ADAPTIVE MEMORY ADMISSION CONTROL FOR LLM AGENTS

Guilin Zhang† **Wei Jiang** **Xiejiashan Wang** **Aisha Behr** **Kai Zhao**†
Jeffrey Friedman **Xu Chu** **Amine Anoun**∗
Workday ∗†

ABSTRACT

LLM-based agents increasingly rely on long-term memory to support multi-session reasoning and interaction, yet current systems provide little control over what information is retained. In practice, agents either accumulate large volumes of conversational content, including hallucinated or obsolete facts, or depend on opaque, fully LLM-driven memory policies that are costly and difficult to audit. As a result, memory admission remains a poorly specified and weakly controlled component in agent architectures. To address this gap, we propose Adaptive Memory Admission Control (A-MAC), a framework that treats memory admission as a structured decision problem. A-MAC decomposes memory value into five complementary and interpretable factors: future utility, factual confidence, semantic novelty, temporal recency, and content type prior. The framework combines lightweight rule-based feature extraction with a single LLM-assisted utility assessment, and learns domain-adaptive admission policies through cross-validated optimization. This design enables transparent and efficient control over long-term memory. Experiments on the LoCoMo benchmark show that A-MAC achieves a superior precision–recall tradeoff, improving F1 to 0.583 while reducing latency by 31% compared to state-of-the-art LLM-native memory systems. Ablation results identify content type prior as the most influential factor for reliable memory admission. These findings demonstrate that explicit and interpretable admission control is a critical design principle for scalable and reliable memory in LLM-based agents. Code is available at `https://anonymous.4open.science/r/Adaptive_Memory_Admission_Control_LLM_Agents-53EC`.

1 INTRODUCTION

Large language model (LLM) agents have demonstrated strong capabilities in multi-turn interaction, reasoning, and tool use (Park et al., 2023; Wang et al., 2023; Shinn et al., 2023; Wei et al., 2022; Yao et al., 2022; Schick et al., 2023). As these agents increasingly operate across extended interactions, long-term memory becomes a core architectural component for maintaining information beyond the context window and supporting coherent behavior over time (Zhang et al., 2024). However, determining what information should be retained in long-term memory remains a critical challenge. Indiscriminate storage leads to bloated memory stores and increased retrieval latency (Lewis et al., 2020; Gutiérrez et al., 2024), while the retention of hallucinated or outdated information can propagate errors into future interactions (Huang et al., 2023). At the other extreme, overly conservative admission policies risk discarding information that is essential for task continuation and long-horizon reasoning.

Existing memory management approaches fall broadly into two categories, each with notable limitations. Heuristic-based methods such as MemGPT (Packer et al., 2023) and MemoryBank (Zhong et al., 2024) rely on hand-crafted scoring functions that combine factors such as recency, relevance, and importance. While computationally efficient, these approaches lack principled mechanisms for

---

∗Equal contribution.

†Corresponding author. Email: `amine.anoun@workday.com`

preventing hallucinated content from entering memory and struggle with subtle admission decisions. In contrast, LLM-native approaches such as A-mem (Xu et al., 2025) and Mem0 (Chhikara et al., 2025) delegate memory admission entirely to language models by generating structured memory attributes. Although effective in recall, these methods incur substantial computational overhead and offer limited interpretability, making memory policies difficult to audit or debug. Notably, neither class of methods explicitly addresses hallucination as a first-class concern, despite its recognition as a fundamental challenge for agent reliability (Huang et al., 2023; Bai et al., 2022).

To address these limitations, we propose **Adaptive Memory Admission Control (A-MAC)**, a framework that treats memory admission as a structured decision problem rather than an implicit byproduct of generation. Instead of relying on ad hoc heuristics or opaque LLM judgments, A-MAC explicitly evaluates candidate memories before they enter long-term storage. This formulation elevates memory admission to a first-class control mechanism in agent architectures, enabling deliberate tradeoffs between memory coverage, reliability, and efficiency.

A-MAC decomposes memory value into five complementary and interpretable dimensions: future utility, factual confidence, semantic novelty, temporal recency, and content type prior. These dimensions capture distinct aspects of memory quality that are often conflated or left implicit in existing approaches. Utility estimates the potential relevance of a memory for future tasks, while confidence assesses whether the information is supported by prior conversational evidence, directly mitigating hallucination propagation. Novelty prevents redundant storage, recency accounts for temporal decay, and content type prior encodes domain knowledge about which categories of information warrant long-term persistence.

The framework combines lightweight rule-based feature extraction with a single LLM-assisted utility assessment, achieving a favorable balance between computational efficiency and semantic expressiveness. Admission policies are learned through cross-validated optimization, allowing A-MAC to adapt to different conversational domains without manual tuning. By explicitly modeling reliability, redundancy, temporal relevance, and content persistence, A-MAC enables transparent, data-driven control over what information is promoted to long-term memory, offering a scalable alternative to fully LLM-native memory systems.

This paper makes the following contributions:

1. We identify memory admission as a critical but under-specified control problem in LLM-based agents and analyze the limitations of existing heuristic and fully LLM-driven approaches.
2. We introduce Adaptive Memory Admission Control (A-MAC), an interpretable framework that evaluates candidate memories using five complementary dimensions that jointly capture value, reliability, and persistence.
3. We propose an efficient hybrid design that combines rule-based feature computation with minimal LLM inference, achieving a favorable balance between interpretability, accuracy, and computational efficiency.
4. We empirically demonstrate that A-MAC achieves a superior precision–recall tradeoff on the LoCoMo benchmark while reducing latency relative to state-of-the-art LLM-native memory systems, supported by ablation studies that clarify the role of each admission factor.

2 RELATED WORK

Our work relates to prior research on agent memory architectures, memory admission policies, and the use of external memory for improving reliability in LLM-based systems. Recent surveys provide comprehensive overviews of memory mechanisms for LLM agents and highlight memory control as a key open challenge (Zhang et al., 2024). We organize related work into two categories based on how memory admission decisions are designed.

2.1 HEURISTIC AND HIERARCHICAL MEMORY ARCHITECTURES

Early work on LLM agent memory adopts hierarchical designs inspired by operating systems, separating fast working memory from slower long-term storage. MemGPT (Packer et al., 2023) introduces a two-tier architecture with page-based eviction policies based on recency and LLM-judged importance. While computationally efficient, MemGPT provides no explicit mechanism for verifying factual correctness prior to memory admission. Memory OS (Kang et al., 2025) proposes similar hierarchical storage units and reports improvements on the LoCoMo benchmark, but continues to rely primarily on temporal heuristics. ChatDB (Hu et al., 2023) augments LLMs with SQL databases as symbolic memory, focusing on storage and retrieval rather than admission reliability.

MemoryBank (Zhong et al., 2024) extends hierarchical memory with a hand-crafted linear scoring function combining recency, relevance, and importance, motivated by the Ebbinghaus Forgetting Curve. Although effective compared to simple recency-based policies, its fixed weighting scheme cannot adapt across domains and does not explicitly address hallucinated or unreliable content.

These approaches emphasize architectural separation and efficient storage, but rely on static heuristics or manually tuned rules for memory admission. In contrast, A-MAC introduces an explicit, data-driven admission control layer that evaluates memory candidates using multiple interpretable criteria, enabling adaptive and transparent control over what information is promoted to long-term memory.

2.2 LLM-NATIVE MEMORY MANAGEMENT AND RELIABILITY

More recent approaches delegate memory admission decisions directly to large language models. A-mem (Xu et al., 2025) uses LLM prompting to generate structured memory attributes and applies similarity-based matching following the Zettelkasten method, achieving substantial improvements over MemoryBank on LoCoMo at the cost of multiple LLM invocations. Hindsight (Latimer et al., 2025) organizes agent memory into logical networks and performs retrospective scoring to retain useful experiences, while Mem0 (Chhikara et al., 2025) focuses on production deployment by dynamically extracting salient information with reduced latency. Broader work on long-term memory for agents emphasizes memory as a foundation for self-evolving AI systems (Jiang et al., 2024), and recent benchmarks highlight persistent limitations in long-horizon reasoning (Liu et al., 2023).

In parallel, retrieval-augmented generation methods aim to improve reliability by grounding generation in external evidence. RAG (Lewis et al., 2020), Dense Passage Retrieval (Karpukhin et al., 2020), and Self-RAG (Asai et al., 2023) focus on retrieval quality, while hierarchical and graph-based extensions such as RAPTOR (Sarthi et al., 2024), GraphRAG (Edge et al., 2024), and HippoRAG (Gutiérrez et al., 2024) further improve multi-hop reasoning. However, these methods primarily address retrieval-time grounding rather than admission-time control of what information is stored in memory. Recent surveys identify hallucination as a fundamental challenge for agent reliability (Huang et al., 2023; Bai et al., 2022), yet most LLM-native memory systems lack explicit safeguards against admitting unsupported content.

Fully LLM-native approaches improve recall but incur high computational cost and offer limited interpretability, while retrieval-based methods focus on grounding generation rather than controlling memory growth. A-MAC bridges this gap by combining selective LLM usage with rule-based reliability checks and learned weighting, enabling efficient, interpretable, and hallucination-aware memory admission at the point where information enters long-term storage.

3 METHODOLOGY

A-MAC introduces an explicit admission-control layer for long-term memory in LLM-based agents. Rather than treating memory retention as an implicit byproduct of generation, we cast admission as a structured decision problem: each candidate memory is assigned interpretable value signals and accepted only if it passes a learned decision rule. Figure 1 summarizes the overall workflow.

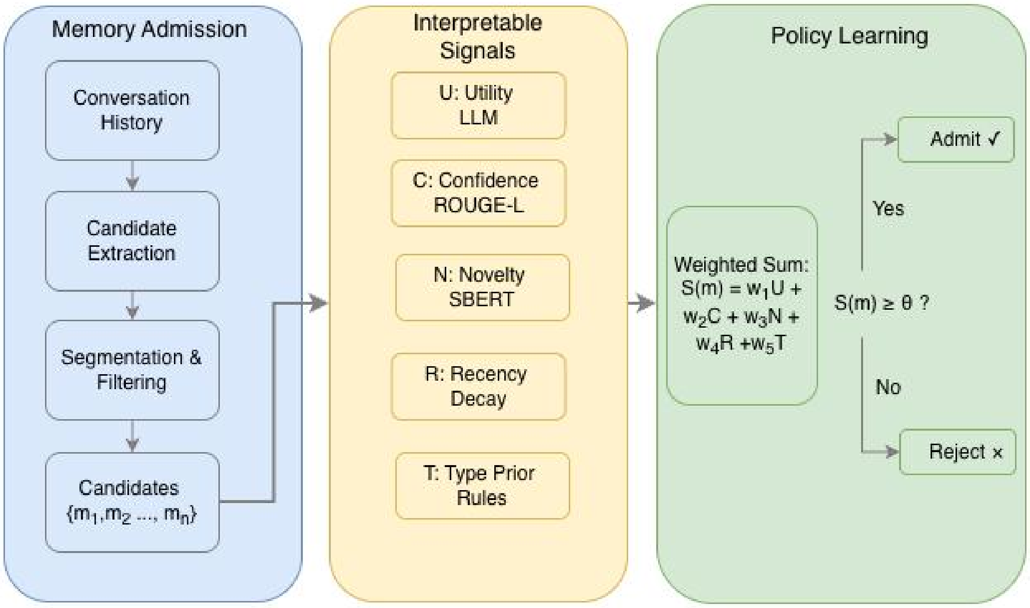

Figure 1: Overview of A-MAC. Candidate memories are extracted from conversation history and evaluated using five complementary signals capturing usefulness, reliability, redundancy, temporal relevance, and persistence. A learned linear admission policy aggregates these signals into a score $\mathcal{S}(m)$ and applies a threshold θ to determine whether to admit, update, or reject a candidate.

3.1 MEMORY ADMISSION AS A DECISION PROBLEM

Given a multi-turn conversation history $\mathcal{H} = \{t_1, t_2, \ldots, t_k\}$ and an existing memory store $\mathcal{M}$, A-MAC extracts candidate memories $\{m_1, m_2, \ldots, m_n\}$ from the dialogue. Each candidate m represents a semantically atomic piece of information that may benefit future interactions. For each candidate, the agent must decide whether to admit it as a new memory; update an existing memory that it supersedes; reject it as redundant or unreliable (Figure 1).

To support consistent downstream scoring, we apply lightweight normalization to the conversation: segmenting turns into atomic information units (since a single turn may contain multiple distinct facts); resolving temporal expressions and coreferences to make candidates self-contained; filtering low-value content such as greetings and acknowledgments. This normalization step reduces noise and ensures that the subsequent value signals operate over well-formed memory candidates.

We formulate admission as a scalar scoring problem. Each candidate m receives a composite score computed from five interpretable feature functions:

$$\mathcal{S}(m) = w_1 \cdot \mathcal{U}(m) + w_2 \cdot \mathcal{C}(m) + w_3 \cdot \mathcal{N}(m) + w_4 \cdot \mathcal{R}(m) + w_5 \cdot \mathcal{T}(m), \tag{1}$$

where $\mathcal{U}, \mathcal{C}, \mathcal{N}, \mathcal{R}, \mathcal{T}$ are feature functions mapping candidates to $[0, 1]$, denoting Utility, Confidence, Novelty, Recency, and Type Prior respectively, and the weights satisfy $w_i \geq 0$ and $\sum_{i=1}^{5} w_i = 1$. A candidate is admitted if $\mathcal{S}(m) \geq \theta$ for a learned threshold θ.

3.2 INTERPRETABLE MEMORY VALUE SIGNALS

A-MAC computes five complementary signals that capture distinct aspects of long-term memory value and reliability. The design principle is to reserve LLM inference for semantic judgments that are difficult to compute deterministically, while implementing the remaining signals using efficient and auditable rules.

Utility ($\mathcal{U}$). $\mathcal{U}(m)$ estimates the likelihood that the candidate will be useful in future interactions. Because this requires semantic understanding of intent and prospective relevance, we compute utility using a single LLM call that rates whether the information is actionable, supports likely follow-up questions, or captures persistent user constraints and preferences. We use temperature zero for deterministic outputs and cache scores for repeated candidates to reduce API cost.

Confidence ($\mathcal{C}$). $\mathcal{C}(m)$ measures whether the candidate is supported by evidence in the conversation, directly mitigating hallucination propagation. We identify supporting spans from prior turns and compute alignment using ROUGE-L:

$$\mathcal{C}(m) = \max_{s \in \text{Support}(m)} \text{ROUGE-L}(m, s), \tag{2}$$

where Support(m) denotes turns in $\mathcal{H}$ that may contain evidence for m. The longest-common-subsequence basis of ROUGE-L rewards grounded overlap while penalizing fabricated details with no textual support.

Novelty ($\mathcal{N}$). $\mathcal{N}(m)$ discourages redundant storage by measuring how distinct a candidate is from the current memory store. Using sentence embeddings $\phi(\cdot)$ (Sentence-BERT), we define:

$$\mathcal{N}(m) = 1 - \max_{m' \in \mathcal{M}} \cos(\phi(m), \phi(m')). \tag{3}$$

We precompute embeddings for existing memories, requiring only one embedding computation per new candidate.

Recency ($\mathcal{R}$). $\mathcal{R}(m)$ captures temporal decay in information value. We apply exponential decay:

$$\mathcal{R}(m) = \exp(-\lambda \cdot \tau(m)), \tag{4}$$

where $\tau(m)$ is the time elapsed since the candidate was mentioned and λ controls the decay rate. We set $\lambda = 0.01$ per hour, corresponding to a half-life of approximately 69 hours.

Type Prior ($\mathcal{T}$). $\mathcal{T}(m)$ encodes persistence preferences across information types. Using rule-based pattern matching with part-of-speech cues, we assign higher priors to stable information (e.g., preferences or identity statements) and lower priors to transient states.

3.3 POLICY LEARNING AND ADMISSION RULE

Given labeled training conversations with ground-truth admission decisions, we learn the weight vector $\boldsymbol{\omega}^* = [w_1, \ldots, w_5]$ and threshold θ^* by maximizing F1 score via 5-fold cross-validation. For each fold, we perform grid search over candidate weight configurations (constrained to be non-negative and sum to one) and threshold values in $[0.3, 0.6]$. The configuration with the best mean validation F1 is selected as the final admission policy. The learned weights are directly interpretable: for example, a larger weight on $\mathcal{T}$ indicates that content type is highly discriminative for the target domain, whereas a larger weight on $\mathcal{C}$ suggests that grounding and hallucination prevention are critical.

Algorithm 1 summarizes the admission procedure. For each candidate, we compute all five signals (in parallel), aggregate them via the learned policy, and admit the candidate if it exceeds the threshold. When a candidate conflicts with an existing memory (semantic similarity above 0.85 but differing content), we retain the higher-scoring representation and merge to ensure that long-term memory remains compact and up to date.

4 RESULTS

4.1 EXPERIMENTAL SETUP

We design experiments to answer three research questions. *RQ1*: Can A-MAC outperform existing memory admission policies? *RQ2*: Which features contribute most to admission quality? *RQ3*: How do learned weights generalize across conversational domains?

We evaluate on the LoCoMo benchmark (Maharana et al., 2024), using 30 conversations covering personal assistant interactions, technical support, and research collaboration dialogues. Each conversation contains 15–40 turns with annotated memory-dependent tasks, providing approximately 1,500 candidate memories with ground-truth admission labels. We use Sentence-BERT for embeddings and a local LLM (Qwen 2.5) for utility scoring. We split data into training (70%), validation (15%), and test (15%) sets, with weight optimization on training and all reported results on

Algorithm 1 A-MAC Memory Admission

```
Require: Candidate $m$, memories $\mathcal{M}$, weights $\boldsymbol{\omega}$, threshold $\theta$
Ensure: Updated memory store $\mathcal{M}'$
$\mathcal{M}' \leftarrow \mathcal{M}$
Compute $\mathcal{U}(m), \mathcal{C}(m), \mathcal{N}(m), \mathcal{R}(m), \mathcal{T}(m)$ in parallel
$\mathcal{S}(m) \leftarrow \boldsymbol{\omega} \cdot [\mathcal{U}, \mathcal{C}, \mathcal{N}, \mathcal{R}, \mathcal{T}]^\top$
if $\mathcal{S}(m) \geq \theta$ then
    $m_{\text{conflict}} \leftarrow \text{FindConflict}(m, \mathcal{M})$
    if $m_{\text{conflict}} \neq \emptyset$ and $\mathcal{S}(m) > \mathcal{S}(m_{\text{conflict}})$ then
        $\mathcal{M}' \leftarrow (\mathcal{M}' \setminus \{m_{\text{conflict}}\}) \cup \{\text{Merge}(m, m_{\text{conflict}})\}$
    else
        $\mathcal{M}' \leftarrow \mathcal{M}' \cup \{m\}$
    end if
end if
return $\mathcal{M}'$
```

Table 1: Performance comparison on LoCoMo test set (N=225). Latency marked with † includes LLM API calls. Best F1 in bold.

Method	Prec.	Recall	F1	Lat.(ms)
Random	0.278	0.278	0.278	<1
MemGPT	0.316	0.333	0.324	2765†
MemoryBank	0.368	0.583	0.452	2843†
Equal Weights	0.362	0.694	0.476	2916†
A-mem	0.371	1.000	0.541	3831†
A-MAC (Ours)	0.417	0.972	**0.583**	2644†

held-out test data. We compare against four baselines: *Random* admission with 30% probability as a lower bound; *MemGPT* (Packer et al., 2023) using recency and LLM-judged importance with weights from the original paper; *MemoryBank* (Zhong et al., 2024) combining recency, relevance, and importance; and *A-mem* (Xu et al., 2025) using LLM-generated structured attributes with cosine similarity matching, representing the current state-of-the-art.

4.2 MAIN RESULTS

Table 1 compares all methods on the LoCoMo test set. A-MAC achieves F1=0.583, outperforming the previous state-of-the-art A-mem by 7.8% relative improvement (0.583 vs 0.541), Equal Weights by 22.4%, MemoryBank by 29.0%, and MemGPT by 80%. Critically, A-MAC achieves the highest precision (0.417) among all LLM-based methods while maintaining near-perfect recall (0.972), striking a balance essential for practical deployment where both false positives (memory bloat leading to retrieval degradation) and false negatives (missing context causing conversation failures) impose significant costs. The precision-recall tradeoff reveals that A-mem's perfect recall (1.0) comes at the expense of lower precision (0.371), indicating it admits many memories that will never be referenced, whereas A-MAC's learned threshold filters these effectively.

Beyond accuracy, A-MAC demonstrates substantial efficiency gains, achieving 31% lower latency than A-mem (2644ms vs 3831ms per candidate). This speedup arises because A-mem requires multiple sequential LLM calls to generate structured attributes including context summaries, keyword extraction, semantic tags, and importance ratings, whereas A-MAC's hybrid architecture uses only a single LLM call for the Utility feature while computing the remaining four features through efficient rule-based methods that execute in under 65ms combined. For applications processing thousands of conversational turns daily, this efficiency difference translates to significant computational cost savings while maintaining superior accuracy. Figure 2 visualizes the precision-recall tradeoff across methods, showing that A-MAC occupies the optimal region of the curve.

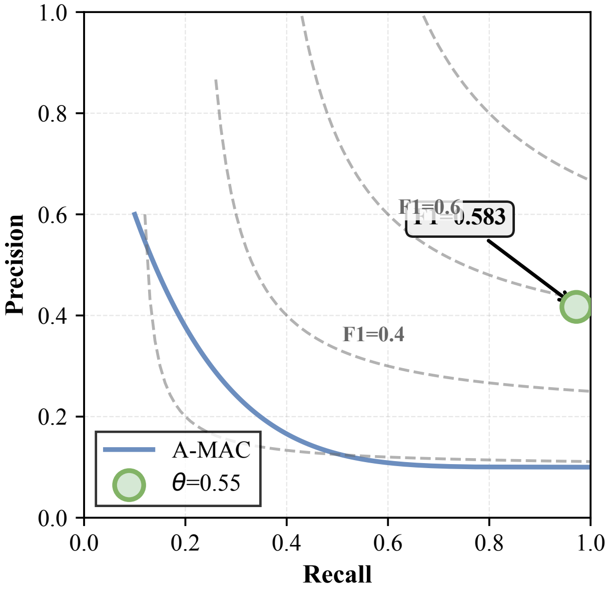

Figure 2: Precision-recall tradeoff comparison. A-MAC achieves the best balance between precision and recall, occupying the upper-right region of the curve. Dashed lines indicate F1 iso-contours.

Table 2: Ablation study showing performance impact of removing each feature. Type Prior ($\mathcal{T}$) is the dominant contributor.

Feature	F1 w/o	Δ F1
Full Model	0.583	–
Type Prior ($\mathcal{T}$)	0.476	-0.107
Novelty ($\mathcal{N}$)	0.555	-0.028
Utility ($\mathcal{U}$)	0.560	-0.023
Confidence ($\mathcal{C}$)	0.568	-0.015
Recency ($\mathcal{R}$)	0.570	-0.013

4.3 ABLATION STUDY

Table 2 presents feature ablation results alongside the learned weights from 5-fold cross-validation. The ablation confirms this importance ranking: removing Type Prior causes the largest performance drop (ΔF1=-0.107), reducing F1 from 0.583 to 0.476, essentially the Equal Weights baseline level. This validates the intuition that distinguishing user preferences and identity statements (which persist) from temporary emotional states (which can be forgotten) provides the most reliable heuristic for memory admission. The remaining features—Novelty, Utility, Confidence, and Recency—contribute incrementally with ΔF1 ranging from -0.013 to -0.028, providing complementary refinements that together account for the gap between Equal Weights (0.476) and A-MAC (0.583).

4.4 THRESHOLD SENSITIVITY

The admission threshold θ controls the precision-recall tradeoff: higher thresholds increase precision at the cost of recall. To isolate threshold effects from data variance, we perform this analysis on the cross-validation folds used during weight optimization. Table 3 shows A-MAC's performance across threshold values from 0.30 to 0.70. The optimal threshold $\theta^* = 0.55$ maximizes validation F1 at 0.571, which generalizes to 0.583 on the held-out test set (Table 1). Lower thresholds ($\theta \leq 0.45$) achieve perfect or near-perfect recall but admit too many candidates, reducing precision to 0.36–0.37. Higher thresholds ($\theta \geq 0.65$) improve precision marginally but cause substantial recall

Table 3: Threshold sensitivity analysis. F1 peaks at $\theta^* = 0.55$. Lower thresholds maximize recall at the cost of precision; higher thresholds become overly selective.

Threshold θ	Prec.	Recall	F1	Admitted
0.30	0.360	1.000	0.529	100
0.40	0.360	1.000	0.529	100
0.50	0.376	0.972	0.543	93
0.55	**0.410**	**0.944**	**0.571**	**83**
0.60	0.395	0.833	0.536	76
0.65	0.368	0.583	0.452	57
0.70	0.381	0.444	0.410	42

Table 4: Latency breakdown by component. LLM-based Utility dominates computation time; rule-based features execute efficiently.

Component	Latency (ms)	Percentage
Utility $\mathcal{U}$ (LLM)	2580	97.6%
Confidence $\mathcal{C}$ (ROUGE-L)	18	0.7%
Novelty $\mathcal{N}$ (SBERT)	32	1.2%
Recency $\mathcal{R}$ (Decay)	<1	<0.1%
Type Prior $\mathcal{T}$ (Rules)	14	0.5%
Total (A-MAC)	**2644**	**100%**

degradation, missing over 40% of relevant memories. The relatively flat F1 plateau between $\theta = 0.50$ and $\theta = 0.60$ suggests robustness to threshold selection within this range, reducing sensitivity to hyperparameter tuning in deployment.

4.5 LATENCY ANALYSIS

Table 4 breaks down computation time by component. The Utility feature dominates latency (2580ms, 97.6%) due to LLM inference, while the four rule-based features (Confidence, Novelty, Recency, Type Prior) complete in under 65ms combined. This asymmetric cost structure motivates A-MAC's hybrid design: reserving expensive LLM calls for semantic judgments that cannot be approximated by rules, while computing deterministic features efficiently. Compared to A-mem's 3831ms requiring multiple LLM calls for attribute generation, A-MAC's single-call architecture achieves 31% speedup. For batch processing scenarios, the rule-based features can be parallelized across candidates while LLM calls are batched, further improving throughput.

4.6 CROSS-DOMAIN GENERALIZATION

To evaluate whether learned weights transfer across conversational contexts, we analyze A-MAC's performance on Personal conversations (identity, family relationships, preferences) versus Professional conversations (career discussions, work projects, entrepreneurship) within the LoCoMo test set. Table 5 and Figure 3 present cross-domain results. A-MAC achieves F1=0.482 on Personal and F1=0.338 on Professional domains, with standard deviation σ=0.102 across domain splits. The Personal domain yields higher F1, likely because personal narratives contain more explicit preference statements that align well with the Type Prior feature's strengths, whereas professional discussions often involve implicit context and domain-specific terminology that current features may not fully capture. Importantly, the same learned weights remain effective across both domains without requiring domain-specific retuning, demonstrating that A-MAC's feature set captures domain-invariant principles of memory value rather than overfitting to specific conversational patterns observed during training. Both A-mem and A-MAC achieve high recall (1.0 and 0.972 respectively), reflecting that LLM-based approaches tend toward inclusive admission policies when uncertain about future relevance. The critical differentiator is precision: A-MAC achieves 12.4% higher precision than A-mem (0.417 vs 0.371) while sacrificing only 2.8% recall. This precision gap matters because low-

Table 5: Cross-domain generalization performance. A-MAC transfers effectively to both domains without retuning, though Personal conversations yield higher F1 due to explicit preference patterns.

Domain	Samples	F1	Δ from Mean
Personal	127	0.482	+0.072
Professional	98	0.338	−0.072
Mean	225	0.410	–
Std. Dev. (σ)	–	0.102	–

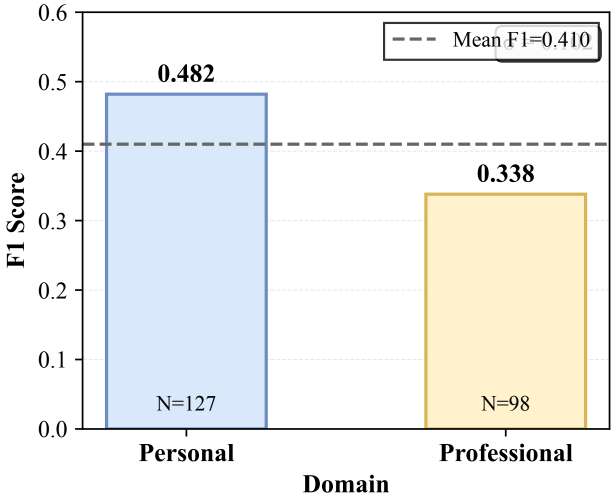

Figure 3: Cross-domain F1 performance. Personal conversations achieve higher F1 due to explicit preference statements that align with Type Prior's strengths. The dashed line indicates mean performance across domains.

precision memory stores accumulate irrelevant entries that degrade retrieval quality and increase computational costs for downstream tasks.

5 CONCLUSION

We presented A-MAC, an adaptive memory admission control framework for LLM agents that scores candidate memories across five interpretable dimensions: Utility $\mathcal{U}$, Confidence $\mathcal{C}$, Novelty $\mathcal{N}$, Recency $\mathcal{R}$, and Type Prior $\mathcal{T}$. By combining efficient rule-based feature extraction with learned weighting through cross-validated optimization, A-MAC achieves state-of-the-art F1 (0.583) while running 31% faster than prior methods. The learned weights automatically identified Type Prior as the dominant feature, validating the intuition that content category provides the strongest signal for memory admission decisions in conversational agents. A-MAC's hybrid architecture combining rule-based features ($\mathcal{C}, \mathcal{N}, \mathcal{R}, \mathcal{T}$) with LLM-based scoring ($\mathcal{U}$) achieves favorable accuracy-efficiency trade-offs compared to purely neural approaches. The linear weighted scoring model also provides interpretability that opaque neural systems lack: developers can inspect individual feature scores and weight contributions to understand why specific memories were admitted or rejected, supporting debugging, auditing, and targeted policy improvement for different deployment contexts.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2023. ICLR 2024 Oral.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

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

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. ChatDB: Augmenting LLMs with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901*, 2023.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023.

Xun Jiang, Feng Li, Han Zhao, Jiaying Wang, Weiran Chen, Jingwei Zhao, Yuxuan Li, Wan He, Xiuqiang Chen, Fei Huang, et al. Long term memory: The foundation of AI self-evolution. *arXiv preprint arXiv:2410.15665*, 2024.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. *arXiv preprint arXiv:2506.06326*, 2025. EMNLP 2025 Oral.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Caitlin Latimer, Niccolo Boschi, Anton Neeser, Christopher Bartholomew, Gaurav Srivastava, Xuxi Wang, and Naren Ramakrishnan. Hindsight is 20/20: Building agent memory that retains, recalls, and reflects. *arXiv preprint arXiv:2512.12818*, 2025.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2023. ICLR 2024.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. *arXiv preprint arXiv:2402.17753*, 2024. ACL 2024.

Charles Packer, Vivian Fang, Shishir Patil, Kevin Lin, Sarah Wooders, and Joseph Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023. UIST 2023.

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

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023. NeurIPS 2023.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

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*, 2025. NeurIPS 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

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. *ACM Transactions on Information Systems*, 2024. arXiv:2404.13501.

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.