REMEMBER WHAT MATTERS: RELEVANCE-BANDED COMPRESSION FOR LONG-HORIZON LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

While long-horizon large language model (LLM) agents show immense promise in complex environments, long-horizon execution inevitably leads to a continuously growing interaction history, often triggering context explosion and performance degradation. To mitigate this issue, existing context compression methods typically compress historical context using uniform, blanket strategies. However, such indiscriminate operations fail to recognize the dynamic value of intermediate steps, risking over-compression of fine-grained, task-critical execution details. To address this limitation, we propose **Re**levance-**Band**ed (**ReBand**) Compression, a training-free framework that uses semantic relevance between historical chunks and the agent's current runtime state to guide differentiated compression. Specifically, ReBand assigns unsummarized chunks to low-, medium-, and high-relevance bands, applying aggressive compression, concise factual summarization, and structured key-value retention, respectively. A single compressor call applies this guidance to update the rolling summary, while the latest interaction remains verbatim. Across three long-horizon task benchmarks, ReBand consistently outperforms ACON-UTCO, improving primary task metrics by 3.16–4.8 points while reducing peak context by up to 42.4%. These results demonstrate that runtime relevance provides a simple, transferable signal for allocating compression fidelity across context chunks.

1 INTRODUCTION

Large language model (LLM) agents have demonstrated remarkable success in various tasks, showing strong proficiency across tool usage, API calling, and structured dialogue (Yao et al., 2023; Schick et al., 2023; Patil et al., 2024). These advances support increasingly complex workflows in web interaction, office automation, and software engineering (Lee et al., 2025; Yang et al., 2024; Wang et al., 2024). As execution proceeds, however, tool outputs, intermediate results, and completed actions accumulate into a growing interaction history, resulting in context explosion. This growing context creates two major challenges for effective context compression: (1) *Rapidly growing context*: unconstrained history accumulation leads to prohibitive token consumption and exposes the agent to overwhelming redundant information that degrades decision quality (Shi et al., 2023); and (2) *Critical information loss in the middle*: in long-horizon tasks, critical information from earlier turns is easily overlooked (Liu et al., 2024), while aggressive truncation or indiscriminate compression leads to loss of essential intermediate information, even for long-context models. Therefore, sustaining long-horizon execution requires controlling context growth while preserving the intermediate information on which later actions depend.

Existing approaches address context growth through token pruning, external memory retrieval, and rolling summarization (Liu et al., 2025; Dong et al., 2026). Specifically, token pruning removes low-utility tokens to stay within budget, but it easily ruptures structural dependencies across multi-turn workflows (Jiang et al., 2023). Moreover, external-memory approaches such as PACE select representations at different levels of detail from a persistent history store, but they introduce substantial query latency and often fail to retrieve when agents do not realize an omitted variable must be recalled (Wei et al., 2026). Consequently, recent long-horizon agents increasingly adopt an irreversible, single rolling summary carried forward across execution turns (Kang et al., 2026; Zhou et al., 2026). However, existing summarization-based methods treat all accumulated historical chunks indiscriminately by enforcing a uniform compression guideline (Li et al., 2025). Such in-

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.

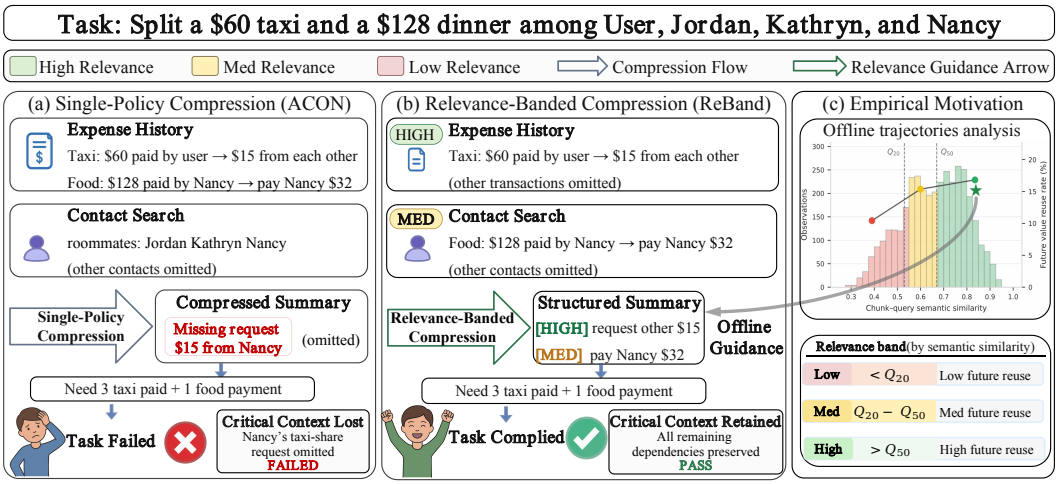


Figure 1: Illustration and empirical motivation for relevance-banded compression. (a) and (b) On the same AppWorld bill-splitting task, ACON omits Nancy's outstanding $15 taxi-share request, whereas ReBand preserves it and completes the task (see Appendix G for the detailed trajectory analysis). (c) Offline analysis of successful AppWorld training trajectories shows that future-value reuse increases across bands of chunk-query semantic similarity.

discriminate compression presents an acute dilemma: it either over-summarizes history and erases task-critical variables needed for downstream execution, or preserves excessive redundant content. Effective compression therefore requires a runtime signal that can estimate the potential future utility of each historical chunk and accordingly determine its compression fidelity. A natural candidate is its semantic relevance to the agent's current state. This leads to two fundamental questions: *Does runtime semantic similarity reflect the potential future utility of a historical chunk? Can this signal guide differentiated compression that preserves useful details while controlling context growth?*

To investigate these questions, inspired by the locality principle in nonparametric learning (Cover & Hart, 1967; Stone, 1977), we empirically analyze successful AppWorld training trajectories by measuring chunk semantic similarity against a runtime state query. We discover that concrete future-value reuse strongly correlates with this semantic similarity, rising progressively from the low-relevance band to the medium and high bands (Figure 1(c)). The current execution state thus provides evidence about which earlier details are likely to matter later. Because reuse remains non-negligible even in the low band, this observation motivates graded compression that retains essential facts across all bands.

Motivated by this insight, we propose **Re**levance-**Band**ed (**ReBand**) Context Compression, a training-free compression framework that requires no model parameter updates. At each compression event, ReBand preserves the latest interaction verbatim and scores newly eligible chunks against a query formed from the task, previous summary, and current interaction. Fixed quantile thresholds, calibrated once on AppWorld training trajectories, assign these chunks to low-, medium-, or high-relevance bands. The resulting guidance calls for aggressive compression of low-relevance content while retaining essential fields, concise factual summaries for medium-relevance chunks, and structured key-value retention for high-relevance chunks. All annotated chunks are processed jointly with the previous summary in a single LLM compressor call, allowing related facts to be consolidated into a coherent rolling state.

In summary, our contributions are threefold:

- We empirically identify a positive correlation between runtime semantic relevance and future-value reuse in successful AppWorld training trajectories, motivating relevance-guided context compression.
- We introduce ReBand, a training-free framework that assigns distinct compression strategies at the chunk level before consolidating history into a single rolling summary, without model parameter updates or persistent storage of raw history.

- We validate ReBand across three diverse long-horizon benchmarks, demonstrating consistent gains over the leading baseline under identical settings. Specifically, ReBand improves accuracy by 4.2 and 3.16 points on AppWorld and OfficeBench, respectively, and F1 by 4.8 points on Multi-objective QA, while reducing peak context size by 42.4%.

2 Related Work

2.1 Context Compression for LLMs

One family of context compression methods reduces prompts through selective retention. The LLMLingua (Jiang et al., 2023; 2024) line estimates token utility from language-model likelihood, query relevance, or a distilled bidirectional encoder (Pan et al., 2024), while TACO-RL (Shandilya et al., 2025) learns selection from downstream task rewards. Despite using different supervision signals, these methods share a token-selection view: they preserve the prompt's basic structure and allocate a limited budget to locally important content.

A second family constructs compact evidence representations instead of pruning isolated tokens. Extractive or abstractive compressors can select passages, synthesize retrieved documents, or iteratively refine evidence as reasoning progresses (Xu et al., 2024; Yun & Kim, 2026). This approach is organized around a query and a supplied evidence set. Long-horizon agents introduce a different lifecycle: new actions and observations continually extend the history, and the relevance of an earlier interaction changes with the execution state.

2.2 Context Compression for Long-Horizon Agents

Summarization-based memory replaces an expanding trajectory with a compact state carried across decisions, optimized via periodic compression, joint reinforcement learning, or reusable natural-language guidelines (Wu et al., 2025; Lu et al., 2026; Kang et al., 2026). ACON serves as our direct baseline by optimizing guidelines from failure feedback without model updates. A complementary family treats memory as part of the agent policy through consolidation or folding (Zhou et al., 2026; Ning et al., 2026; Ye et al., 2026; Sun et al., 2026). Notably, PACE introduces relevance-conditioned representation selection over a persistent history store (Wei et al., 2026).

Unlike PACE's repeated selection across an external memory store, ReBand operates under an irreversible rolling-summary regime without historical storage buffers. Newly eligible chunks are scored against a runtime state query and assigned low-, medium-, or high-compression guidance via calibrated quantiles, followed by a single consolidated compressor call. We focus our main evaluations on rolling-memory baselines (e.g., ACON), with our reproduction details for PACE and empirical observations deferred to Appendix D.

3 Methodology

3.1 Problem Definition

As illustrated in Figure 2, we consider a long-horizon agent that repeatedly interacts with an environment, accumulating action–observation history throughout execution. We formalize this context compression setting below. Given a task x, an agent interacts with an environment over iterations indexed by t, with final interaction index T. At iteration t, the agent executes an action a_t and receives an observation o_t, forming an interaction chunk $c_t = (a_t, o_t)$. System instructions and the original task are excluded from the chunk sequence. Let S_{t-1} denote the rolling summary available before iteration t, with an empty initial summary S_0, and let U_t denote the ordered sequence of older raw chunks that have not yet been incorporated into the summary and are eligible for compression.

3.2 Empirical Motivation

Our relevance-banded context compression framework is inspired by the locality principle in statistical learning, where nearby instances in a representation space ofter exhibit related predictive behavior (Cover & Hart, 1967; Stone, 1977). This suggests that one can explore whether the semantic

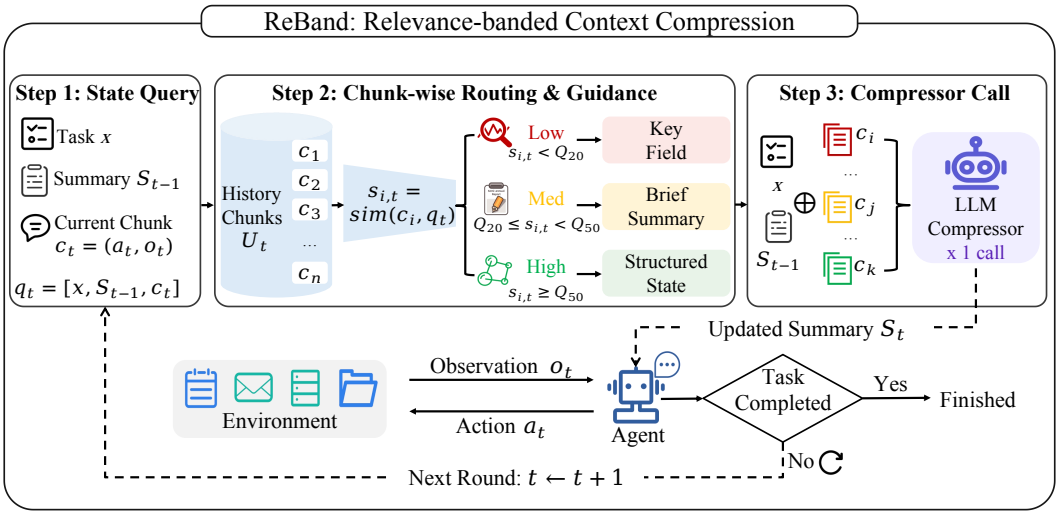


Figure 2: **Overview of ReBand**. A state query built from the task, previous summary, and current interaction scores each unsummarized chunk. Calibrated quantile bands assign chunk-specific compression guidance, and $\oplus$ denotes concatenation, packaging the task, prior summary, and tiered chunks into a single compressor call to update the rolling summary for subsequent agent decisions.

relevance between the most recent historical interaction segments and the agent's current execution state serves as a valid signal for assessing potential future utility. As illustrated in Figure 1 (c), historical chunks with higher semantic similarity exhibit higher future-value reuse rates, while chunks with lower similarity also exhibit non-negligible reuse rates (see Appendix F for the formal definition and computation of future-value reuse). This finding motivates allocating compression intensity by semantic relevance, rather than uniformly compressing or pruning history.

3.3 Relevance-Banded Compression Allocation

At runtime, ReBand constructs the same state query used to calibrate the relevance distribution on successful AppWorld training trajectories. When compression is triggered at iteration t, we construct

$$q_t = [x, S_{t-1}, c_t], \tag{1}$$

where $[\cdot, \cdot, \cdot]$ concatenates the task, previous summary, and current interaction. We use a frozen Qwen3-Embedding model E (Zhang et al., 2025) to encode this query and every chunk in U_t. The relevance of chunk c_i is

$$s_{i,t} = \mathrm{sim}(c_i, q_t) = \frac{E(c_i)^\top E(q_t)}{\|E(c_i)\|_2 \, \|E(q_t)\|_2}. \tag{2}$$

Here, sim is cosine similarity between the chunk and query embeddings, and all chunks in U_t are scored against the same query. The thresholds Q_{20} and Q_{50} are the 20th and 50th percentiles of the pooled calibration scores and are frozen before evaluation. The compression-band label $b_{i,t}$ is assigned as follows:

$$b_{i,t} = \begin{cases} \text{low}, & s_{i,t} < Q_{20}, \\ \text{medium}, & Q_{20} \le s_{i,t} < Q_{50}, \\ \text{high}, & s_{i,t} \ge Q_{50}. \end{cases} \tag{3}$$

At runtime, Q_{20} and Q_{50} are not recomputed on any evaluation split. The routing is dynamic because q_t, U_t, and the resulting scores change with execution progress. Each raw chunk receives a routing decision only when it enters a compression batch. Successfully compressed raw chunks are not reclassified at later events.

3.4 Semantically-Guided Batch Compression

The router converts each compression-band label into its corresponding compression guidance. Let G denote this mapping and define $g_{i,t} = G(b_{i,t})$. The annotated compression batch is

$$\widetilde{U}_t = \{(c_i, g_{i,t}) \mid c_i \in U_t\}. \tag{4}$$

These bands define three compression-fidelity levels:

Low band ($s_{i,t} < Q_{20}$): aggressive compression. The chunk is treated as background context; the compressor keeps only the few key fields that are explicitly needed for the current task and discards or heavily condenses the remainder.

Medium band ($Q_{20} \leq s_{i,t} < Q_{50}$): moderate compression. The chunk is summarized in one or a few sentences that capture the action taken, the outcome, and any facts that may still matter, while omitting verbose observations and intermediate reasoning.

High band ($s_{i,t} \geq Q_{50}$): conservative, structured compression. The chunk is preserved in a structured form that operationalizes the original unstructured interaction: identifiers, parameters, constraints, state transitions, and completed actions are extracted and kept as structured facts.

All annotated chunks are processed jointly in one history-compression call. Here, $\mathrm{Tok}(\cdot)$ denotes the token count under the agent's tokenizer. The rolling summary is updated as

$$S_t = \begin{cases} C\Big(x, S_{t-1}, \widetilde{U}_t\Big), & \mathrm{Tok}(S_{t-1} \oplus U_t) > \tau_{\mathrm{ctx}}, \\ S_{t-1}, & \mathrm{Tok}(S_{t-1} \oplus U_t) \leq \tau_{\mathrm{ctx}}. \end{cases} \tag{5}$$

This joint call allows the compressor to consolidate related facts across chunks. After a valid summary is returned, the memory manager removes the compressed raw chunks, and the active context contains the task, updated summary, and verbatim current chunk. If compression is not triggered, $S_t = S_{t-1}$, and the unsummarized chunks remain in the active context. The next compression event consumes the carried-forward summary and only the newly eligible raw chunks. Each raw chunk receives one routing and compression decision, while retained information may be rewritten through the rolling sequence $S_{t-1} \to S_t$. The concrete prompt templates for the compressor across all bands and comparative baselines are provided in Appendix E.

4 Experiments

We evaluate ReBand on three long-horizon benchmarks spanning interactive applications, office productivity, and multi-objective question answering. We organize the experiments around four questions:

- **Performance (Section 4.2).** Does ReBand improve task performance across heterogeneous long-horizon benchmarks?
- **Mechanism effectiveness (Section 4.3).** Do band-conditioned compression and relevance-guided band assignment each contribute to the gains of ReBand?
- **Performance–efficiency trade-off (Section 4.4).** How does ReBand trade task performance against peak context, trajectory steps, and dependency cost?
- **Cross-model transfer (Section 4.5).** Do relevance thresholds calibrated on AppWorld transfer to a different agent/compressor LLM model without recalibration?

4.1 Experimental Settings

Benchmarks. In long-horizon tasks, each benchmark requires at least ten interaction steps on average. **AppWorld** (Trivedi et al., 2024) provides nine simulated applications, including Venmo, Spotify, and SimpleNote, for approximately 100 simulated users; we report its official task-completion score. **OfficeBench** (Wang et al., 2024) evaluates document-centric tasks across six applications, including Word, Excel, and Email, using benchmark-defined accuracy functions. **Multi-objective QA** combines eight questions sampled from natural questions into one long-horizon task (Kwiatkowski et al., 2019; Zhou et al., 2026). The agent searches iteratively and returns eight consolidated answers, which we evaluate using question-level EM and F1. Additional dataset and metric details are provided in Appendix B.

Baselines, models, and controlled implementation. For a fair comparison, we follow ACON (Kang et al., 2026) and compare against seven context-management settings. **No Compression** retains the full interaction history. **FIFO** keeps the latest k interactions(Yang et al., 2024), whereas **Retrieval** selects the k interactions most similar to the current query (Xu et al., 2025). **LLMLingua**

Table 1: AppWorld results by difficulty. Each block reports accuracy, steps (averaged over completed trajectories), peak input tokens (10^3) and dependency (10^6) for agents. Metric definitions are provided in Appendix B.2. **Bold** indicates the best result in each column, Underlined marks the second-best result. Row in green background indicate the our results.

Method	Average (168)				Easy (57)			Medium (48)			Hard (63)		
	Acc. ↑	Steps ↓	Peak ↓	Dep. ↓	Acc. ↑	Peak ↓	Dep. ↓	Acc. ↑	Peak ↓	Dep. ↓	Acc. ↑	Peak ↓	Dep. ↓
Agent: GPT-4.1 / **Compressor:** GPT-4.1													
No Compression	56.0	16.14	9.93	5.96	80.7	7.57	2.98	47.9	10.10	5.36	39.7	11.95	9.11
FIFO	45.8	28.48	**6.73**	5.69	84.2	**5.85**	2.89	39.6	7.26	6.24	15.9	**7.14**	7.80
Retrieval	27.4	33.17	8.39	6.68	61.4	7.40	3.97	12.5	8.74	7.72	7.9	9.02	8.33
LLMLingua	39.3	24.42	7.50	6.37	66.7	6.38	3.04	37.5	8.04	7.39	15.9	8.09	8.59
Prompting	47.0	27.88	7.22	5.21	73.7	6.96	2.81	45.8	7.18	5.06	23.8	7.49	7.50
ACON-UT	51.2	**20.92**	7.17	**4.49**	77.2	6.45	**2.43**	50.0	7.39	4.47	28.6	7.65	6.37
ACON-UTCO	56.5	22.82	7.33	4.69	**86.0**	7.09	2.84	56.2	7.48	**4.43**	30.2	7.44	6.55
ReBand	**60.7**	21.77	7.30	4.54	80.7	7.34	2.73	**58.3**	**6.90**	4.60	**44.4**	7.58	**6.13**

retains high-importance tokens under a target budget (Jiang et al., 2023), and **Prompting** summarizes history with a manually written general-purpose instruction (Lee et al., 2025). **ACON** learns task-specific compression guidelines from trajectory-level feedback: ACON-UT optimizes downstream utility, and ACON-UTCO further optimizes compression length (Kang et al., 2026). The main cross-benchmark comparison uses GPT-4.1 (OpenAI, 2025) as both agent and compressor. Section 4.5 additionally replaces both components with Qwen3.6-27B on AppWorld to examine whether the relative gain over ACON-UTCO persists across model backbones.

For the direct ACON comparison, our implementation follows ACON's agent/compressor role assignment and aligns all non-method settings with ACON: the same agent and compressor model within each comparison, benchmark split and task set, interaction/iteration budget, context-compression trigger, number of most-recent interactions preserved, verifier setting, random seed, and completion-conditioned evaluation protocol. The only intended change is the compression policy applied to the history chunks. ACON-UT and ACON-UTCO use their corresponding ACON guidelines, whereas ReBand replaces the single uniform guideline with relevance-banded, chunk-specific compression guidance and then performs the same rolling-summary update. Therefore, comparisons are conducted at the levels of the proxy, compressor, and experimental setup. The observed differences are intended to evaluate the compression methods rather than a change in model capability or execution budget.

Threshold transfer. The band boundaries are calibrated once from the pooled similarity distribution across all eligible chunks in the AppWorld training split (Section 3.2), yielding operational cutoffs $Q_{20} = 0.53$ and $Q_{50} = 0.67$. Rather than re-calibrating per domain, we freeze these quantile cutoffs as zero-shot anchors across AppWorld test-normal, OfficeBench, and Multi-objective QA. No threshold is recomputed or tuned on any evaluation split, strictly testing whether percentile banding transfers without per-dataset calibration. Crucially, runtime chunk tracking confirms this frozen anchor adaptively aligns compression fidelity with task dynamics (e.g., favoring task-critical tracking for OfficeBench while aggressively pruning QA distractors) without single-band collapse (full statistics in Appendix C).

4.2 Main results

As reported in Tables 1–3, ReBand is evaluated across interactive application execution, office automation, and multi-objective information seeking. In each setting, ReBand improves the primary task metric over ACON-UTCO while maintaining a comparable or smaller Peak.

Interactive application execution. AppWorld requires the agent to carry execution state through extended interactions. ReBand reaches 60.7% accuracy, exceeding ACON-UTCO and No Compression by 4.2 and 4.7 points. Its gains are concentrated on Medium (+2.1 points) and Hard (+14.2 points) tasks. While Easy accuracy drops slightly by 5.3 points—where short trajectories need minimal tracking and summarization risks minor rewrite noise—relevance-banded compression decisively benefits complex, long-horizon workflows, driving a 4.2 points overall improvement.

Table 2: OfficeBench results by application count. Each block reports accuracy, steps (averaged over completed trajectories), peak input tokens (10^3) and dependency (10^6) follow Table 1. **Bold** marks the best compression method, underlined marks the second-best result. ReBand consistenly improves acc/efficiency trade-offs.

Method	Average (95)				1 app (42)			2 apps (22)			3 apps (31)		
	Acc. ↑	Steps ↓	Peak ↓	Dep. ↓	Acc. ↑	Peak ↓	Dep. ↓	Acc. ↑	Peak ↓	Dep. ↓	Acc. ↑	Peak ↓	Dep. ↓
Agent: GPT-4.1 / **Compressor:** GPT-4.1													
No Compression	76.84	11.52	7.27	4.43	92.86	6.23	4.05	77.27	6.14	1.81	54.84	8.37	6.08
FIFO	67.37	12.26	4.02	2.64	83.33	4.19	0.72	63.64	**3.51**	**1.01**	48.39	**4.23**	4.39
Retrieval	65.26	16.20	4.33	2.06	85.71	4.35	0.84	63.64	3.52	1.37	38.71	4.78	2.99
LLMLingua	70.53	10.89	4.65	1.85	83.33	4.17	**0.67**	68.18	4.61	1.18	54.84	4.88	2.74
Prompting	71.58	**10.13**	4.40	**1.10**	85.71	4.18	0.81	**77.27**	4.53	1.08	48.39	4.42	**1.23**
ACON-UT	74.74	13.13	4.93	3.85	85.71	4.71	6.89	72.73	4.64	1.44	**61.29**	5.19	3.89
ACON-UTCO	72.63	11.54	4.54	1.91	88.10	3.92	0.76	72.73	4.72	1.16	51.61	4.71	2.84
ReBand	**75.79**	10.78	**3.95**	1.24	**92.86**	**3.56**	1.02	72.73	4.62	1.04	54.84	4.43	1.68

Table 3: Multi-objective QA results (EM/F1 in [0,1]). Each block reports EM/F1, steps, peak input tokens (10^3) and dependency (10^6) follow Table 1. **Bold** marks the best compression method, underlined marks the second-best result.

Method	EM ↑	F1 ↑	Steps ↓	Peak ↓	Dep. ↓
Agent: GPT-4.1 / **Compressor:** GPT-4.1					
No Compression	0.366	0.488	15.78	10.35	3.32
FIFO	0.293	0.388	19.26	5.09	2.51
Retrieval	0.331	0.438	20.06	5.11	2.62
LLMLingua	0.363	0.481	17.68	5.68	2.24
Prompting	**0.376**	0.478	18.70	4.73	1.66
ACON-UT	0.373	0.494	17.14	4.71	1.57
ACON-UTCO	0.335	0.458	17.79	4.65	**1.50**
ReBand	0.375	**0.506**	**12.01**	**2.68**	2.13

Cross-application office automation. OfficeBench extends the evaluation to document-centric workflows spanning one or more office tools. ReBand achieves the highest average accuracy (75.79%) and lowest average Peak (3.95k) among compression methods. Accuracy is 3.16 points above ACON-UTCO, with gains on 1-app and 3-app tasks and unchanged 2-app accuracy. Relative to ACON-UTCO, Peak decreases by 13.0%, Steps by 6.6%, and Dep. by 35.1%. Compared to No Compression, accuracy drops 1.05 points, while Peak and Dep. fall 45.7% and 72.0%, respectively.

Multi-objective information seeking. The Multi-objective QA setting requires repeated search and answer consolidation across multiple objectives. Under ACON's completion-conditioned protocol with a consistent denominator, ReBand achieves the highest F1 (0.506), an EM only 0.001 below the best result, and the lowest Steps (12.01) and Peak (2.68k). Against ACON-UTCO, F1 increases by 4.8 points as Steps and Peak fall by 32.5% and 42.4%. Dep. rises from 1.50M to 2.13M, limiting the efficiency gain to trajectory length and peak context rather than cumulative computation.

Taken together, these results show that ReBand generalizes across heterogeneous long-horizon settings. Despite using the same AppWorld-calibrated thresholds for all datasets, ReBand improves over ACON-UTCO on interactive execution, office automation, and multi-objective information seeking, while maintaining a comparable or smaller peak context.

4.3 Ablation Results

ReBand combines two core components: band-conditioned compression and semantic relevance-guided band assignment. The former introduces differentiated Low/Medium/High compression guidance within a unified prompt, while the latter routes each history chunk to its corresponding rule based on semantic relevance. Table 4 evaluates these components incrementally. Base Prompting uses uniform compression, whereas Shuffled-Band retains the same specific guidance and band

Table 4: Mechanism ablation on AppWorld test-normal tasks with GPT-4.1. We isolate the effects of band-conditioned compression and relevance-guided band assignment by comparing base prompting, shuffled band assignments, and the full ReBand method. Metric definitions follow Table 1. Δ Acc. denotes the accuracy change relative to the preceding row. **Bold** marks the best compression method, underlined marks the second-best result.

Method	Acc. ↑	Steps ↓	Peak ↓	Dep. ↓	Δ Acc.
Base Prompting	47.0	27.88	**7.22**	5.21	–
+ Band-Conditioned Compression (Shuffled)	55.3	24.70	7.53	4.94	+8.3
+ Relevance-Guided Band Assignment (ReBand)	**60.7**	**21.77**	7.30	**4.54**	+5.4

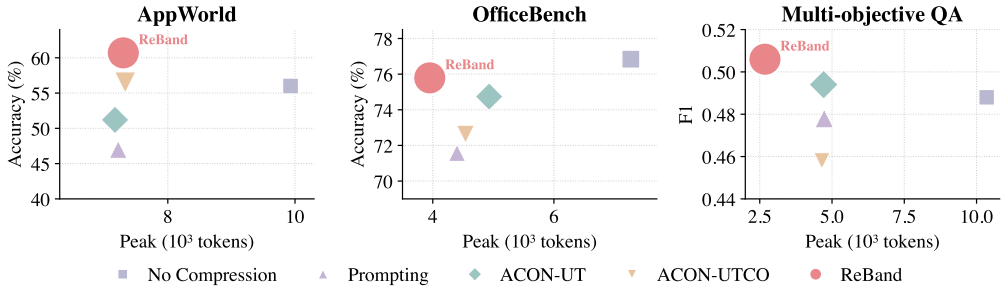


Figure 3: Performance versus peak context trade-off across AppWorld, OfficeBench, and multi-objective QA. Peak denotes the input tokens during the agent. Performance is measured by accuracy for AppWorld and OfficeBench and F1 for multi-objective QA. Lower Peak and higher performance are preferred. ReBand consistently outperforms ACON-UTCO at similar or lower Peak.

counts as ReBand but randomly assigns bands to chunks. ReBand then restores relevance-banded assignment under otherwise identical settings.

As reported in Table 4, band-conditioned compression with random assignment improves accuracy from 47.0% to 55.3% (+8.3 points). This initial boost demonstrates that tiered compression guidance provide a beneficial structural prior, preventing the rolling summary from homogenizing all historical details into uniform brevity. Crucially, replacing random permutation with semantic relevance-guided assignment yields a further 5.4 points gain (reaching 60.7%), while also reducing trajectory steps, peak tokens, and dependency. These results indicate that tiered compression itself provides a useful structural prior, while aligning compression strategy with semantic relevance further improves both task performance and execution efficiency.

4.4 Performance vs Peak Context

As visualized in Figure 3, where the x-axis represents peak tokens and the y-axis represents accuracy (with the top-left corner indicating superior performance), ReBand consistently achieves higher task performance than ACON-UTCO at a similar or lower Peak across all three benchmarks. On AppWorld, accuracy is 4.2 points above ACON-UTCO, while Peak decreases slightly from 7.33k to 7.30k. On OfficeBench, accuracy is 3.16 points above ACON-UTCO, while Peak decreases from 4.54k to 3.95k, a 13.0% reduction. On multi-objective QA, F1 increases by 4.8 points while Peak decreases by 42.4%. Overall, these results indicate that relevance-banded compression provides a better performance-context trade-off, achieving stronger task performance without increasing peak context usage.

4.5 Transferability Across Models

As shown in Table 5, when both the agent and compressor are changed from GPT-4.1 to Qwen3.6-27B on the same AppWorld test-normal tasks, ReBand remains effective under same experiment settings. Across both GPT-4.1 and Qwen3.6-27B, ReBand achieves the highest accuracy among Base Prompting, ACON-UTCO, and ReBand. In particular, ReBand improves over ACON-UTCO by 4.2 points with either model, supporting its effectiveness and transferability without model-specific threshold calibration. Relative to Qwen Base Prompting, ReBand gains 8.4 points Acc. with 0.87

Table 5: AppWorld test-normal results across agent/compressor models. Metrics follow Table 1. ReBand improves Acc. over ACON-UTCO by 4.2 points with both models. **Bold** marks the best result within each model, underlined marks the second-best result.

Model	Method	Acc. ↑	Steps ↓	Peak ↓	Dep. ↓
GPT-4.1	Base Prompting	47.0	27.88	**7.22**	5.21
	ACON-UTCO	56.5	22.82	7.33	4.69
	ReBand	**60.7**	**21.77**	7.30	**4.54**
Qwen3.6-27B	Base Prompting	56.5	29.07	**6.76**	4.87
	ACON-UTCO	60.7	30.20	6.82	**4.40**
	ReBand	**64.9**	**28.20**	6.87	4.84

less Steps and a 0.11k larger Peak, while Dep. decreases from 4.87M to 4.84M. Its gain over Qwen ACON-UTCO requires 0.11k additional Peak.

4.6 Analysis of ReBand Behaviors

As can be seen in Figure 4, ReBand preserves pending items in multi-app execution, combines summarized progress with precise recent values in office workflows, and retains ealier answers in multi-objective question answering. These cases demonstrate that relevance-banded compression can adapt to different task structures while maintaining actionable information for downstream tasks.

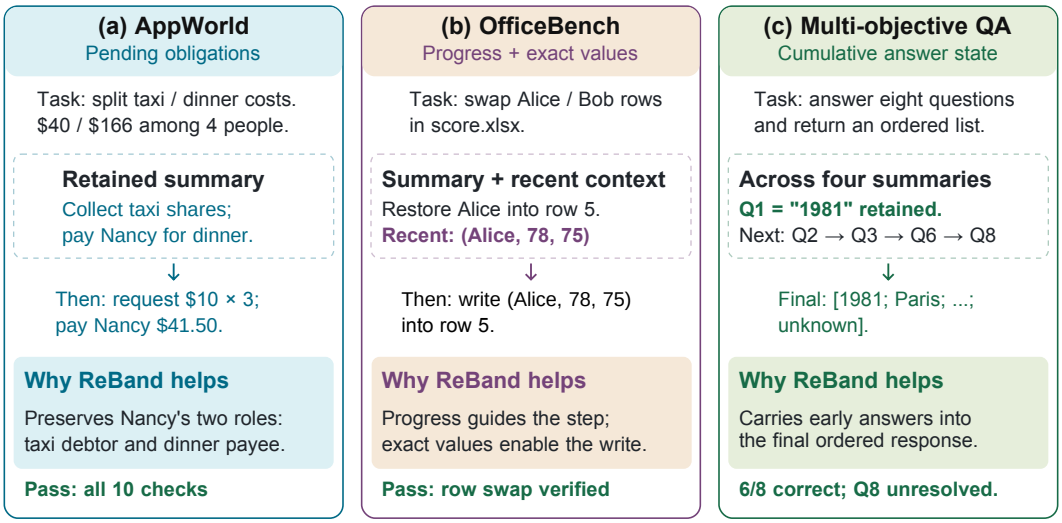


Figure 4: Analysis of ReBand's compression behavior across long-horizon benchmarks. The panels demonstrate how relevance-banded compression preserves critical details through three distinct mechanisms: (a) maintaining pending obligations in AppWorld (passing all 10 checks), (b) pairing progress summaries with exact values in OfficeBench for verified row swaps, and (c) carrying cumulative answer states across summaries in Multi-objective QA (achieving 6/8 correct).

5 Conclusion

In this work, we introduce ReBand, a training-free context compression framework for long-horizon LLM agents. ReBand dynamically estimates the semantic relevance of historical interaction chunks to the agent's current runtime state and uses it to allocate different compression strategies across chunks, reducing the loss of critical information caused by uniform compression. Across three benchmarks, ReBand consistently outperforms uniform compression methods while maintaining comparable or lower peak context usage. These results demonstrate that semantic relevance provides a simple and effective signal for guiding differentiated compression in long-horizon agents.

AI USE STATEMENT

An AI assistant was utilized to assist with language polishing, draft LaTeX formatting, organize experimental results into tables, and check textual consistency. No scientific ideas, methodologies, algorithms, or experimental results were generated by the AI assistant. The authors have rigorously reviewed, verified, and take full responsibility for all content, claims, and citations in this work.

REPRODUCIBILITY STATEMENT

The core methodology, mathematical formulations, and experimental configurations of ReBand are fully specified in Section 3, Section 4 and Appendix F. All evaluation benchmarks used in this work are publicly available, please refer to Appendix B.1. To facilitate reproducibility and future research, our complete source code and calibrated threshold configurations will be made publicly available upon publication.

REFERENCES

Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. doi: 10.1109/TIT.1967.1053964.

Guanting Dong, Xiaoshuai Song, Yuyang Hu, Jiajie Jin, Chenghao Zhang, Yifei Chen, Xiaoxi Li, Huaying Yuan, Xinyu Yang, Tongyu Wen, Jiejun Tan, Hongjin Qian, Shijue Huang, Junting Lu, Zhenyu Li, Wanjun Zhong, Yutao Zhu, Tat-Seng Chua, Zhicheng Dou, and Ji-Rong Wen. Towards long-horizon agents: A survey. *Preprints*, July 2026. doi: 10.20944/preprints202607.1328.v1. URL https://doi.org/10.20944/preprints202607.1328.v1.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825/.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.91. URL https://aclanthology.org/2024.acl-long.91/.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. ACON: Optimizing context compression for long-horizon LLM agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=5EmOOLtH5P.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Dongjun Lee, Juyong Lee, Kyuyoung Kim, Jihoon Tack, Jinwoo Shin, Yee Whye Teh, and Kimin Lee. Learning to contextualize web pages for enhanced decision making by LLM agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=3Gzz7ZQLiz.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7182–7195, Albuquerque,

New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.368. URL https://aclanthology.org/2025.naacl-long.368/.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, Yuanxing Zhang, Zhuo Chen, Hangyu Guo, Shilong Li, Ziqiang Liu, Yong Shan, Yifan Song, Jiayi Tian, Wenhao Wu, Zhejian Zhou, Ruijie Zhu, Junlan Feng, Yang Gao, Shizhu He, Zhoujun Li, Tianyu Liu, Fanyu Meng, Wenbo Su, Yingshui Tan, Zili Wang, Jian Yang, Wei Ye, Bo Zheng, Wangchunshu Zhou, Wenhao Huang, Sujian Li, and Zhaoxiang Zhang. A comprehensive survey on long context language modeling, 2025. URL https://arxiv.org/abs/2503.17407.

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, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Miao Lu, Weiwei Sun, Weihua Du, Zhan Ling, Xuesong Yao, Kang Liu, and Jiecao Chen. Beyond the context window: Scaling agentic RL via end-to-end optimized context compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21074–21125, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.966. URL https://aclanthology.org/2026.acl-long.966/.

Yansong Ning, Jun Fang, Naiqiang Tan, and Hao Liu. Agent-omit: Adaptive context omission for efficient LLM agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YX4rgoI8pB.

OpenAI. Introducing gpt-4.1 in the api. OpenAI Blog, April 2025. URL https://openai.com/index/gpt-4-1/.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.57. URL https://aclanthology.org/2024.findings-acl.57/.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=tBRNC6YemY.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Yacmpz84TH.

Shivam Shandilya, Menglin Xia, Supriyo Ghosh, Huiqiang Jiang, Jue Zhang, Qianhui Wu, Victor Rühle, and Saravan Rajmohan. TACO-RL: Task aware prompt compression optimization with reinforcement learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 1582–1597, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.81. URL https://aclanthology.org/2025.findings-acl.81/.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. 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. 31210–31227. PMLR, 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

Charles J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–620, 1977. doi: 10.1214/aos/1176343886.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling long-horizon agent via context folding. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lNRgWoGfYg.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. OfficeBench: Benchmarking language agents across multiple applications for office automation, 2024. URL https://arxiv.org/abs/2407.19056.

Lei Wei, Xiao Peng, Tt, Guannan Zhang, Chenhao Jiang, Hongyu Li, Lanbo Lin, Yuanwu Xu, Jiayao Liu, Kesu Wang, and Bin Wang. PACE: Predictive adaptive context extraction for long-horizon LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27184–27199, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1252. URL https://aclanthology.org/2026.acl-long.1252/.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Minhao Cheng, Shuai Wang, Hong Cheng, and Jingren Zhou. ReSum: Unlocking long-horizon search intelligence via context summarization, 2025. URL https://arxiv.org/abs/2509.13313.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented lms with compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*. OpenReview.net, 2024. URL https://openreview.net/forum?id=mlJLVigNHp.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=FiM0M8gcct.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=mXpq6ut8J3.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Jingren Zhou, Siheng Chen, and Yong Jiang. Agentfold: Long-horizon web agents with proactive context folding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IuZoTgsUws.

JungMin Yun and YoungBin Kim. IterCOMP: Reasoning-aware adaptive prompt compression for multi-hop question answering. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33827–33840, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1559. URL https://aclanthology.org/2026.acl-long.1559/.

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

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Pu Liang. MEM1: Learning to synergize memory and reasoning for efficient long-horizon agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XY8AaxDSLb.

A Limitations and Future Works

Relevance as a compression-allocation signal. Our analysis shows that semantic relevance provides a useful signal for allocating compression detail in long-horizon agent histories. Howerver, in extreme cases, information critical to execution may appear in data segments with low relevance. Therefore, an important direction for expansion is to combine relevance-based allocation strategies with factual causal signals and to investigate how these signals interact during the iterative process of rolling summarization.

Calibration and transfer. ReBand uses relevance-band thresholds calibrated once on successful AppWorld train-split trajectories and applied them unchanged across the evaluated benchmarks. The resulting gains suggest that the calibrated bands transfer beyond the source environment without specific tuning. Future work can further study calibration robustness across different domains when larger distribution shifts.

B Experimental Setup Details

This section supplements the experimental settings in Section 4.1 with dataset composition, benchmark-specific execution details, and evaluation metrics. The main experiments use GPT-4.1 as both agent and compressor, with the additional AppWorld model comparison described in Section 4.5.

B.1 Datasets

AppWorld. AppWorld is a benchmark for interactive coding agents operating in a simulated ecosystem of everyday applications (Trivedi et al., 2024). Its environment provides nine applications through 457 APIs and models the digital activities of approximately 100 fictitious users. Tasks require agents to write and execute code, inspect intermediate results, and coordinate operations across applications. Evaluation uses tests of the resulting application state to check task completion and unintended changes, allowing different valid execution paths.

OfficeBench. OfficeBench evaluates language agents on office workflows involving applications such as Word, Excel, PDF, Calendar, and Email (Wang et al., 2024). Tasks use synthetic documents, messages, and calendar events in an executable environment, and are organized by whether they require one, two, or three applications. Agents must select appropriate operations and switch applications as the workflow progresses. Task-specific evaluators check the resulting artifacts using exact matching, fuzzy matching, or execution-based tests.

Multi-objective QA. Multi-objective QA extends question answering into a longer interaction by combining multiple questions into one task, following the construction introduced in MEM1 (Zhou et al., 2026). The Multi-objective variant groups eight Natural Questions items and requires a consolidated set of answers after interaction with a search tool (Kang et al., 2026). Natural Questions consists of real search queries paired with human-annotated answers in Wikipedia pages (Kwiatkowski et al., 2019). The composite task requires agents to gather evidence across successive searches and retain information needed to answer all constituent questions.

B.2 Evaluation Metrics

Task performance. For AppWorld and OfficeBench, Acc. is the percentage of tasks judged successful by the respective benchmark evaluators. We report overall accuracy and the difficulty or application-count breakdowns in Tables 1 and 2. For QA, question-level exact match (EM) and F1 are reported on a scale from 0 to 1. Following the completion-conditioned aggregation used by ACON, ReBand obtains an EM of 0.375 and an F1 of 0.506. These scores describe completed trajectories under that protocol.

Execution steps. Steps is the mean number of agent interaction steps over completed trajectories.

Peak tokens. Peak tokens measure the maximum sequence length observed during an agent trajectory, excluding system prompts. This metric serves as a proxy for inference-time memory requirements. Following the efficiency evaluation used in ACON and MEM1 (Kang et al., 2026; Zhou et al., 2026), we report the mean trajectory-level peak in units of 10^3 tokens as Peak. Lower values indicate a smaller maximum context footprint.

Dependency. Dependency approximates the cumulative computational cost of action generation over an agent trajectory. For a trajectory with T steps, let $n_i^{(t)}$ and $n_o^{(t)}$ denote the numbers of input and output tokens at step t, respectively. Following the dependency metric used in ACON (Kang et al., 2026), we compute

$$\text{Dep.} = \sum_{t=1}^{T} \frac{\left(n_i^{(t)} + 2n_o^{(t)}\right) n_o^{(t)}}{2}. \tag{6}$$

The metric accumulates the contribution of each action-generation step, accounting for both its input context and generated output. We report the mean trajectory-level dependency in units of 10^6. Lower values indicate a smaller estimated cumulative cost of action generation.

Metric aggregation. AppWorld token statistics use recorded trajectories with the common `o200k_base` tokenizer and message-accounting convention across methods. QA step and cost statistics follow the same completion-conditioned evaluation protocol as its answer-quality metrics.

C Distribution of Relevance Bands

A potential concern regarding frozen numerical thresholds is whether domain shift induces distributional collapse (e.g., all chunks falling into a single band). As shown in Table 6, neither extreme degradation (0% allocation) nor single-band saturation ($\geq 90\%$) occurs, confirming the stable operational validity of the frozen cutoffs.

Crucially, the cross-benchmark divergence in band proportions directly reflects the distinct informational dynamics of each task regime:

- **OfficeBench (61.8% High Band):** Document-centric workflows exhibit dense state continuity, where intermediate parameters (e.g., file paths, extracted cells, recipient identities) are consistently re-accessed across subsequent tool invocations. The state query faithfully assigns High-band compression to these essential state transitions, while safely compressing isolated operational noise.
- **Multi-objective QA (42.3% Low Band):** In multi-objective information seeking, the trajectory alternates across disjoint sub-questions. Information gathered for resolved questions transitions into background context, while candidate search trails often yield irrelevant distractors. The framework adaptively demotes 42.3% of chunks to Low-band aggressive compression, explaining the dramatic 42.4% reduction in peak context window while preventing distractor accumulation.

These distributions confirm that rather than acting as a rigid, uniform partition, the calibrated thresholds provide a principled semantic anchor that dynamically aligns compressi g fidelity with benchmark-specific information density.

Table 6: Empirical distribution of unsummarized history chunks across calibrated relevance bands during runtime evaluation under frozen thresholds ($Q_{20} and Q_{50}$). Low, Medium, and High bands correspond to similarity intervals $< Q_{20}$, $[Q_{20}, Q_{50})$, and $\geq Q_{50}$, respectively.

Benchmark	Total Chunks (N)	Low Band	Medium Band	High Band
AppWorld (Train)	1,482	20.0%	30.0%	50.0%
OfficeBench	709	21.8%	16.4%	61.8%
Multi-objective QA	1,111	42.3%	17.6%	40.1%

D COMPARISON WITH EXTERNAL-STORE MEMORY(PACE)

To contextualize relevance-guided compression within the broader landscape of agent memory, this section details our exploration of PACE (Wei et al., 2026) on AppWorld. As official source code has not been released at the time of submission, we implemented a PACE-style context manager faithful to the architectural formulation described in Wei et al. (2026).

D.1 IMPLEMENTATION CONFIGURATION

The context manager retains every historical action–observation chunk in an external memory buffer, together with cached detailed summaries, brief summaries, and lightweight placeholder representations.

- **Summary Generation and Semantic Scoring:** We generate both detailed and brief summaries synchronously using GPT-4.1. Semantic relevance is evaluated using a frozen Qwen3-Embedding-8B model.
- **Runtime Selection:** At each agent decision step, the runtime query is constructed by concatenating the original task description with the two most recent interaction chunks. Older chunks in the memory buffer are scored against this query via cosine similarity. These raw similarity scores are normalized into relative representation weights using a temperature-scaled softmax.
- **Dynamic Thresholding:** Chunk representations (full chunk, detailed summary, brief summary, or placeholder) are assigned dynamically using thresholds that adaptively tighten with task step progression and cumulative context pressure.
- **Recent Chunk Buffering and Active Retrieval:** The two most recent interaction turns are unconditionally preserved uncompressed. To allow retrospective recovery of compressed information, the agent is equipped with a `Glimpse` command to fetch and restore the verbatim contents of older chunks on demand.

D.2 EMPIRICAL RESULTS AND FAILURE ANALYSIS

We evaluated this re-implementation on the 168 AppWorld test-normal tasks under identical environment limits, iteration budgets (50 steps), and base model backbones (GPT-4.1) as reported in Section 4. In our experiments, the re-implemented PACE context manager achieves 54.7% accuracy on AppWorld (Table 7). We report these empirical findings as a representative reference for external-store memory in programmatic API environments, where missing variables directly trigger execution exceptions.

Table 7: AppWorld test-normal (168 tasks) comparison between rolling-summary methods and the re-implemented PACE framework under identical GPT-4.1 backbones. Metrics follow Table 1.

Method	**Acc. (%)** ↑	**Steps** ↓	**Peak** (10^3) ↓	**Dep.** (10^6) ↓
Rolling-Summary / Pruning Paradigm				
No Compression	56.0	16.14	9.93	5.96
ACON-UTCO	56.5	22.82	7.33	4.69
ReBand (Ours)	**60.7**	**21.77**	**7.30**	**4.54**
External-Store / Dynamic Extraction Paradigm				
PACE (Re-implementation)	54.7	23.80	7.88	5.04

Through detailed trajectory inspections, we identify two primary factors contributing to this performance gap in complex coding environments:

1. **Passive Retrieval Failure:** PACE's architectural advantage hinges on the agent proactively calling the `Glimpse` tool when encountering missing execution variables. In AppWorld's programmatic API setting, when earlier identifiers (e.g., specific payment request tokens or intermediate database IDs) were collapsed into brief summaries or placeholders, GPT-4.1 rarely invoked `Glimpse` to retrieve the raw string. Instead, it frequently hallucinated missing variables or terminated prematurely with execution exceptions.

2. **Representation Oscillation and Context Fragmentation:** Because chunk representations are re-evaluated and swapped at every interaction step based on shifting runtime queries, the structure of the agent's prompt history changed dynamically across iterations. This dynamic switching occasionally fragmented the agent's programmatic reasoning flow compared to a steadily evolving, coherent rolling summary.

D.3 PARADIGMATIC DISTINCTION

These observations underscore the fundamental difference in operational paradigms between the two approaches:

- **PACE** relies on a persistent external store to support repeated representation selection and retrospective recovery of omitted details.
- **ReBand** enforces an irreversible rolling-summary pipeline that assigns graded compression fidelity to unsummarized chunks exactly once before permanently consolidating them into the unified state.

Because of these differing memory assumptions and the absence of canonical open-source codebases for exact hyperparameter replication, we report PACE as a supplementary comparison and focus our primary benchmark evaluations on direct rolling-summary baselines such as ACON.

E COMPLETE PROMPT TEMPLATES

This appendix provides the complete AppWorld history-compression prompts for the summary compression baseline, ACON-UT, ACON-UTCO, and ReBand. The templates are presented in the following order:

- **Base Prompting (Figure 5):** a general summary template covering reasoning, completed subtasks, and information needed to resume the task.
- **ACON-UT (Figure 6):** a quality-oriented template organizing essential state, completed actions, remaining work, and answer requirements into four sections.
- **ACON-UTCO (Figures 7–8):** a length-optimized template specifying compact state retention, optional sections, and omission of information that can be recovered when needed.
- **ReBand (Figures 9–10):** relevance-banded compression instructions specifying aggressive compression for Low-band chunks, concise factual summaries for Medium-band chunks, and structured operational facts for High-band chunks within a single rolling summary.

All four templates specify the task, previous summary, interaction history, and output format. ReBand and Shuffled-Band use exactly the same compression prompt and banded-specific compression rules. Shuffled-Band randomly permutes ReBand's relevance-guided band labels among chunks within each compression batch, preserving the number of chunks in each band. Each chunk therefore receives a Low, Medium, or High label through random reassignment rather than its own semantic relevance.

Base Prompting

General history compression

You are maintaining a structured context-aware summary for a productivity agent. You will be given the user instruction for the agent, a list of interactions corresponding to actions taken by the agent, and the most recent previous summary if one exists. Produce the following:

REASONING

Summarize key progress, decisions made, important observed outcomes, and rationale behind actions taken so far. Include how earlier steps influenced later ones and why certain data is retained in the summary.

COMPLETED

List completed subtasks or successful outcomes, with brief results if applicable.

[Information Source]

USER INSTRUCTION

{{ task }}

[PREVIOUS SUMMARY] (if any)

{{ prev_summary }}

[HISTORY OF INTERACTIONS]

{{ history }}

PRIORITIZE

1. Keep all sections relevant and concise.
2. Use reusable structured formats when summarizing artifacts.
3. Ensure agent can resume task with no loss of information.
4. Include key info from errors or failed attempts to prevent repeated mistakes.
5. Preserve all essential artifacts and data needed to complete the task.

[Output Format]

Do **not** include the input or any additional explanation. Only return the formatted summary.

AppWorld | History-compression user prompt | 1/1

Figure 5: The Base Prompting compression template.

ACON-UT

Quality-oriented unified compression guideline

You are maintaining a concise, stateful summary for a productivity agent. Generate a compressed history summary that enables seamless task resumption across sessions. Follow these rules strictly:

1. **Always include essential state.** Never produce an empty summary. Preserve:
   - Exact authentication tokens (e.g., `access_token='eyJ...'`)
   - Concrete data structures: lists of IDs (e.g., `target_song_ids=[1,5,6,...]` never numeric ranges), computed values, contact details, file paths, dates.
   - Completed irreversible actions: e.g., "Sent Venmo to alice@x.com for $45.50 - success".
2. **Structure the summary into the following sections:**
   - `### STATE` - all critical variables (tokens, IDs, amounts, etc.) in compact form. Use key = value assignments or tables.
   - `### COMPLETED` - list of actions already successfully executed with results.
   - `### TODO` - prioritized checklist of remaining sub-tasks, referencing stored state.
   - `### META` - whether the original task requires an answer. If it is a pure action (command, move, disable, etc.) output: "This task does NOT require an answer; call complete_task() without the answer argument." If it asks for information (e.g., pattern "What is…?", "How many…?") output: "Task requires an answer; call complete_task(answer = <computed value>)."
3. **Prevent re-discovery and re-authentication.**
   - If a token is already present (from previous summary or this session), do not re-login. Summaries must store the token exactly as obtained, never truncated.
   - Cache fetched API documentation and critical data (e.g., song IDs, contact lists) to avoid re-fetching; include them in STATE.
4. **Guard against instruction drift.** In META, explicitly remind the agent to check whether an answer is needed before calling complete_task. For action-only tasks, enforce the no-answer rule.
5. **Avoid verbosity and truncation.** Keep the summary compact but never cut off tokens, IDs, or computed numbers. Use concise notation (lists, sets). Do not use numeric ranges where IDs are non-sequential.
6. **Place session boundaries logically.** If a subtask is mid-execution, ensure the TODO section captures the exact next actions and remaining items, with all needed state.

[INFORMATION SOURCE]

USER INSTRUCTION

{{ task }}

[PREVIOUS SUMMARY] (if any)

{{ prev_summary }}

[HISTORY OF INTERACTIONS]

{{ history }}

[Output Format]

Output only the final summary. Do not include any extra commentary. The summary must follow the sections above.

AppWorld | History-compression user prompt | 1/1

Figure 6: The ACON-UT Prompting compression template.

ACON-UTCO

Length-optimized unified compression guideline

You are maintaining a minimal, compressed summary for a productivity agent. Generate the shortest possible history summary that enables seamless task resumption. Strictly adhere to these rules:

1. **Allow empty summary** - If the next session can resume solely from runtime variables and `{{ task }}` (no essential state to record), output an empty string. Never include information that will be re-derived or re-fetched.

2. **Essential state only**

- Keep exact authentication tokens **only** if they are directly used in the immediate next API calls and are not cheaply re-obtainable. Otherwise, replace with a login status (e.g., "logged into Spotify as …").
- Preserve concrete data: lists of IDs (always exact lists, never ranges), critical file paths, unique identifiers (playlist IDs, note IDs, amounts), and irreversible completed results (payment sent, note created).
- Drop computed values that will be recalculated, intermediate API doc outputs, null/empty variables, supervisor personal data that is not required later, and long enumerations when a short list or count suffices.

3. **Omit empty sections** - If STATE is empty, omit `### STATE`. If COMPLETED is fully captured by state or not needed for next steps, omit it. META may be merged into TODO when the final instruction is clear.

4. **STATE section** (if needed)

- Store only login result objects (e.g., `spotify_login = …`) or minimal key variables.
- Do not store passwords separately; include credentials only inside the login action result when absolutely required.
- Keep only tokens that appear in the immediate next tool calls.
- For file lists, provide a short enumeration (e.g., `internet_bill_files = ["2023-01.txt", "2023-02.txt", "2023-03.txt", "2023-04.txt"]`).

5. **COMPLETED section** (optional)

- List only irreversible actions whose outcome is not obvious from STATE (e.g., "Sent Venmo to … for $45.50").
- Drop preparatory steps (API exploration, password retrieval) unless their exact result is directly reused. Condense multiple items into one line.

6. **TODO section** - Merge with META into a single line that states the immediate next action and whether an answer is required.

- Example: `Next: fetch voice messages, then delete all messages from 5708520672; call complete_task() without answer.`
- Do not micro-plan; just give the next high-level objective. Do not repeat parameter details that STATE already holds.
- Drop TODO entirely if the task is complete in this session (summary may be empty).

7. **META handling**

- For action-only tasks, end the TODO line with `call complete_task() without answer`. No extra explanation.
- For tasks requiring an answer, append `call complete_task(answer=<value>)`. Keep it terse.
- If META is already implied, omit it.

8. **Token & credential rules**

- Replace long token strings with a reference to the login result object unless the token must be inserted verbatim in the next step.
- Drop ephemeral tokens if re-authentication is trivial.
- Remove tokens for apps not used in the next session.
- Never cache API documentation; the agent re-fetches as needed.

AppWorld | History-compression user prompt | 1/2

Figure 7: The ACON-UTCO compression template (page 1 of 2).

ACON-UTCO

Length-optimized unified compression guideline

9. **Avoid verbosity**
- Remove all code comments, natural language explanations, and error messages.
- Replace long lists of IDs with a concise enumeration (minimal set) or a count when IDs are not individually needed.
- Drop any formatting (e.g., full note titles after IDs) that is not critical.
- Use compact bullet points, never multi-paragraph prose.

10. **Session boundaries** - If mid-task, the TODO line must capture the exact next function call or step with all required identifiers from STATE.

[INFORMATION SOURCE]

USER INSTRUCTION

{{ task }}

[PREVIOUS SUMMARY] (if any)

{{ prev_summary }}

[HISTORY OF INTERACTIONS]

{{ history }}

[Output Format]

Output only the final summary. No extra commentary.

AppWorld | History-compression user prompt | 2/2

Figure 8: The ACON-UTCO compression template (page 2 of 2).

ReBand

Semantic-band-guided history compression

You are maintaining a minimal, compressed summary for a productivity agent. Generate the shortest possible history summary that enables seamless task resumption.

Each historical chunk below is wrapped in a `<CHUNK>` tag that carries its semantic_score and semantic_band. The semantic_score measures how similar the chunk is to the current task/runtime state; empirical analysis over this benchmark shows that the probability a chunk is needed later increases monotonically with this score. The numeric boundaries below are supplied by the selected experiment configuration:

- low-similarity chunks (semantic_score < {{ "%.2f"|format(low_threshold|default(0.53)) }}): use the lowest-retention band policy.
- medium-similarity chunks ({{ "%.2f"|format(low_threshold|default(0.53)) }} <= semantic_score < {{ "%.2f"|format(high_threshold|default(0.67)) }}): use the intermediate-retention band policy.
- high-similarity chunks (semantic_score >= {{ "%.2f"|format(high_threshold|default(0.67)) }}): use the highest-retention band policy.

Use the semantic_band shown on each chunk as the primary signal for how aggressively to compress it. The compression_plan below is a secondary reference only; the annotation on each `<CHUNK>` tag takes precedence.

Semantic band policy

- **low (semantic_score < 0.53)**: lowest dependency density. Aggressively compress or omit. Keep only a generic outcome if it provides useful background. Drop identifiers, parameter values, API response details, and long enumerations unless the compression_plan explicitly lists them under must_preserve.
- **medium (0.53 <= semantic_score < 0.67)**: moderate dependency density. Brief summary. Preserve key facts, concrete identifiers, dates, amounts, file references, and outcomes, but condense each chunk into one or a few compact statements. Do not retain full API responses or verbose observations.
- **high (semantic_score >= 0.67)**: highest dependency density and the main risk region for information loss. Do NOT summarize these chunks into flowing prose. Extract concrete operational facts as structured key-value entries and place them in STATE or an equivalent high-fidelity section. Preserve exact identifiers, parameter values, authentication states, file references, contact targets, amounts, playlist/note IDs, and any value that may be referenced in subsequent steps.

Core compression rules

1. **Allow empty summary** - If the next session can resume solely from runtime variables and the user instruction (no essential state to record), output an empty string. Never include information that will be re-derived or re-fetched.

2. **Essential state only**

- Keep exact authentication tokens **only** if they are directly used in the immediate next API calls and are not cheaply re-obtainable. Otherwise, replace with a login status (e.g., "logged into Spotify as …").
- Preserve concrete data: lists of IDs (always exact lists, never ranges), critical file paths, unique identifiers (playlist IDs, note IDs, amounts), and irreversible completed results (payment sent, note created).
- Drop computed values that will be recalculated, intermediate API doc outputs, null/empty variables, supervisor personal data that is not required later, and long enumerations when a short list or count suffices.

3. **Omit empty sections** - If STATE is empty, omit `### STATE`. If COMPLETED is fully captured by state or not needed for next steps, omit it. META may be merged into TODO when the final instruction is clear.

4. **STATE section** (if needed)

- Store only login result objects (e.g., `spotify_login = …`) or minimal key variables.
- Do not store passwords separately; include credentials only inside the login action result when absolutely required.
- Keep only tokens that appear in the immediate next tool calls.
- For file lists, provide a short enumeration (e.g., `internet_bill_files = ["2023-01.txt", "2023-02.txt", "2023-03.txt", "2023-04.txt"]`).
- High-band chunks should contribute their concrete values here as structured key-value entries.

AppWorld | History-compression user prompt | 1/2

Figure 9: The ReBand compression template (page 1 of 2).

ReBand

Semantic-band-guided history compression

5. **COMPLETED section** (optional)
- List only irreversible actions whose outcome is not obvious from STATE (e.g., "Sent Venmo to … for $45.50").
- Drop preparatory steps (API exploration, password retrieval) unless their exact result is directly reused. Condense multiple items into one line.

6. **TODO section** - Merge with META into a single line that states the immediate next action and whether an answer is required.
- Example: `Next: fetch voice messages, then delete all messages from 5708520672; call complete_task() without answer.`
- Do not micro-plan; just give the next high-level objective. Do not repeat parameter details that STATE already holds.
- Drop TODO entirely if the task is complete in this session (summary may be empty).

7. **META handling**
- For action-only tasks, end the TODO line with `call complete_task() without answer`. No extra explanation.
- For tasks requiring an answer, append `call complete_task(answer=<value>)`. Keep it terse.
- If META is already implied, omit it.

8. **Token & credential rules**
- Replace long token strings with a reference to the login result object unless the token must be inserted verbatim in the next step.
- Drop ephemeral tokens if re-authentication is trivial.
- Remove tokens for apps not used in the next session.
- Never cache API documentation; the agent re-fetches as needed.

9. **Avoid verbosity**
- Remove all code comments, natural language explanations, and error messages.
- Replace long lists of IDs with a concise enumeration (minimal set) or a count when IDs are not individually needed.
- Drop any formatting (e.g., full note titles after IDs) that is not critical.
- Use compact bullet points, never multi-paragraph prose.

10. **Session boundaries** - If mid-task, the TODO line must capture the exact next function call or step with all required identifiers from STATE.

[INFORMATION SOURCE]

USER INSTRUCTION

{{ task }}

PREVIOUS SUMMARY

{{ prev_summary }}

HISTORY OF INTERACTIONS

{{ history }}

SEMANTIC COMPRESSION PLAN

{{ compression_plan }}

[Output Format]

Output only the final summary. No extra commentary.

AppWorld | History-compression user prompt | 2/2

Figure 10: The ReBand compression template (page 2 of 2).

F Definition and Computation of Future-Value Reuse

We define *future-value reuse* as the reappearance of a concrete value from a historical interaction chunk in the agent's subsequent responses. For trajectory i, let $c_{i,j}$ denote a historical chunk available before decision step t, and let $\mathcal{V}(c_{i,j})$ be the set of values extracted from that chunk, such as identifiers, dates, amounts, and file paths. Let $A^{+}_{i,t}$ denote the concatenated agent responses from step t onward. The reuse label is

$$y_{i,j,t} = \mathbf{1}\left[\exists v \in \mathcal{V}(c_{i,j}) : \mathrm{match}(v, A^{+}_{i,t}) = 1\right], \tag{7}$$

where match checks for an exact substring occurrence. A chunk–query pair receives a positive label if at least one extracted value appears in the remaining agent responses. Multiple matches still contribute a single positive label, pairs with no extracted values receive a zero label.

Let $s_{i,j,t}$ denote the embedding cosine similarity between the historical chunk and the current query. For a similarity band B, define $\mathcal{D}_B = \{(i,j,t) : s_{i,j,t} \in B\}$. Its future-value reuse rate is

$$\widehat{\mathrm{FVR}}(B) = \frac{\sum_{(i,j,t)\in\mathcal{D}_B} y_{i,j,t}}{|\mathcal{D}_B|}, \qquad |\mathcal{D}_B| > 0. \tag{8}$$

The numerator counts positively labeled pairs, and the denominator counts all pairs in the band. Thus, this ratio equals the empirical conditional probability $\widehat{\Pr}(Y = 1 \mid S \in B)$. We multiply it by 100 when reporting percentages.

The statistic weights chunk–query pairs equally, allowing the same historical chunk to be evaluated at different decision steps. It measures observable textual value recurrence as a proxy for future utility.

G Case Analysis

As shown in Figure 11, ACON-UTCO and ReBand exhibit paired execution trajectories on the representative AppWorld task `2d9f728_1`: *"Split a $60 taxi fare and a $128 dinner bill equally among four participants (User, Jordan, Kathryn, and Nancy)"*. Both runs share the exact same model (GPT-4.1), seed, 4,096-token trigger threshold, and latest-turn preservation rule.

Steps 1–12 (Shared Setup): Both agents discover the environment apps, authenticate via Phone, inspect Venmo login credentials, and identify the three roommates (Jordan, Kathryn, and Nancy).

Compression 1 (Post-Step 12): When the first compression threshold is reached:

- **ACON-UTCO:** Under uniform compression, the compressor aggregates rooming information but drops Nancy's reciprocal $15 taxi-share request from the state, keeping only the requests for Jordan and Kathryn and the $32 dinner payment to Nancy.
- **ReBand:** Chunk-query semantic routing classifies the financial split calculations into the High-relevance band ($s_{i,t} \geq Q_{50}$), enforcing conservative structured compression. The complete participant set and individual payment parameters (taxi share = $15, food share = $32) are explicitly preserved in the rolling summary.

Steps 13–24 / 21 (Execution Preparation): Both agents log into Venmo, retrieve all three roommate email addresses, and inspect the transaction APIs. ReBand progresses more efficiently, creating taxi requests for Jordan and Kathryn by step 21, whereas ACON-UTCO spends additional turns re-inspecting endpoints.

Compression 2 (Decisive Boundary): At the second compression event:

- **ACON-UTCO:** The rolling summary consolidates Venmo tokens and roommate emails, but repeats the previous omission: Nancy's $15 taxi request remains completely omitted from the active state.
- **ReBand:** The compressor records Jordan's completed request and maintains Nancy's pending $15 taxi request as a preserved action item alongside the scheduled $32 dinner payment.

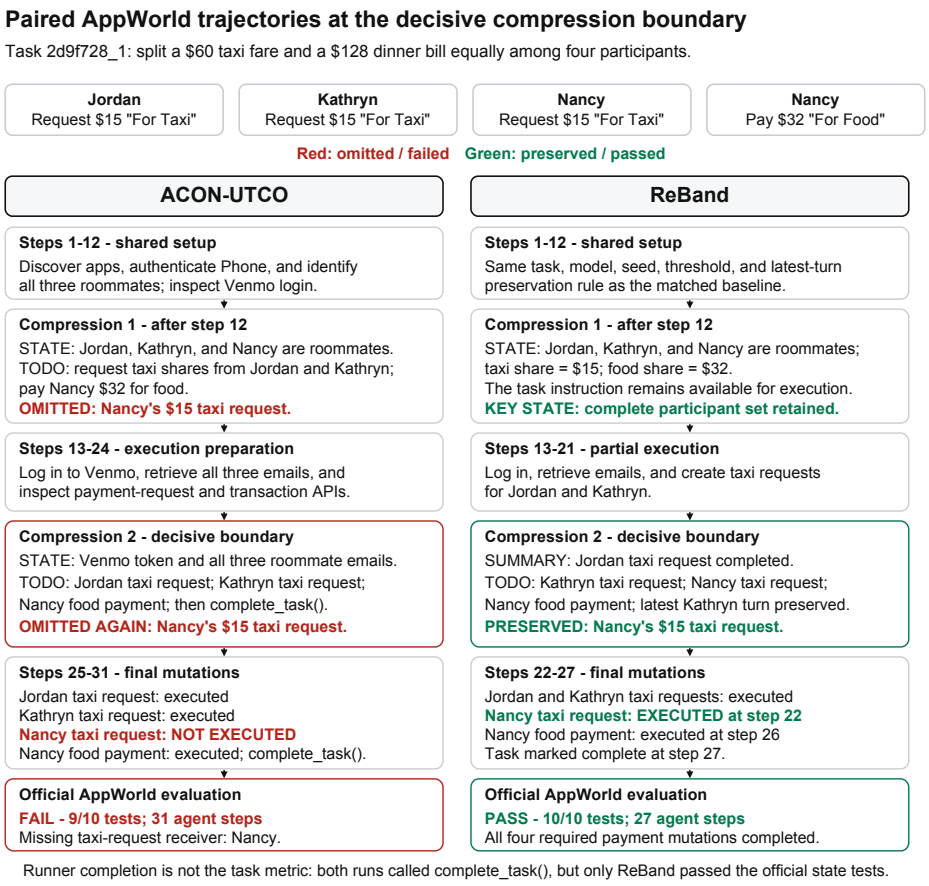


Figure 11: Paired AppWorld trajectories on task `2d9f728_1` comparing ACON-UTCO and ReBand. While ACON-UTCO repeatedly drops Nancy's $15 taxi-share request during uniform compression, ReBand's relevance-calibrated guidance preserves the complete participant action set, resulting in full test pass.

Final Mutations and Benchmark Evaluation:

- **ACON-UTCO:** In steps 25–31, the agent executes taxi requests for Jordan and Kathryn and pays Nancy $32. Lacking the omitted context, it never executes Nancy's $15 taxi request. Although the agent calls `complete_task()`, it fails the official AppWorld evaluation (9/10 tests passed; missing taxi-request receiver: Nancy).
- **ReBand:** In steps 22–27, the agent executes the Nancy taxi request at step 22, pays the $32 dinner expense at step 26, and successfully passes all 10/10 official state tests with 27 agent steps.

This comparison confirms that runner completion does not imply task success: both methods completed their trajectory loops, but only relevance-banded compression preserved the task-critical state dependencies required to pass the ground-truth environment mutations.