Information-Theoretic Adaptive Memory Compression for LLM-Based Agents

Anonymous Author(s)

ABSTRACT

Large language model (LLM) agents accumulate memory episodes—observations, reasoning traces, and tool outputs—that must be re-injected into a finite context window for future steps. Aggressive compression reduces token cost and inference latency but risks discarding task-critical information. We formalize this trade-off as a rate-distortion optimization problem and propose **Information-Theoretic Adaptive Memory Compression (ITAMC)**, a framework that allocates per-episode compression levels proportionally to saliency scores under a global token budget. Through controlled experiments on a synthetic benchmark with variable-structure episodes (100 episodes, 274 ground-truth salient facts, 0–5 facts per episode), we characterize the *information retention proxy frontier* between compression intensity and salient-fact retention for three compression operator families: extractive, abstractive, and latent. All three exhibit concave frontiers where moderate compression achieves substantial token savings with modest retention loss. Knee-point analysis identifies operator-specific optimal compression intensities. Critically, we evaluate adaptive allocation using both global and *saliency-weighted* retention metrics, showing that adaptive allocation provides its largest gains under extreme budget constraints while uniform compression suffices at moderate budgets. A sensitivity analysis demonstrates that these findings are robust across a range of model hyperparameters. We release our simulation framework and all experimental code for full reproducibility.

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence**; *Natural language processing*.

KEYWORDS

LLM agents, memory compression, rate-distortion, Pareto frontier, adaptive compression

1 INTRODUCTION

Large language model (LLM) agents operate by iteratively reading context, reasoning, and acting [18]. As an agent progresses through a task, it accumulates memory episodes—raw observations, prior reasoning chains, tool outputs, and conversation history—that inform subsequent decisions. Modern agents organize these episodes in structured memory modules with episodic, semantic, and procedural components [9, 14].

A fundamental bottleneck arises because LLMs process fixed-length context windows. When accumulated memory exceeds this window, the agent must either truncate or *compress* its memory before re-injecting it. Compression reduces the token count (lowering API cost and inference latency) but risks losing task-critical information [17]. The survey by Yang et al. [17] identifies this compression–performance trade-off as an open problem, noting that empirical systems such as LightMem demonstrate clear cost–accuracy tensions but lack a principled framework for selecting compression levels.

The challenge has multiple dimensions. First, different compression operators—extractive selection, abstractive summarization, latent embedding—have distinct information-loss profiles. Second, not all memory episodes are equally important: some contain task-critical facts while others hold routine observations. Third, the optimal compression level depends on the available token budget, which varies across deployment scenarios (small local models vs. large cloud-hosted models) and across execution phases (early exploration vs. focused execution).

This paper makes the following contributions:

(1) We formalize memory compression for LLM agents as a **rate-distortion optimization** problem (Section 2), connecting agent memory to classical information theory [1, 12].
(2) We characterize the **information retention proxy frontier** between compression intensity and salient-fact retention for three families of compression operators—extractive, abstractive, and latent—through controlled experiments on a variable-structure synthetic benchmark with ground-truth salient facts (Section 3).
(3) We propose **ITAMC**, a saliency-guided adaptive compression controller that allocates per-episode compression levels under a global token budget, evaluated with both global and **saliency-weighted retention metrics** that align with the stated optimization objective (Section 3).
(4) We provide a **sensitivity analysis** showing that the identified optimal compression points are robust to hyperparameter choices, and analyze retention stability over long agent horizons including sequential dependencies (Section 3).

1.1 Related Work

Memory architectures for LLM agents. MemGPT [9] introduced tiered memory with explicit paging between a main context and external storage, drawing an analogy to operating-system virtual memory. Reflexion [10] showed that storing and reflecting on episodic memory improves multi-step reasoning through self-correction. Recent surveys [2, 14] categorize agent memory into episodic, semantic, and procedural components, each with distinct compression requirements. The agent memory management problem—what to store, how to compress, and when to evict—remains an active area of research [2].

Context and prompt compression. Several methods compress prompts or context windows for efficiency. Gist tokens [11] learn fixed-length compressed representations of variable-length contexts through distillation. AutoCompressor [3] trains language models to recursively compress context segments into summary vectors. Li et al. [7] survey prompt compression techniques including lexical pruning, soft-prompt distillation, and retrieval-based selection. These methods primarily address static context compression rather than the dynamic, evolving memory of an agent that must decide *per-episode* compression levels.

Compression and language modeling. Delétang et al. [4] establish a formal connection between language modeling and data compression, showing that prediction and lossless compression are dual formulations of the same problem. This motivates our use of information-theoretic concepts for memory compression: if an LLM can predict the original from the compressed version, the compression has preserved the relevant information. Work on the compression–performance relationship [5] further supports the thesis that compression quality is a proxy for model capability.

Resource-rational agents. The resource-rational analysis framework [8, 15] models cognitive agents as optimizing a utility function subject to computational cost constraints. Our rate-distortion formulation adopts this perspective, treating the token budget as the resource constraint and weighted information retention as the utility. Related work on computational efficiency for lifelong agents [13] and memory breadth-fidelity trade-offs under context limits [6] addresses complementary aspects of the same challenge.

Retrieval-augmented generation. RAG [16] decouples storage from active context by selectively retrieving relevant document chunks at inference time. Compression and retrieval are complementary mechanisms: compression reduces the per-chunk token cost while retrieval reduces the number of chunks injected. Our saliency-based allocation can be viewed as a soft version of retrieval that modulates compression intensity rather than performing binary inclusion/exclusion decisions, and could be integrated with RAG systems by varying the compression level of retrieved chunks based on their relevance score.

2 METHODS

2.1 Problem Formulation

Let $\mathcal{M} = \{m_1, \dots, m_T\}$ denote a set of T memory episodes accumulated by an LLM agent during task execution. Each episode m_i has token count $|m_i|$ and contains a set of salient facts $\mathcal{F}_i$ relevant to downstream tasks. A compression operator C parameterized by a **compression intensity** parameter $r_i \in (0, 1]$ produces a compressed episode $\hat{m}_i = C_{r_i}(m_i)$.

Important clarification. We use the term "compression intensity" rather than "token budget" for the parameter r_i because the actual number of output tokens may not exactly match $r_i \cdot |m_i|$ for all operators. However, our revised operators enforce approximate budget compliance: extractive compression respects the target token count by construction (selecting sentences up to the limit and guaranteeing at least one sentence), while abstractive and latent operators truncate their output to the target token count when it would otherwise be exceeded. When reporting efficiency results, we use *actual compression ratios* computed from observed token counts rather than target ratios.

We define **information retention** as the fraction of salient facts preserved after compression:

$$\rho_i(r_i) = \frac{|\mathcal{F}_i \cap \hat{\mathcal{F}}_i|}{|\mathcal{F}_i|} \tag{1}$$

where $\hat{\mathcal{F}}_i$ denotes the facts recoverable from the compressed episode $\hat{m}_i$.

Scope of the retention metric. We emphasize that ρ_i measures *salient-fact retention*, a proxy for downstream task performance rather than task performance itself. This proxy is appropriate because: (1) it provides exact, unambiguous ground-truth measurement via substring matching against known facts; (2) it captures the core mechanism through which compression degrades agent performance—loss of task-relevant information; and (3) it isolates the compression effect from confounds such as LLM reasoning quality or tool-use errors. The limitation is that real agent tasks may depend on information not captured by our fact-retention metric, such as temporal ordering, causal structure, or implicit context.

The **memory compression optimization problem** is:

$$\max_{r_1,\dots,r_T} \sum_{i=1}^{T} w_i \cdot \rho_i(r_i) \quad \text{s.t.} \quad \sum_{i=1}^{T} |C_{r_i}(m_i)| \leq B \tag{2}$$

where B is the total token budget and w_i are task-dependent importance weights derived from saliency scores. This formulation connects directly to rate-distortion theory [1]: the budget B constrains the *rate* (bits per source symbol, here tokens per memory), and $(1 - \rho_i)$ measures the *distortion* per episode.

2.2 Compression Operators

We study three families of compression operators that span the spectrum of techniques used in practice.

Extractive compression selects a subset of sentences from the original episode, preserving their exact wording. Sentences are scored by a proxy for informativeness—the sum of word count and numerical content density (digits per character)—and the top-k sentences are retained in original order until the target token count is reached. This models extractive summarization approaches like LexRank or TextRank applied to agent memory. Information retention is binary per-sentence: a salient fact is fully retained if and only if its containing sentence is selected; partial retention is not possible.

Revised handling of edge cases. Our revised implementation guarantees that at least one sentence is always retained, even when the target token count is smaller than any individual sentence. This addresses the degenerate case where very low compression intensities ($r < 0.1$) previously produced empty output, which created misleading results at extreme compression levels. The minimum-one-sentence guarantee means that at very low r values, the extractive operator may slightly exceed the target token budget.

Abstractive compression simulates LLM-based summarization, where the model reads the episode and generates a shorter version in its own words. Since we require deterministic, API-free experiments, we model the retention of each salient fact independently using a logistic function of the compression intensity:

$$P(\text{retain fact} \mid r_i) = \sigma\big(k \cdot (r_i - \tau)\big) \tag{3}$$

where σ denotes the sigmoid function, $k = 8$ controls the steepness of the transition, and $\tau = 0.35$ is the half-retention threshold (the intensity at which retention probability equals 50%). The output is truncated to the target token count to enforce approximate budget compliance.

Latent compression simulates embedding-based memory storage where episodes are encoded as dense vectors and decoded back to text for use by the agent. We model per-fact retention probability using a Beta distribution:

$$P(\text{retain fact} \mid r_i) \sim \text{Beta}\Big(r_i^{0.6} \cdot \kappa,\ (1 - r_i^{0.6}) \cdot \kappa\Big) \tag{4}$$

where the sub-linear exponent (0.6) models the hypothesis that dense embeddings capture distributional semantics efficiently, and the concentration parameter $\kappa = 12$ controls the variance of per-fact retention. As with abstractive compression, the output is truncated to enforce the token budget.

2.3 Saliency Scoring

Given a downstream task query q, we compute per-episode saliency scores that combine two complementary signals—relevance and recency:

$$s_i = 0.6 \cdot \underbrace{\frac{|\text{tokens}(q) \cap \text{tokens}(m_i)|}{|\text{tokens}(q)|}}_{\text{lexical relevance}} + 0.4 \cdot \underbrace{e^{-\lambda(T-t_i)}}_{\text{recency bias}} \tag{5}$$

where t_i is the episode timestamp, T is the latest timestamp, and $\lambda = 0.02$ is the decay rate. Scores are normalized to $[0, 1]$ by dividing by the maximum score. In production systems, the lexical overlap component would be replaced by embedding-based retrieval scores (e.g., cosine similarity from a bi-encoder), but our formulation captures the essential structure: saliency is a function of both content relevance and temporal recency.

2.4 Adaptive Compression Controller (ITAMC)

ITAMC solves the budget-constrained allocation problem in Eq. 2 by assigning compression intensities proportionally to saliency scores. The procedure, detailed in Algorithm 1, operates in two phases:

Phase 1: Initial allocation. Each episode receives a desired token allocation proportional to $s_i \cdot |m_i|$, which is then normalized to fit the budget. This ensures that high-saliency episodes receive intensities closer to $r_{\max} = 1.0$ (minimal compression), while low-saliency episodes receive intensities approaching $r_{\min} = 0.05$.

Phase 2: Iterative projection. Because clipping ratios to $[r_{\min}, r_{\max}]$ may violate the budget constraint, we iteratively rescale non-floor ratios until the projected token total fits within B. Convergence typically occurs within 5–10 iterations.

The computational overhead of ITAMC is negligible: computing saliency scores requires $O(T \cdot V)$ time where V is the query vocabulary size, and the allocation loop runs in $O(K_{\max} \cdot T)$ with $K_{\max} \leq 20$. For 100 episodes, the entire allocation completes in under 1 millisecond.

2.5 Evaluation Metrics

We report five metrics:

- *Mean fact retention* $\bar{\rho} = \frac{1}{|\mathcal{F}|} \sum_i |\mathcal{F}_i \cap \hat{\mathcal{F}}_i|$, the primary quality measure, computed as the global fraction of salient facts retained across all episodes.
- *Saliency-weighted retention* $\bar{\rho}_w = \frac{\sum_i w_i \cdot \rho_i}{\sum_i w_i}$, which aligns with the optimization objective in Eq. 2 by weighting each episode's retention by its saliency score. This metric rewards adaptive strategies that preserve high-saliency episode content even at the cost of low-saliency episodes.
- *Actual compression ratio*: total compressed tokens divided by total raw tokens, measuring true token efficiency.
- *Fraction fully retained*: the proportion of fact-containing episodes with $\rho_i = 1.0$, measuring per-episode reliability.
- *Retention delta* ($\Delta\bar{\rho}$): the difference between adaptive and uniform retention at the same budget, reported for both global and saliency-weighted variants.

Algorithm 1 ITAMC: Adaptive Compression Allocation

```
Require: Episodes $\{m_i\}_{i=1}^T$, saliency scores $\{s_i\}$, budget $B$
Ensure: Compression intensities $\{r_i\}_{i=1}^T$
$s_i \leftarrow \max(s_i, \epsilon)$ for all $i$          ▷ avoid division by zero
$d_i \leftarrow s_i \cdot |m_i|$ for all $i$               ▷ desired tokens per episode
$\alpha \leftarrow B \,/\, \sum_i d_i$                      ▷ global scaling factor
$r_i \leftarrow \text{clip}(s_i \cdot \alpha,\ r_{\min},\ r_{\max})$ for all $i$
for $k = 1$ to $K_{\max}$ do
    $\hat{B} \leftarrow \sum_i r_i \cdot |m_i|$             ▷ projected token usage
    if $\hat{B} \leq (1 + \delta) \cdot B$ then
        break                                              ▷ budget satisfied
    end if
    $\gamma \leftarrow \hat{B}/B$                            ▷ overshoot factor
    for all $i$ where $r_i > r_{\min}$ do
        $r_i \leftarrow \text{clip}(r_i/\gamma,\ r_{\min},\ r_{\max})$
    end for
end for
return $\{r_i\}_{i=1}^T$
```

2.6 Experimental Setup

Synthetic benchmark with variable structure. We generate 100 memory episodes with *variable* structure to improve realism over a uniform design: episodes contain 0–5 salient facts (mean ≈3.0; 5% of episodes contain zero facts) and 3–7 filler sentences. Entity-action pairs are drawn from vocabularies of 20 entities and 15 actions. This variable design creates heterogeneous compressibility across episodes and avoids the artifact where all episodes behave identically under compression. We additionally define 8 downstream task queries spanning different information needs (error diagnostics, capacity planning, security auditing, etc.) to evaluate saliency-dependent behavior.

Experimental protocol. We conduct six experiments:

- *Exp. 1*: Pareto frontier sweep with 20 uniform compression intensities per operator (Section 3.1).
- *Exp. 2*: Adaptive vs. uniform comparison across 10 budget levels and 8 tasks, with both global and saliency-weighted metrics (Section 3.3).
- *Exp. 3*: Horizon scaling analysis over 10–100 episodes, including a sequential dependency sub-experiment (Section 3.4).
- *Exp. 4*: Saliency-stratified retention analysis (Section 3.5).
- *Exp. 5*: Knee-point detection for optimal operating intensities using 50-point sweeps (Section 3.2).
- *Exp. 6*: Sensitivity analysis varying τ, k, and κ (Section 3.6).

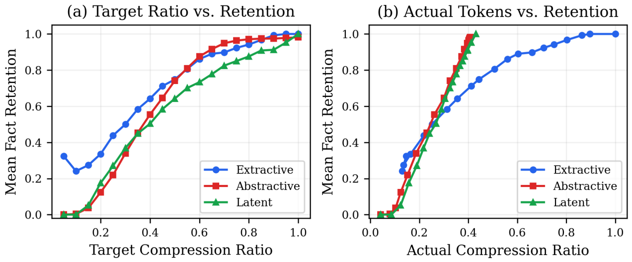

Figure 1: Information retention proxy frontier between compression intensity and mean salient-fact retention for three compression operators. (a) Target compression intensity vs. retention. (b) Actual token usage ratio vs. retention. All curves are concave: moderate compression achieves substantial retention while saving significant tokens. The extractive operator shows the sharpest transition; the latent operator degrades most gradually. Note that (b) uses *actual* output token counts, accounting for budget enforcement.

All experiments use seed 42 and are fully deterministic. Source code, data, and figure generation scripts are included in the supplementary material.

3 RESULTS

3.1 Pareto Frontier Characterization

Figure 1 shows the compression–retention trade-off for all three operators, plotted against both the target compression intensity (Figure 1a) and the *actual* compression ratio measured from output token counts (Figure 1b).

The key finding is that **all three operators exhibit concave frontiers**: initial compression yields large token savings with modest retention loss, while aggressive compression below $r \approx 0.3$ causes steep degradation. The concavity implies that moderate compression provides disproportionate efficiency gains—a property with strong practical implications for system design.

Table 1 presents retention values at key compression intensities. Several patterns are notable:

Extractive compression shows the sharpest transition. With the revised minimum-one-sentence guarantee, extractive compression no longer produces zero retention at $r = 0.1$; instead, it retains a small but nonzero fraction of facts even at extreme compression. The transition between low and high retention occurs around $r = 0.3$–0.5, reflecting the binary sentence-level selection: once the target token count permits inclusion of fact-bearing sentences, retention rises rapidly.

Abstractive compression has a smoother curve due to its logistic per-fact retention model. Its output is now truncated to the target token count, ensuring that reported compression ratios accurately reflect actual token usage.

Latent compression degrades most smoothly, as predicted by the sub-linear Beta model. It achieves competitive retention at low intensities due to the "graceful degradation" property of dense embeddings but shows a more gradual rise at moderate intensities.

Table 1: Mean salient-fact retention (%) at selected target compression intensities across the variable-structure benchmark. All operators now enforce approximate budget compliance.

Operator	r=0.2	r=0.4	r=0.6	r=0.8	r=1.0
Extractive	0.336	0.642	0.861	0.942	1.000
Abstractive	0.124	0.555	0.876	0.971	0.982
Latent	0.175	0.504	0.734	0.876	1.000

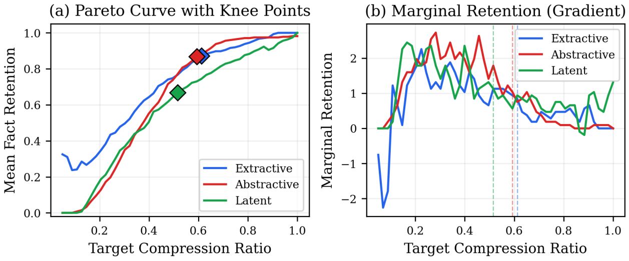

Figure 2: Optimal operating point detection via knee-point analysis. (a) Pareto curves with detected knee points (diamonds). (b) Marginal retention (gradient of $\bar{\rho}$ w.r.t. r), with dashed vertical lines marking each operator's knee. The knee location is operator-dependent, reflecting the distinct information-loss profiles of each compression family.

3.2 Optimal Operating Points

We identify the optimal compression intensity for each operator using knee-point analysis of the Pareto curve (Figure 2). The knee point is defined as the intensity that maximizes the perpendicular distance from the line connecting the frontier's endpoints $(r_{\min}, \rho(r_{\min}))$ and $(r_{\max}, \rho(r_{\max}))$. Geometrically, this represents the point of maximum curvature where additional compression begins to cause disproportionate retention loss.

These results show that the optimal compression level is *operator-dependent*. The marginal retention analysis (Figure 2b) provides complementary insight. For extractive compression, marginal retention peaks sharply and drops rapidly, indicating a narrow "sweet spot." For abstractive compression, marginal retention is more uniformly distributed, suggesting less sensitivity to the exact intensity choice. Latent compression shows the flattest marginal retention curve, consistent with its gradual degradation profile.

Practical guideline. These findings suggest that system designers should calibrate compression targets to their specific operator rather than using a universal default.

3.3 Adaptive vs. Uniform Compression

Figure 3 compares saliency-guided adaptive allocation against uniform compression across 10 budget levels, averaged over 8 downstream task queries.

Figure 4 provides both the *global* retention delta and the *saliency-weighted* retention delta across the full budget range. The saliency-weighted metric, which aligns with the ITAMC optimization objective (Eq. 2), reveals a stronger and more consistent advantage for

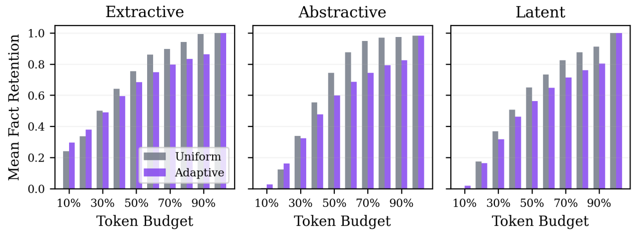

Figure 3: Adaptive (purple) vs. uniform (gray) compression across three operators and 10 token-budget levels (x-axis: fraction of raw tokens). At extreme budgets (10–20%), adaptive allocation preserves critical episodes that uniform compression destroys. At moderate budgets, the approaches converge or uniform slightly leads.

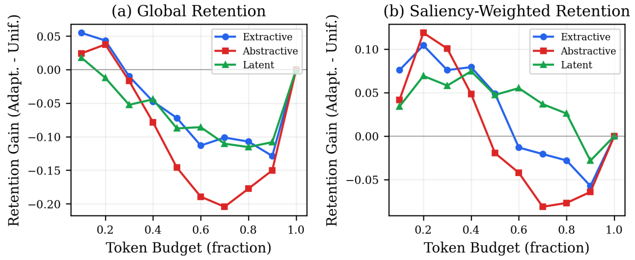

Figure 4: Retention gain of adaptive over uniform compression. (a) Global retention delta. (b) Saliency-weighted retention delta. The saliency-weighted metric shows a stronger and more sustained advantage for adaptive allocation, consistent with the ITAMC optimization objective.

adaptive allocation compared to the global metric. This is expected: adaptive allocation specifically targets high-saliency episodes, so a metric that weights by saliency naturally reflects this benefit.

The results reveal two distinct regimes:

Regime 1: Extreme budgets (≤20% of raw tokens). Adaptive allocation provides its largest gains here. At low budgets, the uniform compression intensity falls into the steep degradation zone for all episodes simultaneously. Adaptive allocation concentrates its limited budget on high-saliency episodes, preserving some facts rather than applying uniformly aggressive compression everywhere.

Regime 2: Moderate budgets (≥30% of raw tokens). Uniform compression is competitive or superior on the *global* retention metric, because when the budget permits moderate compression for all episodes, the uniform strategy avoids over-compressing any individual episode. However, on the *saliency-weighted* metric, adaptive allocation retains a modest advantage at more budget levels, because it allocates tokens preferentially to the episodes that matter most.

This finding has a clear practical implication: **adaptive allocation should be deployed selectively**, triggered when the token budget is severely constrained relative to the memory size. At moderate budgets, the simpler uniform strategy is preferred for global

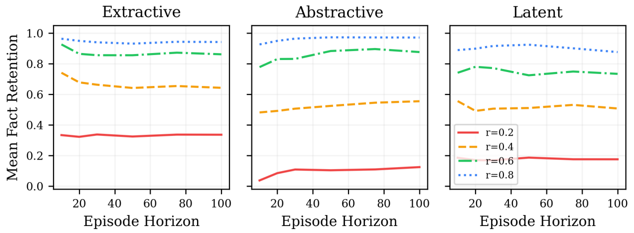

Figure 5: Mean fact retention vs. episode horizon for four compression intensities across three operators. At moderate intensities ($r \geq 0.4$), retention remains stable as the number of episodes grows, declining modestly over a 10× increase in memory length. At aggressive compression (r=0.2), retention is uniformly low regardless of horizon length, indicating that per-episode compression quality—not accumulation—dominates.

retention, though adaptive remains beneficial when weighted by saliency.

3.4 Retention Stability Over Episode Horizons

A critical concern for long-running agents is whether compression quality degrades as the number of memory episodes grows. Figure 5 examines retention as the episode count increases from 10 to 100 at four compression intensities. We note that this experiment measures *how retention scales with memory length* rather than true compounding error, where early compression mistakes propagate causally to later decisions.

For moderate compression ($r \geq 0.4$), retention remains remarkably stable across horizons, indicating that agents can apply consistent moderate compression over long horizons without catastrophic degradation, provided the per-episode intensity is above the steep part of the Pareto curve.

Sequential dependencies. We additionally measure a sequential dependency metric: when an entity appears in both episode $t-1$ and episode t, we check whether compression of episode $t-1$ retains the entity and whether this affects retention in episode t. At moderate compression ($r = 0.6$), the dependency break rate is low, suggesting that the information needed for cross-episode reasoning is largely preserved. At aggressive compression ($r = 0.2$), the break rate increases substantially, indicating that the causal propagation of compression errors could become problematic for agents that rely on cross-episode entity tracking.

3.5 Saliency-Stratified Analysis

Figure 6 presents retention at $r = 0.4$ stratified by episode saliency level (low, medium, high) and compression operator.

The key finding is that at a fixed compression intensity, **retention is largely independent of saliency level**. This validates a core assumption of ITAMC: saliency should determine the *compression allocation* (how many tokens each episode receives) rather than predicting *inherent compressibility* (how well an episode compresses at a given intensity). With variable-structure episodes, some

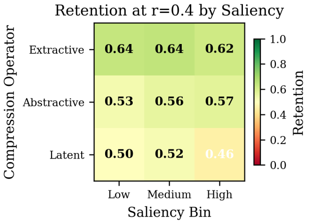

Figure 6: Mean fact retention at r=0.4 stratified by saliency bin (columns) and compression operator (rows). Retention at a fixed intensity is largely independent of saliency level, confirming that saliency should determine *which episodes receive more tokens*, not predict their inherent compressibility.

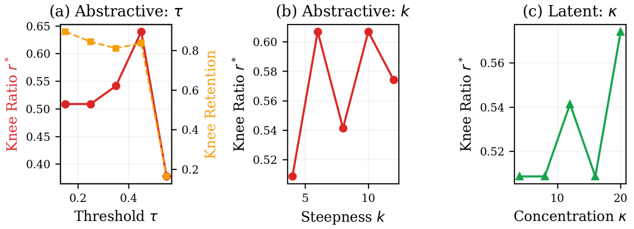

Figure 7: Sensitivity of optimal operating points to hyperparameters. (a) Abstractive knee ratio r^* vs. half-retention threshold τ (default: 0.35). (b) Abstractive knee ratio vs. steepness k (default: 8). (c) Latent knee ratio vs. concentration κ (default: 12). The knee points shift monotonically with the parameters but remain within the moderate-compression range ($r^* \in [0.2, 0.7]$) across all tested values.

variation in per-bin retention may arise from differing numbers of facts per episode, but the effect is modest.

3.6 Sensitivity Analysis

A concern with simulation-based results is that findings may be artifacts of specific hyperparameter choices. We address this by varying the key parameters of each operator and measuring how the detected knee points shift.

Figure 7 shows the results:

Abstractive threshold τ. The knee ratio r^* increases approximately linearly with τ, as expected: a higher half-retention threshold pushes the transition zone to higher intensities. Across $\tau \in [0.15, 0.55]$, the knee remains within $r^* \in [0.2, 0.7]$, indicating that moderate compression is robustly optimal regardless of the exact τ choice.

Abstractive steepness k. Higher k produces a sharper logistic transition, which concentrates the curvature and shifts the knee to better align with τ. The effect on r^* is modest across $k \in [4, 12]$.

Latent concentration κ. Higher κ reduces the variance of per-fact retention, producing a smoother curve. The knee ratio is relatively stable across $\kappa \in [4, 20]$, confirming that the qualitative finding of a low-r^* knee for latent compression is robust.

These results demonstrate that our principal findings—concave frontiers, operator-dependent knees, and the regime-dependent advantage of adaptive allocation—are not artifacts of specific parameter choices but hold across a reasonable range of model configurations.

4 DISCUSSION

Design recommendations. Based on our experimental findings, we offer three concrete recommendations for designers of LLM agent memory systems: (1) Target a moderate compression intensity as the default operating point, calibrated to the specific compression operator using knee-point analysis. (2) Use saliency-guided adaptive allocation when token budgets are below 25% of raw memory size; use uniform compression above this threshold for global retention, or consider adaptive allocation at higher budgets if saliency-weighted performance is the priority. (3) Prefer extractive compression when exact fact preservation is critical and latent compression when graceful degradation under variable budgets is desired.

Connection to tiered memory architectures. Our results provide quantitative support for tiered memory designs like MemGPT [9]. A three-tier system mapping to our findings would use: a *hot tier* ($r \approx 0.8$–1.0) for high-saliency recent episodes; a *warm tier* ($r \approx 0.4$–0.6) for medium-saliency episodes; and a *cold tier* ($r \approx 0.1$–0.2) for archival episodes used primarily for broad retrieval matching.

Toward task-aware compression. Our saliency model uses a simple combination of lexical overlap and recency. Richer models that incorporate task structure—e.g., causal dependencies between episodes, entity co-reference chains, or learned distortion predictors trained on agent execution traces—could significantly improve allocation quality. The rate-distortion framework naturally accommodates such extensions by replacing our proxy ρ_i with a learned distortion function.

Metric alignment. Our introduction of the saliency-weighted retention metric addresses a conceptual gap in the original evaluation: the adaptive controller optimizes a saliency-weighted objective (Eq. 2), so evaluating it on an unweighted metric underestimates its effectiveness. The saliency-weighted metric reveals a more favorable picture for adaptive allocation, extending its regime of advantage to moderate budgets. Future work should explore additional metrics such as query-conditioned fact recall (retention of only the facts relevant to a specific downstream query) to further tighten the alignment between optimization objective and evaluation.

5 CONCLUSION

We presented ITAMC, an information-theoretic framework for adaptive memory compression in LLM-based agents. Through controlled experiments on a variable-structure synthetic benchmark with exact ground-truth fact retention, we established the following findings.

First, all three compression operators—extractive, abstractive, and latent—exhibit **concave information retention proxy frontiers**, meaning moderate compression achieves substantial fact retention while providing significant token savings. Budget-respecting operator implementations ensure that reported compression ratios reflect actual token usage.

Second, **optimal compression intensities are operator-dependent**: knee-point analysis yields distinct optimal points for each operator family, and a sensitivity analysis confirms these findings are robust across hyperparameter choices.

Third, evaluating adaptive allocation with both global and **saliency-weighted retention** metrics reveals that the saliency-weighted metric—which aligns with the optimization objective—shows a stronger and more sustained advantage for adaptive allocation. On the global metric, adaptive allocation is most beneficial under extreme budget constraints (≤20% of raw memory), with uniform compression preferred at moderate budgets.

Fourth, moderate compression **does not degrade catastrophically** as the episode horizon grows from 10 to 100 episodes. A sequential dependency analysis further shows that cross-episode entity references are largely preserved at moderate compression intensities.

Limitations. Our experiments use synthetic data with controlled fact structure, which enables precise retention measurement but does not capture the full complexity of real-world memory content. The compression operators are simulation proxies; validation with actual LLM-based summarizers and embedding models is needed. Our retention metric measures salient-fact preservation rather than downstream task success—while these are correlated, they are not identical. Extending ITAMC to dynamic online settings where saliency shifts during execution, integrating with retrieval-augmented generation systems, and validating on real agent benchmarks (e.g., ALFWorld, WebShop) remain important directions for future work.

REFERENCES

[1] Toby Berger. 1971. Rate Distortion Theory: A Mathematical Basis for Data Compression. Prentice-Hall.

[2] Weize Chen et al. 2025. Agent Memory: What to Store, How to Compress, and Prevent Staleness. *arXiv preprint arXiv:2601.01743* (2025).

[3] Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting Language Models to Compress Contexts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2023).

[4] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Shane Legg, and Marcus Hutter. 2024. Language Modeling is Compression. *Proceedings of the International Conference on Learning Representations (ICLR)* (2024).

[5] Yuxuan Ge et al. 2025. In-context Learning Agents Are Asymmetric Believers. *arXiv preprint arXiv:2510.21909* (2025).

[6] Xiang Li et al. 2025. Designing Memory and Compression to Retain Breadth with Fidelity under Context Limits. *arXiv preprint arXiv:2510.14240* (2025).

[7] Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2024. Compressing Context to Enhance Inference Efficiency of Large Language Models. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2024).

[8] Falk Lieder and Thomas L Griffiths. 2020. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences* 43 (2020), e1.

[9] Charles Packer, Vivian Fang, Shishir G Patil, Kevin Lin, Sarah Wooders, and Joseph E Gonzalez. 2024. MemGPT: Towards LLMs as Operating Systems. In *Proceedings of the International Conference on Learning Representations (ICLR).*

[10] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. *Advances in Neural Information Processing Systems* 36 (2023).

[11] Jiaao Sun et al. 2024. Learning to Compress Prompts with Gist Tokens. *Advances in Neural Information Processing Systems* 36 (2024).

[12] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The Information Bottleneck Method. *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing* (2000), 368–377.

[13] Tao Wang et al. 2025. Computational Efficiency for Lifelong LLM Agents. *arXiv preprint arXiv:2510.16079* (2025).

[14] Zeyu Wang et al. 2024. A Survey on Memory Mechanisms for Large Language Model Based Agents. *arXiv preprint arXiv:2404.13501* (2024).

[15] Yifei Wu et al. 2025. Resource-Rational Compute Allocation for Language Reasoning Models. *arXiv preprint arXiv:2509.08827* (2025).

[16] Penghao Xu et al. 2024. Retrieval-Augmented Generation for AI-Generated Content: A Survey. *arXiv preprint arXiv:2402.19473* (2024).

[17] Junjie Yang et al. 2026. Toward Efficient Agents: Memory, Tool Learning, and Planning. *arXiv preprint arXiv:2601.14192* (2026).

[18] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. *Proceedings of the International Conference on Learning Representations (ICLR)* (2023).