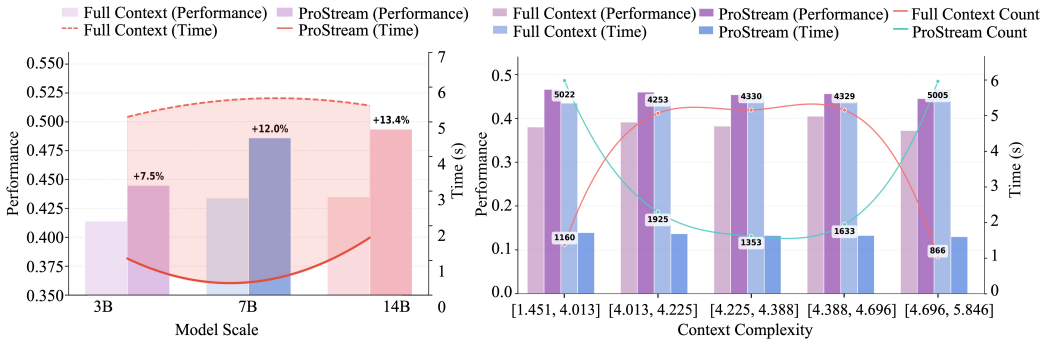
construct $\mathcal{M}_{\text{tree}}$, we employ a top-down traversal strategy. We first encode the ad-hoc query $q$ into a dense vector $\mathbf{v}_q = f_{\text{Enc}}(q)$. Using this representation, the model recursively prunes irrelevant branches starting from the *Scene* level, drilling down to *Event* summaries and finally to *AMUs*. At each level, branch relevance is determined by the most relevant descendant node. For a candidate AMU $o$, we compute a composite relevance score: $S(o) = \cos(\mathbf{v}_q, \mathbf{v}_o) \cdot u_{o,t}$, where $u_{o,t}$ is the time-variant utility weight defined in Eq. 1. This mechanism filters low-utility noise while preserving the hierarchical paths needed for memory recall. Given the functional context $\mathcal{K}$, the model generates the final answer sequence $a$ according to the standard autoregressive objective: $\mathcal{J}(\theta) = \sum_{j=1}^{|a|} \log P_\theta(a_j \mid a_{<j}, q, \mathcal{K})$, where $a_j$ denotes the $j$-th output token of answer $a$. This formulation conditions answer generation on recent context, pending distilled representations, and retrieved long-term evidence, balancing fidelity and coherence.

# 6 Experiment

## 6.1 Experimental Setting

**Implementation Details.** For *Proactive Semantic Stream Perception*, we set the drift threshold to $0.7$, the buffer flush capacity to $5$, and $W_{\text{STSB}}$ to $10$. Here, the buffer capacity determines when the current semantic block is flushed, whereas $W_{\text{STSB}}$ determines the amount of context retained across the detected boundary. For *Hierarchical Multi-Granular Distillation*, we use a concept similarity threshold of $0.85$ for AMU novelty filtering. For *Adaptive Spatiotemporal Optimization*, we set the utility weights to $\alpha = 0.6$ for frequency and $\beta = 0.4$ for recency. For *Probabilistic Evidence-Grounded Generation*, retrieval selects the top-$5$ scene nodes and the top-$10$ event nodes within each selected scene. The final context contains the top-3 AMUs ranked by the composite relevance score, subject to a minimum cosine-similarity threshold of $0.5$. Detailed model configurations and prompt templates are provided in Appendix B.

**Baseline Methods.** We compare ProStream with representative retrieval, summarization, and structured-memory baselines from two groups. The first group consists of **retrieval and summarization baselines**, including Standard RAG, Full-Context, Rolling Summary, RQ-RAG (Chan et al., 2024), RAPTOR (Sarthi et al., 2024), GraphRAG (Edge et al., 2024), and HippoRAG2 (Gutiérrez et al., 2025). The second group consists of **agentic memory systems**, including MemoRAG (Qian et al., 2025), A-Mem (Xu et al., 2025b), and MemGAS (Xu et al., 2025a). Unless otherwise specified, all methods use the same answer-generation protocol and are evaluated with the same dataset, query set, and latency measurement procedure. All compared methods receive the same ASR-derived textual input, so the evaluation primarily measures memory organization and reasoning rather than differences in speech recognition.




(a) Different LLM backbones

(b) Different context complexity (entropy)

Figure 5: Average performance $Q$ (bars) and end-to-end response latency (lines) across different experimental settings. (a) Results for different LLM backbones; percentages above the bars indicate the relative improvement of ProStream over the Full-Context baseline. (b) Results across context-complexity levels, and the curves show the corresponding score distributions.

### 6.2 Main Results

We first compare the methods in terms of response quality and efficiency. Table 1 shows that ProStream achieves a strong quality-latency trade-off. Compared with the Full-Context baseline, it attains higher scores on several response-quality and evidence-alignment metrics while using substantially lower latency, although the Full-Context baseline remains competitive on some entity-level metrics. These results suggest that proactive distillation can preserve task-relevant information without exposing the generator to the complete dialogue history. ProStream does not dominate every individual metric: it may trail retrieval-centric methods such as MemoRAG on evidence-alignment measures. This pattern is consistent with the intended trade-off between structured semantic abstraction and verbatim evidence retrieval. A stage-wise cost breakdown of online memory maintenance is provided in Appendix E, and transfer evaluations on existing conversation-memory benchmarks are reported in Appendix F.

### 6.3 Ablation Study.

We next examine the contribution of individual components using the ablations in Table 1. Removing the Short-Term Sensing Buffer (w/o $\mathcal{B}$) causes a substantial drop in response and evidence metrics, highlighting the role of recent local context in reducing contextual fragmentation. Removing the hierarchical memory component (w/o $\mathcal{M}_{\text{tree}}$) mainly affects global consistency and entity tracking, supporting the importance of structured long-term retention for cross-temporal reasoning. Removing the pending buffer (w/o $\mathcal{M}_{\text{pend}}$) also degrades performance, suggesting that retaining newly distilled representations before hierarchical insertion helps maintain continuity across memory updates. We use "substantial" rather than "catastrophic" here because the ablation results quantify performance degradation but do not by themselves establish a categorical failure mode.

### 6.4 Quantitative Analysis

We further evaluate ProStream under varying backbone scales and context-complexity levels. Unless otherwise stated, the quality score is computed as the unweighted mean of the selected higher-is-better performance metrics, excluding latency. Additional sensitivity analyses are provided in Appendix D.

**Varying Backbone Scales.** We assess the scalability of ProStream by evaluating it with Qwen-3B, Qwen-7B, and Qwen-14B backbones and comparing it with the Full-Context baseline, as shown in Figure 5(a). Larger backbones generally yield larger relative gains from ProStream, suggesting that additional reasoning capacity helps the generator interpret the structured memory representation. The latency trend is non-monotonic: ProStream-3B is slower than ProStream-7B in our setting, while the Full-Context baseline becomes progressively more expensive as the dialogue history grows. The lower efficiency of ProStream-3B may reflect the fixed overhead of processing structured memory,

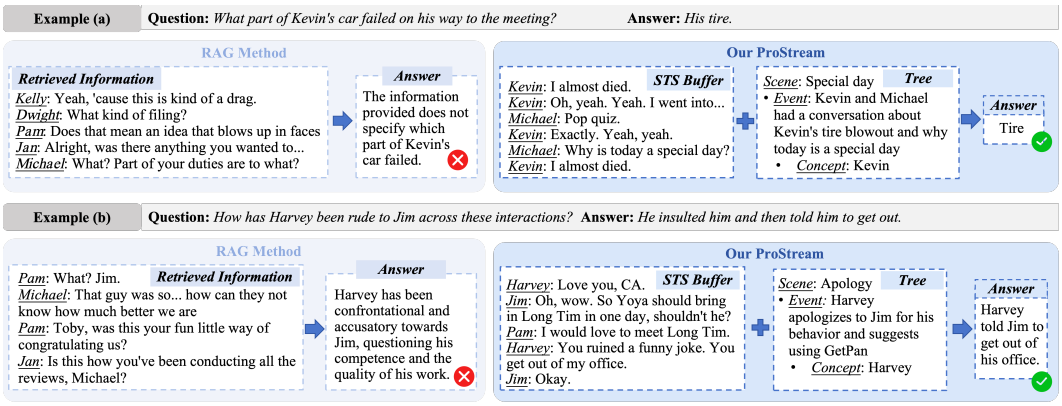



Figure 6: Qualitative comparison (Case Study) of the RAG method and our ProStream.

which can outweigh the computational savings of the smaller backbone. This explanation should be treated as a possible mechanism rather than a directly established causal finding.

**Varying Context Complexity.** We further examine the sensitivity of ProStream to context complexity, measured by GPT-2 cross-entropy loss (Salazar et al., 2020). We stratify examples by the resulting complexity score and report the results in Figure 5(b). ProStream maintains a favorable quality–latency trade-off across the evaluated complexity levels. The distribution curves suggest that the contexts produced by ProStream have a more dispersed complexity profile than those produced by the Full-Context baseline. Rather than interpreting this pattern as evidence that structured memory intrinsically complicates contexts, we view it as an indication that distillation changes the distribution of information presented to the generator. A quantitative comparison of the distributions, such as variance or inter-quantile range, should be reported if the claim about polarization is retained.

### 6.5 Qualitative Analysis

**Case Study.** Figure 6 illustrates how ProStream combines recent context with retrieved long-term memory. In Example (a), the short-term buffer contains the ambiguous statement *"I almost died,"* whereas the hierarchical memory retrieves the more specific fact that the incident involved a *"tire blowout."* In Example (b), the STSB captures only the recent *"eviction"* incident, while the retrieved *"insult"* and *"apology"* events provide earlier context needed to reconstruct the broader behavioral sequence. These examples illustrate the complementary roles of local and hierarchical memory: the former preserves immediate cues, while the latter supplies relevant distal evidence when the recent context is incomplete.

**Error Analysis.** Our error analysis identifies two recurring limitations. First, the model may exhibit an instruction-compliance bias when a prompt asserts a fact that conflicts with the available evidence; for example, it may accept a fallacious *"sole creator"* claim despite contradictory context (Appendix Figure 7(a)). Second, the model may fail to disentangle implicit activity context when salient semantic noise obscures a less prominent cue, leading to intent misattribution in cases where a high-density monologue overshadows a crossword-puzzle context (Appendix Figure 7(b)). These observations motivate future work on explicit conflict detection and activity-aware attention.

## 7 Conclusion

This paper studies ad-hoc memory recall in unbounded streaming dialogues and introduces STEM-Bench to evaluate fine-grained perception, cross-temporal reasoning, and dynamic global awareness under no-look-ahead constraints. We further propose **ProStream**, a proactive memory framework that combines semantic stream perception, hierarchical multi-granular distillation, adaptive spatiotemporal retention, and evidence-grounded generation to maintain a bounded yet informative memory state. Experiments show that ProStream achieves a favorable quality–latency trade-off, substantially reducing latency compared with full-context processing while achieving competitive or superior performance on several reasoning-oriented metrics. These results demonstrate the potential of proactive hierarchical memory for efficient long-term reasoning in streaming dialogue systems.

## AI USE STATEMENT

Generative AI tools were used for audio synthesis, LLM-based semantic summarization and relation induction during benchmark construction, LLM-as-a-Judge evaluation, code refactoring, initial manuscript drafting, and language polishing. They were not used to design or implement the core algorithm, derive mathematical proofs, or collect raw experimental data. All AI-assisted outputs were reviewed by the authors.

## REFERENCES

John R Anderson. *The adaptive character of thought*. Psychology Press, 2013.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 5016–5026, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1547. URL https://aclanthology.org/D18-1547/.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*, 2024.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 2318–2335, 2024.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846, 2023.

Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai Wong. Perltqa: A personal long-term memory dataset for memory classification, retrieval, and fusion in question answering. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pp. 152–164, 2024.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 14125–14134, 2024.

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

Zhengjun Huang, Zhoujin Tian, Qintian Guo, Fangyuan Zhang, Yingli Zhou, Di Jiang, and Xiaofang Zhou. Licomemory: Lightweight and cognitive agentic memory for efficient long-term reasoning. *arXiv preprint arXiv:2511.01448*, 2025.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37:59532–59569, 2024.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.

Jiho Kim, Woosog Chay, Hyeonji Hwang, Daeun Kyung, Hyunseung Chung, Eunbyeol Cho, Yohan Jo, and Edward Choi. Dialsim: A real-time simulator for evaluating long-term dialogue understanding of conversational agents. *arXiv e-prints*, pp. arXiv–2406, 2024.

Fei Li, Yanyan Wang, Yin Xu, Shiling Wang, Junli Liang, Zhengyi Chen, Wenrui Liu, Qiangzhong Feng, Ticheng Duan, Youzhi Huang, et al. Performance evaluations of large language models for customer service. *International Journal of Machine Learning and Cybernetics*, 16(5):2997–3017, 2025.

Xianming Li and Jing Li. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*, 2023.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024a. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

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

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 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)*, pp. 13851–13870, 2024.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. Memgpt: Towards llms as operating systems. 2023.

Egor Pakhomov, Erik Nijkamp, and Caiming Xiong. Convomem benchmark: Why your first 150 conversations don't need rag. *arXiv preprint arXiv:2511.10523*, 2025.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, pp. 2366–2377, 2025.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. Masked language model scoring. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 2699–2712, 2020.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439, 2025.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36:74530–74543, 2023.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. Airdialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3844–3854, 2018.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024a.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. Lamini-lm: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 944–964, 2024b.

Yaxiong Wu, Yongyue Zhang, Sheng Liang, and Yong Liu. Sgmem: Sentence graph memory for long-term conversational agents. *arXiv preprint arXiv:2509.21212*, 2025.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, Tong Xu, et al. Towards multi-granularity memory association and selection for long-term conversational agents. *arXiv preprint arXiv:2505.19549*, 2025a.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*, 2023.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 5180–5197, 2022a.

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

Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang Wang. Long time no see! open-domain conversation with long-term persona memory. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2639–2650, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.207. URL https://aclanthology.org/2022.findings-acl.207/.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. Gliner: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5364–5376, 2024.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*, 2024.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024.

Sizhe Zhou. A simple yet strong baseline for long-term conversational memory of llm agents. *arXiv preprint arXiv:2511.17208*, 2025.

# A Theoretical Analysis of Adaptive Spatiotemporal Optimization

This section formalizes the memory-retention objective used by ProStream and clarifies the scope of its theoretical claims. We first formulate memory maintenance as an online budgeted-retention problem. We then characterize the utility-density rule used by the implemented pruning policy and analyze the boundedness of retained-state maintenance under a fixed memory budget. The analysis does not claim a global approximation guarantee for the complete online pipeline, which also includes semantic merging, distillation, and answer generation.

## A.1 Problem Formulation: Dynamic Budgeted Retention

We model streaming dialogue memory as a budgeted retention problem with time-varying utilities. Let $\mathcal{S} = \{I_1, I_2, \dots\}$ denote an unbounded dialogue stream. At time $t$, the system maintains a set of retained memory nodes $\mathcal{H}_t$. Each node $v$ has a positive storage cost $c_v > 0$ and a time-varying utility $u_{v,t}$. The retained state must satisfy

$$\sum_{v \in \mathcal{H}_t} c_v \le \mathcal{T}_{\max}, \tag{3}$$

where $\mathcal{T}_{\max}$ denotes the memory budget reserved for retained memory under the context-window constraint.

Let $\mathcal{T}_{\mathrm{block},t}$ denote the semantic block produced by the stream-perception stage at time $t$. The candidate set for a memory update is

$$\mathcal{H}_{\mathrm{cand},t} = \mathcal{H}_{t-1} \cup \mathcal{M}_{\mathrm{pend},t} \cup \mathrm{Distill}(\mathcal{T}_{\mathrm{block},t}), \tag{4}$$

where $\mathcal{M}_{\mathrm{pend},t}$ contains newly distilled representations that have not yet been mounted onto the hierarchical tree. The per-update retention objective is to select a feasible subset:

$$\mathcal{H}_t^* = \operatorname*{arg\,max}_{\mathcal{H} \subseteq \mathcal{H}_{\mathrm{cand},t}} \sum_{v \in \mathcal{H}} u_{v,t} \quad \text{s.t.} \quad \sum_{v \in \mathcal{H}} c_v \le \mathcal{T}_{\max}. \tag{5}$$

The long-horizon design objective is to preserve high-utility information over an unbounded stream; Eq. 5 specifies the budgeted selection performed at an individual memory update.

## A.2 Utility Formulation

Drawing on the recency–frequency intuition of rational memory models (Anderson, 2013), we use access frequency and temporal recency as two signals of future retrieval demand. For a node $v$, its utility is defined as

$$u_{v,t} = \alpha \log(f_{v,t} + 1) + \beta \exp\left(-\frac{\Delta t_v}{\tau}\right), \tag{6}$$

where $f_{v,t}$ denotes the cumulative access frequency, $\Delta t_v$ denotes the elapsed time since the most recent access, $\tau$ is a temporal decay scale, and $\alpha, \beta$ balance the two signals. This score is a practical retention surrogate motivated by recency–frequency effects; it should not be interpreted as an exact posterior probability or as a direct derivation of a Poisson process.

**Proposition A.1** (Utility as a Retrieval-Oriented Surrogate)**.** *Under a retrieval policy in which frequently accessed and recently used nodes are more likely to be queried again, Eq. 6 provides a monotone surrogate for future retrieval demand: increasing either access frequency or recency increases the retention score, with the relative contributions controlled by $\alpha$ and $\beta$.*

*Proof.* Both terms in Eq. 6 are non-decreasing in their corresponding signals: $\log(f_{v,t} + 1)$ increases with $f_{v,t}$, while $\exp(-\Delta t_v/\tau)$ increases as the elapsed time $\Delta t_v$ decreases. Therefore, for non-negative $\alpha$ and $\beta$, the score prioritizes nodes that are more frequently accessed or more recently used. This establishes the intended ranking property of the surrogate, but does not imply that it equals the true probability of future retrieval. □

### A.3 Utility-Density Retention Rule

The implemented policy restores feasibility by ranking retained nodes according to utility per storage cost. Let

$$\rho_{v,t} = \frac{u_{v,t}}{c_v} \tag{7}$$

be the utility density of node $v$. When the current state exceeds the budget, ProStream removes nodes in increasing order of $\rho_{v,t}$ until the budget constraint is satisfied.

**Proposition A.2** (Local Utility-Density Retention Rule)**.** *For a fixed memory state and a fixed utility estimate, removing the node with the smallest utility density $u_{v,t}/c_v$ provides a local feasibility-restoration rule that prioritizes information with higher utility per token.*

*Proof.* Let $F(\mathcal{H}) = \sum_{v \in \mathcal{H}} u_{v,t}$ be the additive retention objective. Removing node $v$ reduces the objective by $u_{v,t}$ while freeing $c_v$ units of storage. The ratio $u_{v,t}/c_v$ therefore measures the utility lost per unit of released capacity. Removing the smallest-density node minimizes this local loss-to-capacity ratio among the currently retained nodes. This is a local reverse-pruning rule; it is not claimed to be a global solution to the general 0/1 knapsack problem. □

The complete ProStream update additionally applies semantic merging and cascading abstraction. Because these operations modify the node set and may change the representation cost, the proposition above characterizes only the local pruning component and does not provide an approximation guarantee for the complete online policy.

### A.4 Bounded Retained-State Maintenance

A key property of a streaming memory system is that the cost of maintaining its retained state should not grow with the accumulated stream length. The following result concerns the utility-update and pruning operations over the retained memory state; it does not include ASR, LLM-based distillation, semantic relation induction, answer generation, or other fixed per-block costs.

**Theorem A.3** (Bounded Retained-State Maintenance)**.** *Assume a fixed memory budget $\mathcal{T}_{\max}$ and a positive lower bound $c_{\min}$ on node storage costs. Let $N_t = |\mathcal{H}_t|$. Then*

$$N_t \le N_{\max} = \left\lfloor \frac{\mathcal{T}_{\max}}{c_{\min}} \right\rfloor. \tag{8}$$

*Consequently, utility recalculation and priority-queue pruning over the retained state require $O(N_{\max} + k \log N_{\max})$ time per update, where $k$ is the number of evicted nodes. This cost is bounded with respect to the total number of dialogue turns.*

*Proof.* The capacity constraint and the assumption $c_v \ge c_{\min} > 0$ imply Eq. 8. Recomputing utilities for the retained nodes costs $O(N_{\max})$. If evicted nodes are selected using a priority queue, removing $k$ nodes costs $O(k \log N_{\max})$. Since $N_{\max}$ depends on the fixed memory budget rather than on the cumulative number of dialogue turns, these retained-state operations are bounded with respect to the stream length. Feasibility is restored after finitely many removals because each removal strictly decreases the retained storage cost. □

The theorem establishes boundedness with respect to the accumulated stream length for the analyzed retained-state operations. It should not be read as a claim that the full end-to-end latency of ProStream is independent of all input-dependent processing costs.

## B Implementation and Reproducibility Details

We provide the implementation and prompting configuration used in ProStream. Any method-specific deviation in a baseline configuration should be reported separately.

### B.1 System Configuration

We use a unified configuration across the main experiments. Audio streams are transcribed using Whisper large-v3 (Radford et al., 2023) at a 16-kHz sampling rate with 1-second audio chunks. Dialogue units are embedded using ms-marco-MiniLM-L6-v2 (Wang et al., 2020) for semantic encoding and similarity computation. *Proactive Semantic Stream Perception* uses a fixed drift threshold of $\tau_{\text{drift}} = 0.7$, a buffer flush capacity of 5, and a carry-over context window of $W_{\text{STSB}} = 10$. The buffer capacity determines when the current semantic block is flushed, whereas $W_{\text{STSB}}$ determines the amount of context carried across the detected boundary.

*Hierarchical Multi-Granular Distillation* preserves maximal granularity during initial node construction by disabling initial merging and applies the concept-similarity threshold used for AMU novelty filtering. It uses LaMini-T5-738M for scene and event summarization, GLiNER for candidate entity extraction, and an LLM-based relation-induction step for inferring relations between the extracted entities. The relation-induction step follows the prompt and decoding configuration used in the implementation.

*Adaptive Spatiotemporal Optimization* balances frequency and recency using fixed utility weights $\alpha = 0.6$ and $\beta = 0.4$. *Probabilistic Evidence-Grounded Generation* retrieves a small set of ranked scene and event nodes and generates answers from the most relevant AMUs under the stated similarity constraint. All experiments use Qwen-based answer generators on NVIDIA GPUs with PyTorch memory optimization enabled.

### B.2 Prompt Templates

This section provides the prompt templates used for hierarchical distillation and answer generation.

**1) Event Summarization Prompt ($\rho_{\text{evt}}$)**

```
Summarize this conversation in one short phrase describing what happened.
Speakers: {speaker_list}.
Event phrase: {text}
```

**2) Scene Classification Prompt ($\rho_{\text{scn}}$)**

This prompt abstracts an event summary into a higher-level scene category:

```
Abstract this specific event into a broader, more general scene category (e.g.,
'Work Discussion', 'Social Chat', 'Learning Session').
The scene should be a higher-level category that could contain multiple similar
↪ events.
Event: {event_summary}
Scene category (broader abstraction):
```

**3) Question Answering Prompt**

The question-answering prompt integrates recent context, pending memory, and retrieved long-term memory:

```
Please carefully think about the following question and provide a direct,
concise answer with key points only.
Do NOT start your answer with phrases like "Based on..." or "According to...".
Instead, directly state the answer.
## Short-term Memory: {stsb_buffer_content}
## Pending Memory Buffer: {pending_buffer_content}
## Long-term Memory: **{scene_1} > {event_1}** ({count} AMU(s)):
- {amu_1} - {amu_2} ...
**{scene_2} > {event_2}** ({count} AMU(s)):
- {amu_1} - {amu_2} ...
Question: {question_text}
Answer (provide key points directly, no introductory phrases):
```

**Short-term Memory** contains recent dialogue content from the STSB that has not yet been flushed to long-term memory. **Pending Memory Buffer** contains distilled scene, event, and entity representations that have not yet been mounted onto the hierarchical tree. **Long-term Memory** contains the retrieved paths from the hierarchical tree. Each path follows the structure $\{\text{scene}\} > \{\text{event}\} > \{\text{AMU}\}$.

When using Qwen models, the following system message is prepended:

```
You are a helpful assistant that provides direct, concise answers.
Always answer directly without starting with phrases like 'Based on...'
or 'According to...'. Provide key points only.
```

When no context is available, we use:

```
Please answer the following question directly with key points only.
Do NOT start your answer with phrases like "Based on..." or "According to...".
Question: {question_text}
Answer:
```

### B.3 Evaluation Metrics

We evaluate ProStream using both response-quality and efficiency metrics. Unless otherwise stated, all metrics are computed on the same question set, with the same ground-truth answers and evidence annotations across methods.

**Performance Metrics.** BLEU-4 (**B-4**) and ROUGE-L (**R-L**) measure lexical overlap and sequence-level similarity between the generated answer and the reference answer. We report the standard BERTScore F1 (**B-S**) using DeBERTa-v3-xlarge contextual embeddings. Given a candidate answer $C$ and a reference answer $R$, BERTScore computes precision and recall through bidirectional maximum cosine-similarity matching:

$$P_{\text{BS}} = \frac{1}{|C|} \sum_{c_i \in C} \max_{r_j \in R} \cos(\mathbf{e}_{c_i}, \mathbf{e}_{r_j}), \tag{9}$$

$$R_{\text{BS}} = \frac{1}{|R|} \sum_{r_j \in R} \max_{c_i \in C} \cos(\mathbf{e}_{r_j}, \mathbf{e}_{c_i}), \tag{10}$$

where $\mathbf{e}_{c_i}$ and $\mathbf{e}_{r_j}$ denote the contextual embeddings of candidate and reference tokens, respectively. The reported BERTScore is the harmonic mean:

$$\text{B-S} = \frac{2P_{\text{BS}}R_{\text{BS}}}{P_{\text{BS}} + R_{\text{BS}}}. \tag{11}$$

Key Entity Matching (**KEM**) measures whether the essential entities required by the ground-truth evidence are correctly mentioned in the generated answer. Let $\mathcal{E}_{\text{gt}}(q)$ denote the set of required entities extracted from the ground-truth evidence for query $q$, and let $\mathcal{E}_{\text{ans}}(q)$ denote the entities identified in the generated answer. After case, punctuation, and whitespace normalization, we define the instance-level KEM score as

$$\text{KEM}(q) = \mathbb{I}\left[\mathcal{E}_{\text{gt}}(q) \subseteq \mathcal{E}_{\text{ans}}(q)\right], \tag{12}$$

where $\mathbb{I}[\cdot]$ is one if all required entities are matched and zero otherwise. The final KEM score is averaged over all evaluation instances:

$$\text{KEM} = \frac{1}{N} \sum_{i=1}^{N} \text{KEM}(q_i). \tag{13}$$

Evidence Similarity (**Evid.**) measures the alignment between the retrieved context and the ground-truth evidence. Specifically, we compute the cosine similarity between the retrieved context $C_{\text{ret}}$ and the ground-truth evidence $E_{\text{gt}}$ using the SentenceTransformer encoder:

$$\text{Evid.} = \cos\left(f_{\text{ST}}(C_{\text{ret}}), f_{\text{ST}}(E_{\text{gt}})\right). \tag{14}$$

Table 2: Results of different embedding models on STEM-Bench. Metrics marked with ↑ indicate higher is better, while ↓ denotes lower is better.

| Backbone | Variant | The Big Bang Theory | | | | | | | Friends | | | | | | | The Office | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Time↓ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | Gem↑ | Time↓ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | Gem↑ | Time↓ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | Gem↑ |
| BGE | RAG | 0.814 | 0.087 | 0.301 | 0.796 | 0.583 | 0.572 | 0.413 | 0.814 | 0.081 | 0.296 | 0.785 | 0.621 | 0.576 | 0.399 | 0.801 | 0.094 | 0.339 | 0.796 | 0.647 | 0.611 | 0.413 |
| | ours | 0.393 | 0.116 | 0.383 | 0.816 | 0.460 | 0.623 | 0.498 | 0.442 | 0.097 | 0.354 | 0.802 | 0.464 | 0.597 | 0.469 | 0.516 | 0.114 | 0.407 | 0.819 | 0.484 | 0.622 | 0.498 |
| UAE | RAG | 0.774 | 0.086 | 0.296 | 0.794 | 0.583 | 0.572 | 0.410 | 0.788 | 0.082 | 0.296 | 0.785 | 0.619 | 0.573 | 0.408 | 0.756 | 0.093 | 0.337 | 0.796 | 0.650 | 0.612 | 0.410 |
| | ours | 0.825 | 0.116 | 0.386 | 0.818 | 0.482 | 0.615 | 0.501 | 0.829 | 0.098 | 0.357 | 0.802 | 0.482 | 0.592 | 0.471 | 0.688 | 0.117 | 0.413 | 0.821 | 0.503 | 0.624 | 0.501 |
| MiniLM | RAG | 1.225 | 0.064 | 0.240 | 0.781 | 0.389 | 0.500 | 0.331 | 1.162 | 0.060 | 0.237 | 0.774 | 0.424 | 0.497 | 0.309 | 1.067 | 0.072 | 0.269 | 0.781 | 0.466 | 0.533 | 0.331 |
| | ours | 0.408 | 0.118 | 0.390 | 0.817 | 0.485 | 0.612 | 0.509 | 0.373 | 0.099 | 0.360 | 0.801 | 0.491 | 0.588 | 0.476 | 0.316 | 0.114 | 0.408 | 0.819 | 0.491 | 0.626 | 0.509 |

Table 3: Sensitivity to the drift threshold $\tau_{\text{drift}}$ and concept threshold $\theta_{\text{concept}}$. 'Evid.' denotes Evidence Similarity.

**(a) Sensitivity to $\tau_{\text{drift}}$**

| Setting | The Big Bang Theory | | | | | Friends | | | | | The Office | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ |
| 0.5 | 0.15 | 0.43 | 0.85 | 0.51 | 0.64 | 0.12 | 0.39 | 0.82 | 0.52 | 0.62 | 0.14 | 0.46 | 0.84 | 0.50 | 0.69 |
| 0.6 | 0.16 | 0.44 | 0.85 | 0.51 | 0.66 | 0.12 | 0.40 | 0.82 | 0.52 | 0.63 | 0.14 | 0.46 | 0.83 | 0.50 | 0.68 |
| 0.7 | 0.12 | 0.39 | 0.82 | 0.49 | 0.61 | 0.10 | 0.36 | 0.80 | 0.49 | 0.59 | 0.11 | 0.41 | 0.82 | 0.49 | 0.63 |
| 0.8 | 0.16 | 0.45 | 0.85 | 0.49 | 0.68 | 0.11 | 0.39 | 0.82 | 0.52 | 0.62 | 0.14 | 0.46 | 0.83 | 0.51 | 0.67 |
| 0.9 | 0.16 | 0.45 | 0.85 | 0.49 | 0.66 | 0.12 | 0.40 | 0.82 | 0.52 | 0.63 | 0.15 | 0.47 | 0.83 | 0.50 | 0.70 |

**(b) Sensitivity to $\theta_{\text{concept}}$**

| Setting | The Big Bang Theory | | | | | Friends | | | | | The Office | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ |
| 0.4 | 0.15 | 0.44 | 0.85 | 0.49 | 0.67 | 0.11 | 0.39 | 0.82 | 0.52 | 0.62 | 0.15 | 0.48 | 0.84 | 0.51 | 0.70 |
| 0.6 | 0.16 | 0.45 | 0.85 | 0.49 | 0.67 | 0.12 | 0.40 | 0.82 | 0.52 | 0.66 | 0.15 | 0.47 | 0.84 | 0.50 | 0.69 |
| 0.8 | 0.16 | 0.46 | 0.85 | 0.49 | 0.65 | 0.11 | 0.40 | 0.82 | 0.52 | 0.63 | 0.15 | 0.47 | 0.84 | 0.50 | 0.67 |
| 1.0 | 0.16 | 0.44 | 0.85 | 0.49 | 0.66 | 0.12 | 0.40 | 0.82 | 0.52 | 0.61 | 0.15 | 0.46 | 0.83 | 0.50 | 0.68 |

We additionally use Gemini-2.5-Pro (**Gem**) as a reference-guided LLM judge to evaluate answer correctness. The judge receives the question, the candidate answer, and the ground-truth answer, and returns a score in $[0, 1]$ together with a short textual justification. The reported Gem score is the average score over all evaluation instances.

**Efficiency Metrics.** We report the average end-to-end response latency (**Time**, in seconds) as the primary efficiency metric. For each question probe, the timer starts when the probe becomes available to the system and stops when the final answer is completed. Thus, Time includes all synchronous operations on the response-critical path, including query encoding, memory retrieval, context construction, and answer generation. Any background memory maintenance completed before the probe arrives is excluded from this measurement. The online memory-maintenance cost is measured separately per memory flush. It includes STSB processing, hierarchical distillation, and hierarchy update, and is reported as memory-update latency rather than query latency. Accordingly, the end-to-end response latency reported in the main experiments and the per-flush memory-maintenance cost reported in Appendix E should not be conflated.

## C Analysis of Different Embedding Models

To examine the robustness of ProStream to the semantic encoder, we evaluate it with three embedding models: *ms-marco-MiniLM-L6-v2* (Wang et al., 2020), *UAE-Large-V1* (Li & Li, 2023), and *bge-m3* (Chen et al., 2024). As shown in Table 2, ProStream improves several response-quality metrics and latency across embedding models, although retrieval-based RAG may retain an advantage on evidence-similarity scores. These results suggest that the observed gains are not restricted to a single encoder, while not isolating the effects of embedding dimensionality from other encoder properties.

Table 4: Sensitivity to memory capacity $\mathcal{T}_{\max}$. Metrics reported are BLEU-4 (B-4), ROUGE-L (R-L), BERTScore (B-S), Evidence Similarity (Evid.), and KEM.

| $\mathcal{T}_{\max}$ | The Big Bang Theory | | | | | Friends | | | | | The Office | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ | B-4↑ | R-L↑ | B-S↑ | Evid.↑ | KEM↑ |
| 50k | 0.15 | 0.41 | 0.82 | 0.16 | 0.53 | 0.11 | 0.35 | 0.79 | 0.23 | 0.48 | 0.15 | 0.44 | 0.82 | 0.21 | 0.57 |
| 75k | 0.15 | 0.40 | 0.82 | 0.16 | 0.53 | 0.11 | 0.36 | 0.78 | 0.23 | 0.50 | 0.15 | 0.45 | 0.82 | 0.21 | 0.57 |
| 100k | 0.16 | 0.41 | 0.82 | 0.16 | 0.54 | 0.11 | 0.36 | 0.79 | 0.23 | 0.49 | 0.15 | 0.44 | 0.82 | 0.21 | 0.57 |
| 125k | 0.15 | 0.41 | 0.82 | 0.16 | 0.54 | 0.11 | 0.37 | 0.79 | 0.23 | 0.50 | 0.15 | 0.45 | 0.82 | 0.21 | 0.57 |
| 150k | 0.15 | 0.40 | 0.82 | 0.16 | 0.53 | 0.11 | 0.35 | 0.78 | 0.23 | 0.49 | 0.15 | 0.44 | 0.82 | 0.21 | 0.57 |

# D Additional Robustness Analysis

To further examine the robustness of ProStream, we provide additional analyses: (1) a threshold sensitivity study over $\tau_{\text{drift}}$ and $\theta_{\text{concept}}$ and (2) the sensitivity analysis of memory capacity $\mathcal{T}_{\max}$.

## D.1 Hyperparameter Sensitivity

We first study the sensitivity of ProStream to the drift threshold $\tau_{\text{drift}}$. We sweep $\tau_{\text{drift}}$ from 0.5 to 0.9 and observe consistently stable results across all datasets and metrics, indicating that ProStream does not rely on brittle drift-threshold tuning. We further vary the concept threshold $\theta_{\text{concept}}$ from 0.4 to 1.0. As shown in Table 3, performance remains consistently stable, suggesting that the framework operates in a robust regime without requiring precise threshold tuning.

## D.2 Sensitivity to Memory Capacity

The memory capacity of ProStream is controlled by the token budget $\mathcal{T}_{\max}$. We therefore conduct a sensitivity study by varying $\mathcal{T}_{\max}$ from 50k to 150k tokens. As shown in Table 4, performance remains stable across this range, with the best results typically appearing around 100k–125k. This suggests that ProStream benefits from a moderate memory budget, while not relying on over-provisioned capacity.

# E Cost Analysis of the ProStream Pipeline

The main paper reports average end-to-end response latency per question probe. To characterize the additional online memory-maintenance overhead, we instrument each memory flush and measure STSB processing, hierarchical distillation, and hierarchy update separately. The average runtime is 0.0046 seconds for STSB processing, 0.7231 seconds for hierarchical distillation, and 0.0203 seconds for hierarchy update, resulting in a total of 0.7434 seconds per memory update. Hierarchical distillation is the dominant component, whereas the hierarchy update itself is relatively lightweight. These measurements quantify the per-flush memory-maintenance cost and are reported separately from the end-to-end response latency used in the main experiments.

# F Evaluation on Existing Conversation-Memory Benchmarks

To examine transfer beyond STEM-Bench, we evaluate ProStream on LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2024a). In both cases, each conversation is replayed chronologically as a bounded-state text stream, and ProStream performs online memory maintenance without access to future context. Because the reported baselines may use different backbones and evaluation protocols, these results should be interpreted as indicative transfer evidence rather than strictly controlled state-of-the-art comparisons. The exact backbone, prompt, memory budget, and scoring protocol used by ProStream should be reported alongside each table.

Table 5: Transfer evaluation on LoCoMo. Results are reported for reference because the compared methods may use different backbones and protocols. Best values within each column are bolded.

| Method | Single-hop | Multi-hop | Temporal | Open-domain |
|---|---|---|---|---|
| Mistral-7B-Instruct-v0.2 | 9.1 | 15.1 | 9.3 | 8.6 |
| Llama-2-70B-chat | 20.8 | 18.2 | 15.9 | 18.8 |
| Llama-3-70B-Instruct | 17.0 | 17.0 | 12.0 | 13.0 |
| gpt-3.5-turbo-4K | 23.8 | 18.0 | 15.6 | 20.4 |
| gpt-3.5-turbo-8K | **38.5** | 25.1 | 22.7 | **25.9** |
| ProStream | 25.5 | **31.3** | **38.8** | 20.4 |

Table 6: Transfer evaluation on LongMemEval. Results are reported for reference because the compared methods may use different backbones and protocols.

| Method | Accuracy |
|---|---|
| Vanilla | 10.0 |
| RecurSum | 10.0 |
| ConditionMem | 40.0 |
| MemoryBank | 23.33 |
| Mem-PAL | 46.67 |
| ProStream | **50.0** |

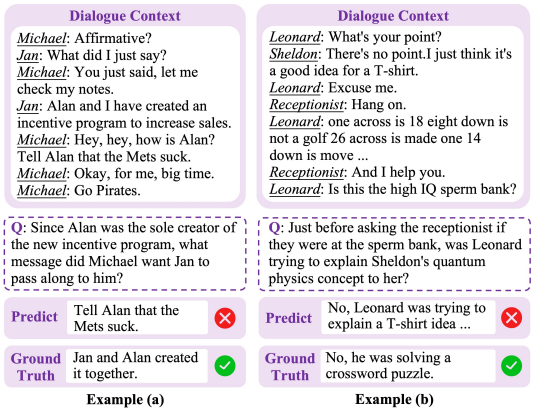



Figure 7: Two representative failure cases of ProStream.

## G Extended Error Analysis

To characterize the operational boundaries of ProStream, we analyze two recurring error modes.

**Instruction–Fidelity Tension.** Figure 7(a) shows a case in which the model follows a user instruction containing a false presupposition. When the prompt claims that Alan was the "sole creator," the model accepts the premise despite dialogue evidence that Jan and Alan created the program together. This behavior indicates a conflict between instruction following and factual consistency and suggests the need for explicit contradiction detection.

**Semantic Noise and Intent Misattribution.** Figure 7(b) shows a case in which salient semantic content obscures a less prominent activity cue. A dense monologue about T-shirt ideas and quantum physics dominates the context, while the fragmented crossword-puzzle cues, such as "one across is 18," are overlooked. The resulting answer misattributes the user's intent by treating the puzzle-solving activity as a continuation of the previous topic.

These failure modes motivate future work on contradiction-aware memory updating and activity-centric attention mechanisms.