WHAT SURVIVES THE LONG HORIZON RUN? BENCHMARKING TASK STATE IN CODING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon agents may run for hours to weeks, requiring automated context compaction to fit constrained LLM context windows. However, such compaction risks corrupting or obscuring critical task state: objective-relevant facts and their provenance. To address this risk, we introduce EndureBench to evaluate language agents' preservation of critical task state across compaction cycles. It builds histories from recorded coding sessions toward a 400K-token context target and compares 15 compressors at fixed checkpoints under two schedules: fresh, which independently compacts each raw history prefix once, and rolling, which recursively updates the previous memory with the next interaction block. The results are consistent across compressors: 14 of 15 lose quality under rolling, and the failures are outright misses rather than partial answers—current-state correctness scores 0 on 61.0% of judgments and scope binding on 38.1%, against 11.3% for verification calibration. Overall quality hides this failure: compacted memory may omit objective-relevant facts and their provenance and should be verified before the agent resumes rather than treated as authoritative. EndureBench thereby isolates state preservation as a distinct object of measurement for long-horizon agents.

1 INTRODUCTION

Long-horizon coding agents may run for hours to weeks, accumulating histories that exceed constrained LLM context windows, so their harnesses automatically replace earlier interactions with compact text memories (Kang et al., 2026; Ye et al., 2026). Continuation depends on whether those memories preserve *task state*: source-scoped facts and their currently valid values, including stable constraints and revised decisions. A useful memory must retain not only which values appeared, but which value is current and which task it belongs to.

Compaction can preserve the relevant vocabulary while losing that distinction. Continuation therefore requires a memory to hold three things at once: (1) the value currently in force for each fact, rather than the set of values the history mentioned; (2) the source scope a fact belongs to, so that a similar choice made in another task is not read as a revision of this one; and (3) constraints that were never revised, which must survive rewriting instead of being replaced by plausible alternatives. Recall of what appeared in the history certifies none of the three.

Existing evaluations cannot certify them, for two reasons. First, functional probes of compressed coding histories (Factory Research, 2025) score the outcomes of agents running on compressed memory, so a compressor that drops a fact still passes whenever the agent recovers that fact while executing. Second, once agents act, different compressors produce different histories (Wu et al., 2025a; Hu et al., 2026; Zerhoudi et al., 2026), so an outcome difference cannot be charged to memory quality; supersession and scope are also measured in general settings whose task-state identity is not anchored to a recorded work history (Liotta, 2025), leaving scope binding unverifiable against a ground-truth source. Both obstacles disappear once the history is held fixed.

We introduce **EndureBench** (Fig. 1), which replays recorded coding histories constructed toward a 400K-token context target and scores compressed memory at fixed checkpoints. The evaluation is organized around three questions: **RQ1**, can a shared reader recover current values and source scope from compressed memory? **RQ2**, how do those scores differ between single-pass (fresh) and

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.

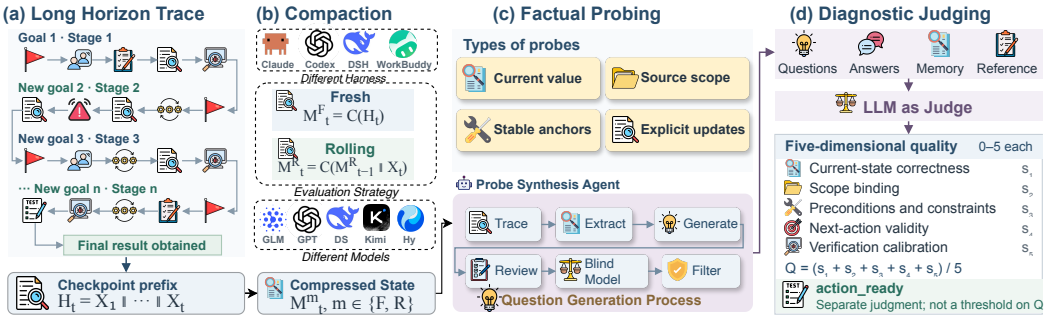

Figure 1: EndureBench checkpoint-replay protocol: fresh compacts each raw history prefix in one call, rolling rewrites the previous memory with the new block **(a–b)**; a shared reader answers source-scoped questions from memory alone, and a judge scores them against withheld references and shared rubric anchors **(c–d)**. Events and values are illustrative.

recursive compaction (rolling) on the same fixed histories? **RQ3**, does the same weakness persist across judging passes, history composition, readiness, and harness configuration?

Two deployable compaction schedules are compared on those histories. The *fresh* condition compresses each raw history prefix independently; the *rolling* condition compresses the previous memory together with the next interaction block. Recorded actions, observations, questions, and checkpoint boundaries are identical across every compressor and condition, references are withheld from the reader, and probe answers never re-enter compaction (Section 2), so compressed memory is the only quantity that varies. The contrast is between two schedules rather than against an uncompressed ceiling: rolling couples recursive rewriting with the loss of raw-prefix access, and fresh is itself one compression call applied to an ever-growing prefix.

If task state were no harder to preserve than preconditions, next-action validity, or verification calibration, measuring it separately would add little. It is harder. Across 15 compressors and 38 constructed histories, current-state correctness and scope binding are the two lowest of the five scored dimensions, and current-state correctness sits 1.30–2.05 points below the three action-discipline dimensions while ranking last in all 30 compressor–condition cells of a separate pooled rescoring. The schedule penalty is specific to the same two dimensions: rolling gives up as much as 0.54 points to fresh on current value and source scope, against 0.07 on verification calibration, and the composite gap favors fresh for 14 of the 15 compressors. These low factual means are not uniformly mediocre answers but outright misses: current-state correctness scores 0 on 61.0% of judgments and scope binding on 38.1%, against 11.3% for verification calibration. Neither pattern is visible in aggregate quality or readiness: a memory can be graded well on action discipline while the facts it reasons over are already wrong. Our contributions are:

- A fixed-history checkpoint-replay protocol that decouples memory evaluation from execution divergence by holding histories, interaction blocks, questions, and checkpoint schedules constant across compressors and conditions (Section 2).
- Main measurements of current-state correctness and scope binding for 15 compressors on 38 histories, establishing that these two dimensions are the lowest and the most schedule-sensitive, while readiness and aggregate quality obscure weak factual preservation (Section 4).
- Robustness diagnostics that reproduce the same dimension ordering under a second judging pass, an alternative compaction strategy, and four agent harnesses (Apps. B and F).

2 ENDUREBENCH CONSTRUCTION

EndureBench constructs long histories from recorded coding sessions and evaluates factual recovery at checkpoints fixed per history. Its reference contract asks for the value valid at each checkpoint and the task to which that value belongs.

Task state and checkpoint references. We represent task state as facts keyed by source scope, entity, and attribute. The checkpoint reference gives the currently valid value for that identity. A

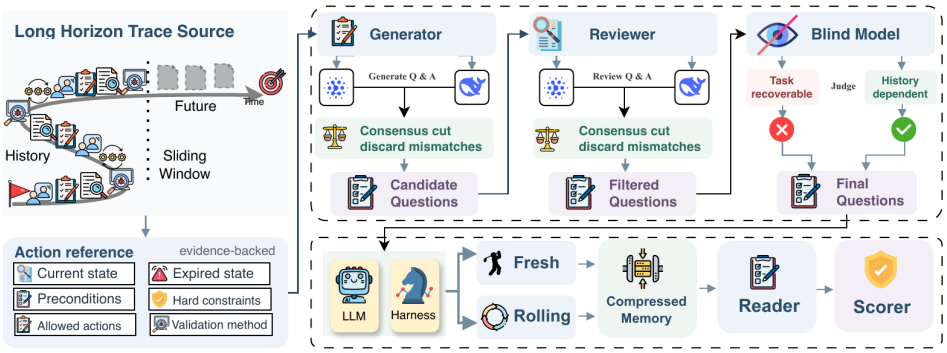

Figure 2: Building and filtering action probes: the strengthened contract separates current from stale state, synthesis and review discard mismatched, leaking, or underspecified candidates, and a blind no-history gate keeps only questions that need the recorded history. The release carries checkpoint-current factual references, not instance-level references for the three action dimensions.

change within the same scope creates a revision; a similar choice in another project remains a different fact. Stable anchors retain one value across scored checkpoints, whereas dynamic chains contain an explicit update. If an earlier value becomes valid again, the reference returns to that value. This operational definition excludes the agent's complete internal state, filesystem snapshot, and execution environment.

Checkpoints and compaction schedules. For constructed history i, let $H_{i,t} = X_{i,1} \| \cdots \| X_{i,t}$ be the raw prefix through checkpoint t, where each block ends at a valid message boundary. One checkpoint falls on each boundary, so the count T_i follows the realized length of history i; the boundaries are computed once per history and reused for every compressor, condition, and judging pass. For compressor configuration c, the two evaluated memories are

$$M^{\mathrm{F}}_{i,t} = C_c(H_{i,t}), \tag{1}$$

$$M^{\mathrm{R}}_{i,t} = C_c(M^{\mathrm{R}}_{i,t-1} \| X_{i,t}), \tag{2}$$

with $M^{\mathrm{R}}_{i,0} = \varnothing$. The fresh condition independently compacts each raw prefix once. The rolling condition recursively carries the previous memory forward. Both use the same prompt, generation settings, questions, and checkpoint schedule. Their contrast is the contrast between two deployable schedules: rolling couples recursive rewriting with the reduced input access and shorter compaction calls that make it attractive in practice, and fresh pays for its raw prefix with a compaction input that grows without bound. Fixing $H_{i,t}$, the question set, and the checkpoint schedule identically across all compressor–condition cells is the necessary condition for making compressed memory the sole variable; any divergence in the recorded trajectory would reintroduce a confound that memory quality alone cannot account for.

With a 400K-token target and approximately 150K-token blocks (Tab. 5), the release scores 115 stages over 38 histories (mean 3.03; indices 1–3 homogeneous, 1–4 heterogeneous), and those same 115 recur in all 30 compressor–condition cells: $115/38$ is non-integral because T_i tracks length, not because cells differ (App. A.1). At each checkpoint, the shared reader receives the memory and a question or question batch; each answer is scored separately. The judge receives the answer, memory context, and withheld checkpoint-current factual reference. Probe answers never enter the recorded history or any later compaction call. We record the generation settings in Tab. 5.

History construction. We construct histories from coding sessions in DeepSWE v1.1. We remove source system messages, merge tool definitions, and concatenate session bodies into blocks that end at legal message boundaries; a short tail may be merged. Homogeneous histories contain one coarse task type, while heterogeneous histories contain several. We exclude duplicate task IDs within each constructed history. Concatenation creates a longer memory problem, but no cross-task dependencies or shared filesystem. The construction yields 38 reported histories, combining 2–25 source sessions with a 400K-token target and approximately 150K-token blocks; these lengths are configuration targets. App. A, Tab. 5, records the source filters, extraction windows, and runtime settings.

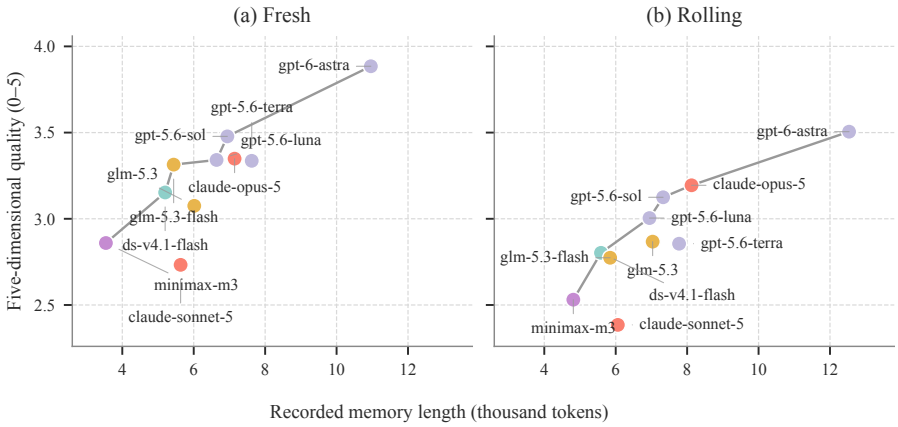

Figure 3: Realized mean memory length versus five-dimensional quality (0–5) for the 10 labels with token records, **(a)** fresh and **(b)** rolling, coloured by model family. Lengths are recorded outputs, not the condition-dependent retention percentage (Section 3.2); nondominated sets are computed within each condition.

Versioned facts. Dual-model extraction and adjudication recover current and historical values, supporting quotations, checkpoint indices, and update indicators. By convention, a fact remains valid when it is not mentioned again. A stable anchor must retain one scoreable value from its first scored checkpoint to the last; a dynamic chain must contain an explicit update. Quotation matching verifies that evidence occurs in the history, while identity collisions, effective updates, and restored historical values are resolved by semantic review.

Questions and evaluation status. Questions are generated from fact chains, with prefix-validity checks removing probes that leak a future value. The retained main-result release supplies checkpoint-bounded factual references for source scope and current versus stale state, which ground the two dimensions we report as primary evidence; the three action dimensions and `action_ready` are scored from the shared rubric anchors applied to the recovered state. We record the target contract in Fig. 2. The reader sees only memory and questions, and neither references nor answers enter later compaction. Evaluation calls may batch up to five questions per call.

Horizon coordinates. History length, recursive exposure, fact age, and revision count impose different demands on task state. A longer context stresses compaction capacity without implying more revisions; multiple rolling rounds accumulate recursive exposure independently of elapsed tokens; and a fact introduced early in the history behaves differently from one introduced just before the scored checkpoint. Separating these dimensions is necessary to avoid conflating degradation effects that have distinct causes: memory omission, reader extraction failure, and identity collision each leave different signatures across the four coordinates. App. G defines the checkpoint-resolved measurements and a focused validation plan for attributing errors to their source.

3 Evaluation Measures

3.1 Primary factual scores and supplementary diagnostics

Table 1: Reported dimension means (0–5). F = fresh; R = rolling. F−R uses rounded values.

Dimension	F	R	F−R
Current-state correctness	1.93	1.55	0.38
Scope binding	2.76	2.22	0.54
Preconditions and constraints	3.23	2.98	0.25
Next-action validity	3.50	3.27	0.23
Verification calibration	3.67	3.60	0.07

Table 2: Reported counts for the 30 compressor–condition cells. Cond. = cells; stages = checkpoint records; probes = scored-probe entries.

Group	Cond.	Stages	Probes
Per cell	1	115	2,782
Fresh (F)	15	1,725	41,730
Rolling (R)	15	1,725	41,730
Both conditions	30	3,450	83,460

Table 3: All 15 compressors (38/38 histories per condition), sorted by rolling quality Q_R. F = fresh; R = rolling; Q: five-dimensional quality (0–5); ready: judge-assigned readiness (%); $\Delta = Q_F - Q_R$: schedule gap. Shading is normalized within each level column and is not comparable across columns; Δ is unshaded.

	Quality			Ready (%)	
Compressor label	Q_R	Q_F	Δ	R	F
`gpt-6-astra`	3.505	3.885	0.379	45.26	49.17
`claude-opus-5`	3.194	3.349	0.154	32.93	30.73
`gpt-5.6-sol`	3.125	3.478	0.353	32.53	36.99
`gpt-5.6-luna`	3.004	3.341	0.337	32.53	33.25
`glm-5.3`	2.867	3.075	0.208	24.98	26.49
`gpt-5.6-terra`	2.855	3.336	0.481	25.66	38.86
`deepseek-v4.1-flash`	2.803	3.153	0.350	21.39	33.00
`glm-5.3-flash`	2.774	3.314	0.540	23.36	33.75
`deepseek-v4-flash`	2.662	3.169	0.507	23.65	28.90
`minimax-m3`	2.531	2.859	0.329	22.03	23.33
`glm-5.2`	2.493	2.549	0.056	18.12	12.98
`claude-sonnet-5`	2.385	2.733	0.348	14.34	22.03
`kimi-k3`	2.305	2.482	0.177	13.48	13.26
`deepseek-v4-pro`	2.285	2.200	−0.085	16.00	13.41
`hy4-preview`	2.063	2.343	0.280	6.22	15.10
Equal-model mean	2.723	3.018	0.295	23.50	27.42

Our primary outcomes are *current-state correctness* (CSC), which scores recovery of the checkpoint-current value, and *scope binding*, which scores attribution to the correct source or task. Both use the reported 0–5 rubric and are discussed separately as *task-state fidelity*, without a new two-dimensional composite; the existing five-dimensional, equally weighted mean is retained as the supplementary *task-state quality* score. *Preconditions and constraints*, *next-action validity*, and *verification calibration* record whether the recovered task state supports a defensible next step. We report the reference-grounded pair as primary evidence and the action dimensions as the accompanying discipline profile. The five-dimensional mean is a composite answer-quality score rather than the fraction of state retained. A separate boolean judgment, `action_ready`, records whether the judge endorses proceeding.

We report the dimension summaries for the evaluated histories in Tab. 1. They are read per dimension rather than as a decomposition of the equal-model quality mean. Let $\mathcal{O}_{cm}$ index the reported scored observations for compressor c and condition m, with $N_{cm} = |\mathcal{O}_{cm}|$. Writing $s_{ocmd} \in [0, 5]$ for dimension d and $b_{ocm} \in \{0, 1\}$ for the judge-assigned `action_ready` decision, the aggregate quantities are

$$Q_{cm} = \frac{1}{N_{cm}} \sum_{o \in \mathcal{O}_{cm}} \frac{1}{5} \sum_{d=1}^{5} s_{ocmd}, \tag{3}$$

$$A_{cm} = \frac{100}{N_{cm}} \sum_{o \in \mathcal{O}_{cm}} b_{ocm}. \tag{4}$$

A_{cm} is expressed as a percentage and is not obtained by thresholding Q_{cm}.

Schedule gap. Evaluating each compressor under both schedules also yields within-compressor contrasts: $\Delta_c = Q_{c,\mathrm{F}} - Q_{c,\mathrm{R}}, \qquad \Delta A_c = A_{c,\mathrm{F}} - A_{c,\mathrm{R}}$. We call Δ_c the *schedule gap* of compressor c: the answer-quality loss when task state must survive through the compressor's previous output. We report it beside $Q_{c,\mathrm{R}}$ because a small gap cannot compensate for poor rolling quality, while a strong single-pass compressor may still be fragile under recursion.

Judging-pass sensitivity. Two independent scoring passes over the same answers calibrate the stability of cell-level composite quality: absolute changes average 0.0916 and reach 0.2206, while cell rankings correlate highly across passes (Section B.1). Current-state correctness has a cross-pass rank correlation of 0.992, supporting the ordering behind the primary factual result. Within-compressor readiness differences are comparable to the composite-quality movement, so we report per-compressor readiness as an aggregate diagnostic; the dimension-level gaps are interpreted from their consistent direction rather than from a direct gap-by-gap rescoring bound.

Score distributions. In the separate pooled rescoring summary, 96.1%–99.8% of sub-item scores fall on 0, 3, or 5 (Section B.2). We report the shares scored 0 and 5 alongside means to show this concentration (Fig. 5): the low factual means come from answers that miss the current value outright, not from uniformly middling answers. Current-state correctness scores 0 on 61.0% of judgments and scope binding on 38.1%, against 11.3% for verification calibration. Given this concentration, we compare well-separated levels rather than adjacent compressor means.

What the measures support. The retained references cover checkpoint-current versus stale values and source scope, so current-state correctness and scope binding are graded against withheld ground truth, while the three action dimensions and `action_ready` are graded from the shared rubric anchors. The measures score recovery from memory; they do not score restored environments, continuation feasibility, or downstream execution.

Evaluation configuration and evidence. The analysis uses saved probe answers produced under a dual-model scoring protocol: for both reader and judge roles, two models score each response independently; when both agree the decision is adopted directly, and a higher-capability model arbitrates disagreements. The primary reported reader is `gpt-5.5` and the primary judge is `glm-5.2`. App. A.1 documents the provenance of these records and the aggregation level at which each reported quantity is defined; full settings and source filters are in Tab. 5 (App. A).

3.2 Aggregation and coverage

The same 15 compressor labels appear in both conditions; we denote the data field `single_shot` as fresh. Each of the 30 compressor–condition cells reports the same 115 stages and 2,782 scored-probe entries (Tab. 2), the identity our contrasts require; the total is not 38×3 because block counts follow history length (Section 2). The retained release does not carry per-history block counts, so we aggregate over checkpoints rather than comparing checkpoint indices.

We use equal-model averages and within-compressor differences, so each compressor has equal influence, while histories and distinct facts are not separately reweighted within a cell. All 30 cells report full history coverage. Because repeated probes of the same facts and related model configurations are not independent samples, we support claims by consistency across cells, conditions, judge passes, and history arms rather than by significance tests (App. A.1). For the model set $\mathcal{C}$, the condition-level aggregates are

$$\bar{Q}_m = \frac{1}{15} \sum_{c \in \mathcal{C}} Q_{cm}, \qquad \bar{A}_m = \frac{1}{15} \sum_{c \in \mathcal{C}} A_{cm}. \tag{5}$$

Why retention is not used as a coordinate. The released summaries define *retention* as total memory-output tokens divided by total compaction-input tokens. For the 10 covered compressors, mean input per call is 298,923–299,132 tokens under fresh and 146,865–151,895 under rolling (App. A.2), because the two schedules feed different inputs: a growing raw prefix versus a new block plus the previous memory. Retention therefore mixes a condition-specific denominator into the ratio and is not a condition-invariant memory-size coordinate. We therefore use realized memory length as the length coordinate throughout.

4 Results and Analysis

We evaluate 15 compressors on 38 constructed histories. RQ1 concerns current-value and source-scope recovery, RQ2 compares the two compaction schedules, and RQ3 asks whether the same weakness recurs in independent diagnostics. The primary outcomes are the factual dimension scores in Tab. 1 over 83,460 scored entries; composite quality and readiness are supplementary. Separate rescoring, anchor, and harness diagnostics are identified below and are reported alongside rather than pooled with the main release.

RQ1: Can current values and source scope be recovered? Current-state correctness (CSC) averages 1.93/5 under fresh and 1.55/5 under rolling; scope binding averages 2.76/5 and 2.22/5 (Tab. 1). These are the two lowest dimensions in both conditions. The weakness is therefore visible

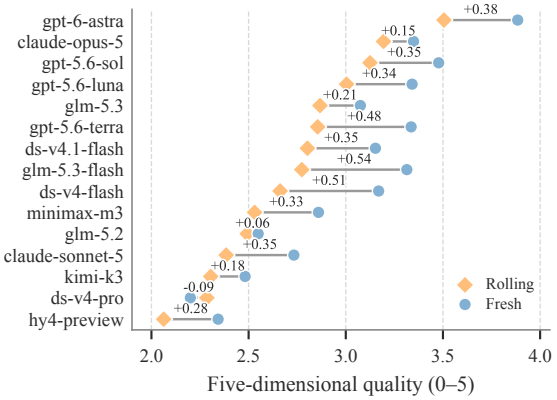

Figure 4: Compressor quality under both schedules, ordered by rolling quality; connectors show $\Delta = Q_F - Q_R$, positive when recursive compaction scores lower.

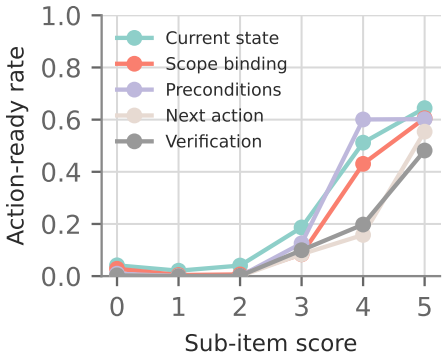

Figure 8: Readiness conditional on each sub-item score, pooled over 170,896 rescored judgments; other dimensions are not held fixed, so the curves show association, not a decision rule.

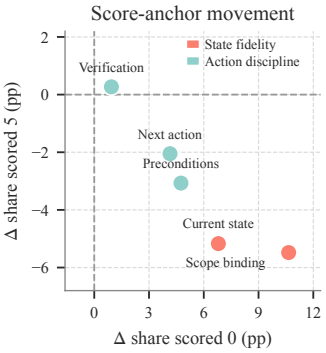

Figure 5: Score-anchor movement by dimension. Colours distinguish state-fidelity from action-discipline dimensions.

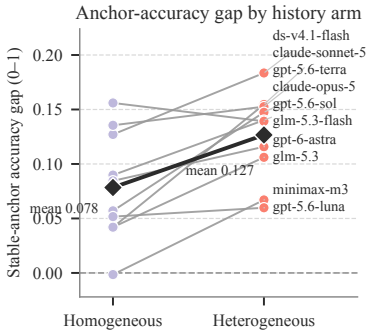

Figure 6: Fresh-minus-rolling stable-anchor accuracy gap by history arm. Diamonds show arm means.

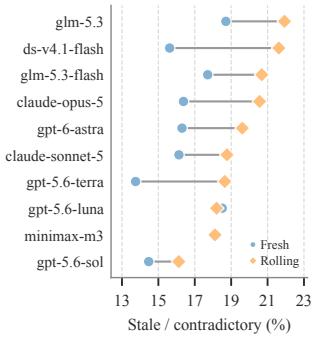

Figure 7: Heterogeneous-arm stale-or-contradictory answer rate, the answer-rate measure of cross-task misattribution.

even when each checkpoint is compressed independently from its raw history prefix. The reader must recover both the value that is currently valid and the task to which it belongs; a plausible value from another source does not answer the target fact.

CSC is lower than scope binding under both schedules, and both are rubric levels rather than percentages of facts recovered. The practical reading is that the best compressor in this population, `gpt-6-astra`, still scores 3.18/5 on current state in the easier schedule, while the median label sits at 1.95 and six labels fall below 1.5: on these histories no evaluated compressor supplies task state a continuation could rely on without rechecking the repository. We keep compressor-level composite quality and readiness for context in Tab. 3, where the same labels look considerably healthier: nine of 15 compressors exceed 3/5 in the fresh condition.

We resolve the same five dimensions by compressor in a separate pooled rescoring of 170,896 judgments (Tab. 4). CSC is the lowest of the five dimensions in all 30 compressor–condition cells, so the aggregate ordering is not produced by a few weak labels. The dimensions also differ in how much they separate compressors: CSC spans 2.40 points across the 15 fresh cells and 2.23 under rolling, whereas verification calibration spans 0.89 and 0.71. A compressor is therefore distinguished mainly by whether it preserves the currently valid value, not by how it hedges about checking: the strongest and weakest compressors are pulled far more sharply apart on current state than on any discipline dimension, and this ordering holds in both schedules. This pool draws on a different population and weighting from Tab. 1, so we keep the two rescorings separate; that they agree on which dimension

Table 4: Five rubric dimensions (0–5) and judge-assigned readiness by compressor, from a separate pooled rescoring of 170,896 judgments sorted by the rolling five-dimensional mean. F = fresh; R = rolling; CSC = current-state correctness. Cell sizes (3,953–6,358 answers) are not matched, and shading is normalized per column, not comparable across columns.

	CSC		Scope		Precond.		Next act.		Verif.		Ready (%)	
Compressor	F	R	F	R	F	R	F	R	F	R	F	R
gpt-6-astra	3.18	2.73	3.57	3.20	3.88	3.54	4.00	3.74	3.95	3.79	46.7	43.0
gpt-5.6-sol	2.50	2.16	3.30	2.92	3.45	3.21	3.53	3.36	3.58	3.47	33.5	28.0
claude-opus-5	2.50	2.11	3.12	2.66	3.31	3.20	3.49	3.38	3.61	3.49	31.7	28.1
gpt-5.6-luna	2.41	2.14	3.27	2.78	3.43	3.05	3.54	3.19	3.59	3.47	31.6	29.4
gpt-5.6-terra	2.41	1.84	3.24	2.45	3.49	3.10	3.51	3.26	3.63	3.57	35.5	25.1
glm-5.3	1.86	1.54	2.67	2.18	3.15	2.95	3.32	3.17	3.52	3.50	21.1	20.3
ds-v4.1-flash	2.10	1.63	2.70	2.23	3.12	2.84	3.28	3.05	3.48	3.41	27.1	19.2
glm-5.3-flash	1.95	1.53	2.84	2.23	3.11	2.87	3.25	3.08	3.40	3.43	26.6	21.0
ds-v4-flash	2.15	1.62	2.84	2.04	3.16	2.73	3.23	2.95	3.38	3.38	25.5	19.8
glm-5.2	1.31	1.31	2.18	2.10	2.83	2.75	3.04	2.98	3.35	3.25	14.8	17.0
claude-sonnet-5	1.46	1.22	2.22	1.93	2.61	2.63	2.90	2.82	3.16	3.35	17.8	13.4
minimax-m3	1.48	1.34	2.38	2.01	2.81	2.51	3.06	2.80	3.28	3.22	18.8	17.5
kimi-k3	0.90	0.72	2.26	1.93	2.65	2.54	2.94	2.87	3.22	3.28	13.8	13.7
ds-v4-pro	0.88	0.94	1.69	1.66	2.27	2.50	2.58	2.75	3.06	3.28	10.7	15.0
hy4-preview	0.78	0.50	2.17	1.55	2.39	2.20	2.75	2.58	3.14	3.09	13.4	8.0

is weakest and on the sign of the aggregate schedule gap is what makes the finding robust to the choice of pool.

RQ2: How do the compaction schedules differ? On fixed recorded histories, fresh exceeds rolling by 0.38 points on CSC and 0.54 on scope binding (Tab. 1). These gaps are larger than those for preconditions and constraints (0.25), next-action validity (0.23), and verification calibration (0.07). The absolute level is lowest for current values, while source scope has the largest schedule gap; the two state dimensions thus carry both the lowest scores and the widest schedule separation, whereas the three discipline dimensions move far less between conditions. Reporting only the five-dimensional mean would blur this structure and hide the distinction.

The comparison holds recorded actions, observations, probes, and checkpoint boundaries fixed, so the gap is a property of the schedule as deployed: recursive rewriting and the reduced access to the raw prefix arrive together, and a system that adopts rolling adopts both. One length-based explanation can be ruled out. Rolling memories are $1.02\times$–$1.36\times$ *longer* for the 10 labels with length data (Fig. 3), so the lower rolling scores are not explained by shorter realized memories. The corresponding compressor-level composite gaps are positive for 14 of 15 labels (deepseek-v4-pro is the exception at $\Delta = -0.085$; Fig. 4).

RQ3: Does the same weakness appear in independent diagnostics? The factual weakness is not an artifact of one scoring run. It reproduces across a second judging pass, a separately pooled rescoring of 170,896 judgments, and both history arms; across four harness configurations, current-state correctness remains the weakest dimension in all 32 cells. What the aggregates do not give is the checkpoint at which a value was lost or a separation of memory omission from reader extraction failure; a source-linked, human-checked subset with an evidence-only reader control is the targeted next step (App. G.1).

Separate factual diagnostics sharpen the picture without entering the main score table. Mean fresh-minus-rolling stable-anchor accuracy gaps are 0.078 in homogeneous and 0.127 in heterogeneous history arms (Fig. 6): even facts that never change are lost more often under recursion, and mixing tasks in one history increases that loss by about 1.6 times. In the heterogeneous arm, stale-or-contradictory answer rates are higher under rolling for 8 of the 10 covered compressors (Fig. 7), which is the pattern expected if recursion erodes source scope rather than merely dropping text. These diagnostics use another judge and cover 10 labels; the stale-or-contradictory rate is the answer-rate measure of cross-task misattribution in the heterogeneous arm.

Readiness behaves as an auxiliary judgment rather than a continuation outcome, and conditioning it on each sub-item score identifies what gates it (Fig. 8). Current-state correctness is the strongest of the five correlates of readiness (Spearman 0.614 versus 0.448 for verification calibration): 4.1%

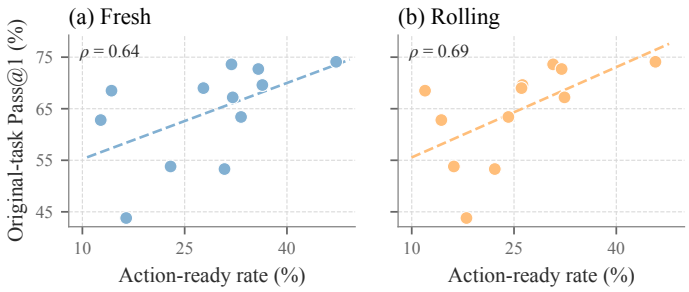

Figure 9: Model-level association between judge-assigned action-ready rate and original-task Pass@1 from the DeepSWE leaderboard, for the 12 labels matched before deduplication (Spearman $\rho = 0.61$, $p = 0.037$ fresh; $\rho = 0.54$, $p = 0.070$ rolling); conditions plotted separately, dashed least-squares guides.

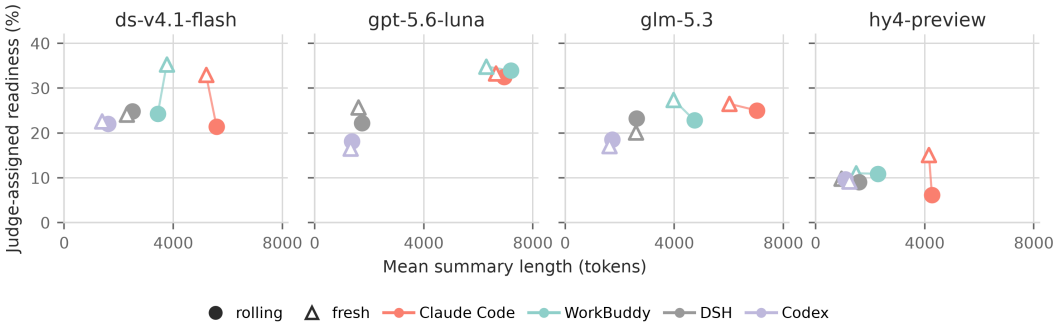

Figure 10: Mean summary length and judge-assigned readiness per compressor under four harness configurations. The harnesses differ in several settings at once, so the spread bounds harness sensitivity rather than isolating one prompt change.

of answers scoring 0 on current state are judged ready, against 64.4% of those scoring 5. Readiness also tracks external ability at the model level (Fig. 9) and shifts with harness configuration (Fig. 10), so we use it as a coarse screening diagnostic rather than a continuation outcome.

5 Conclusion

EndureBench measures checkpoint-current factual recovery while holding actions, probes, and checkpoint boundaries fixed across schedules. Current value and source scope are the weakest capabilities of compressed memory: current-state correctness ranks last in all 30 compressor–condition cells, and no compressor exceeds 3.18/5. Rolling lowers quality for 14 of 15 compressors; its largest gaps occur on scope binding and current state, although its memories are 1.02–1.36× longer for the 10 compressors with token records. Separately, 96.1–99.8% of sub-item scores fall on the rubric anchors 0, 3, or 5 rather than across partial scores. Composite quality and readiness can therefore obscure weak factual recovery: compacted memory should be verified before an agent resumes, not treated as authoritative. Checkpoint-resolved and evidence-only controls (Apps. G.1 and G) can next locate when facts are lost and distinguish memory omission from reader failure.

AI use statement

The research idea and its execution are the authors' own: the formulation of task state and the failure modes at issue, the fixed-history replay protocol and its controls, and the choice of measures. Generative AI was used in a supporting role only.

Generative AI is, however, a component of the benchmark itself rather than merely an authoring aid. As described in Sections 2 and 3, LLMs perform fact extraction, probe generation, probe validation, probe answering, and rubric judging, and the 15 evaluated compressors are themselves LLMs under

test. Probe items and reference answers pass the automated validation stage of Section 2, and rubric scores follow the dual-model agreement-and-arbitration protocol recorded in Tab. 5, so no single AI judgment is adopted unchecked.

As an authoring aid, we used generative AI mainly for editing and readability, and additionally for literature discovery and screening, bibliography preparation, implementation support for the construction and analysis code, and figure preparation. We reconstructed Fig. 1 as an editable vector schematic from an AI-generated layout candidate; its displayed task-state values are illustrative rather than measured, as its caption states. We did not use generative AI for translation, since the manuscript was drafted in English throughout, and we did not use it to produce, impute, or adjust any reported measurement: every number in the paper is computed by the retained analysis scripts from the retained aggregate reports. Questionnaire design and transcription of research recordings are not applicable, as this work involves no human subjects.

We have reviewed all AI-assisted work: AI-assisted code was read and its outputs cross-checked against the retained aggregate reports, with figure and reference consistency additionally verified by the repository check scripts; every bibliography entry was checked against its publication record; and AI-edited prose was checked sentence by sentence against the retained results, with the evidence boundaries stated in Section 5 constraining which claims the aggregates support. We take responsibility for the final content of this work.

Ethics statement

Coding trajectories may contain credentials, personal information, proprietary code, or confidential decisions. Releasing derived instances requires appropriate permissions, source-license review, and evidence-preserving de-identification that retains entity identity, scope, and revision relations. This manuscript releases neither raw trajectories nor credentials and does not claim institutional approval or verified permission for a new dataset release. These requirements must be met before publishing instances or conducting additional evaluations on restricted histories.

Reproducibility statement

Sections 2–3 define the replay protocol, construction, and measures; Tab. 5 records source and configuration details. Tabs. 3 and 1 provide the compressor summaries and dimension means used to interpret Section 4. Figs. 4 and 5 can be redrawn from the retained main-release and pooled-dimension aggregate summaries. The aggregate reports supporting Tab. 3 are retained as well, and every reported quantity is defined and checkable against them. App. A.1 documents the granularity of the retained reports, and App. G lists the minimum metadata needed for checkpoint-resolved replay.

An anonymized code supplement accompanies this submission. It contains the six-step construction pipeline, the two compaction schedules, the shared reader, the rubric judge and readiness prompts, and the pipeline configuration that produced the reported results, so that the protocol of Sections 2–3 and the scoring instrument recorded in Tab. 5 can be read and re-run against histories supplied by the reader. Consistent with the Ethics statement, the supplement ships neither recorded trajectories nor credentials, and running it requires an OpenAI-compatible endpoint and model access that the user must provide; it therefore supports inspection and re-execution of the protocol rather than constituting a verified independent replication of the reported numbers.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Factory Research. Evaluating context compression for AI agents. Factory research blog, December 2025. URL https://factory.com/news/evaluating-compression. Published 2025-12-16. Accessed 2026-09-26.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? arXiv preprint, April 2024. URL https://arxiv.org/abs/2404.06654v3. arXiv:2404.06654v3. First submitted 2024-04-09; revised 2024-08-06. Accepted at COLM 2024.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fd1eff9dd295df50a41f2521942fa31d-Abstract-Conference.html.

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 *International Conference on Machine Learning*, 2026. URL https://www.microsoft.com/en-us/research/publication/acon-optimizing-context-compression-for-long-horizon-llm-agents/. ICML 2026; publication metadata from Microsoft Research.

Matt Liotta. Beyond conversation: A state-based context architecture for enterprise AI agents. Parslee technical report, December 2025. URL https://parslee-ai.github.io/statebench/state-based-context-architecture.pdf. Date as printed on the official PDF. Accessed 2026-09-13.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schuetze. NoLiMa: Long-context evaluation beyond literal matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 44554–44570. PMLR, July 2025. URL https://proceedings.mlr.press/v267/modarressi25a.html.

Charles O'Neill. Repeated KV cache for long-running agents. Baseten Research blog, March 2026. URL https://www.baseten.co/research/repeated-kv-cache-for-long-running-agents/. Accessed 2026-09-13.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. arXiv preprint, October 2023. URL https://arxiv.org/abs/2310.08560v2. arXiv:2310.08560v2. First submitted 2023-10-12; revised 2024-02-12.

Parslee. StateBench. Official project documentation, August 2026. URL https://github.com/Parslee-ai/statebench. Scoring correction notice dated 2026-08-05. Accessed 2026-09-13.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. In *Advances in Neural Information Processing Systems*, volume 36, pp. 74530–74543. Curran Associates, Inc., 2023. doi: 10.5555/3666122.3669381. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ebd82705f44793b6f9ade5a669d0f0bf-Abstract.html.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html.

Di Wu, Zixiang Ji, Asmi Kawatkar, Bryan Kwan, Jia-Chen Gu, Nanyun Peng, and Kai-Wei Chang. LongMemEval-V2: Evaluating long-term agent memory toward experienced colleagues. arXiv preprint, May 2026. URL https://arxiv.org/abs/2605.12493v1. arXiv:2605.12493v1, submitted 2026-05-12. Work in progress.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu 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. arXiv preprint, September 2025b. URL https://arxiv.org/abs/2509.13313v3. arXiv:2509.13313v3. First submitted 2025-09-16; revised 2026-03-26.

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 *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fa948624dfde013671e72c1a7ca4aebc-Abstract-Conference.html.

Weihao Zeng, Yuzhen Huang, and Junxian He. LOCA-bench: Benchmarking language agents under controllable and extreme context growth. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64486. Poster presentation; official proceedings metadata not yet located.

Saber Zerhoudi, Jelena Mitrovic, and Michael Granitzer. The compaction cliff in long-running AI agent memory. arXiv preprint, August 2026. URL https://arxiv.org/abs/2608.22752v1. arXiv:2608.22752v1, submitted 2026-08-24.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Liang. MEM1: Learning to synergize memory and reasoning for efficient long-horizon agents. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5fc8b3bdfbb9167b5144df5d3fae4616-Abstract-Conference.html.

Table 5: Construction and evaluation settings. Lengths are configuration targets, not empirical averages; model identities are reported gateway labels.

Component	Setting
History construction	
Source filters	DeepSWE v1.1; at least 15K tokens, six turns, four tool calls
History composition	2–25 sessions; 400K-token target; approximately 150K-token blocks
Arms	20 homogeneous and 18 heterogeneous histories
Fact extraction	50K-token windows; 4K overlap; dual extraction and adjudication
Evaluation configuration	
Release coverage	38 histories; 15 compressors; two conditions; 38/38 histories in each cell
Checkpoint schedule	One checkpoint per history block; 115 scored stages per cell (mean 3.03 per history); scored block indices 1–3 homogeneous, 1–4 heterogeneous
Compaction	Shared `claude_code` prompt; no explicit summary budget; 16K generation cap
Reader / judges	Dual-model scoring protocol: two models judge independently; agreement is adopted directly, disagreements are arbitrated by a higher-capability model. Primary reader: `gpt-5.5`; primary five-dimensional judge and readiness: `glm-5.2`; four-class anchor diagnostic: `claude-sonnet-5`

A Construction and Evaluation Settings

We distinguish configuration targets from release coverage in Tab. 5. DeepSWE v1.1 supplies the source sessions from which EndureBench constructs histories to evaluate task state. Concatenation lengthens the recorded history without creating shared environments or dependencies across source tasks.

Terminology in the appendices. Some appendix artifacts predate the current terminology. Recorded field names, table rows, and reproduced figure panels therefore retain `fresh` and `rolling`. Throughout the paper, these terms map exactly to the fresh and rolling schedules in Section 2.

A.1 Evidence scope and coverage

The retained reports define the main results at the cell and dimension levels. All 30 compressor–condition cells cover the same 38 histories, 115 stages, and 2,782 scored probes; between-condition contrasts therefore use cell means. Checkpoint boundaries are computed once per history and reused across compressors, conditions, and judging passes, while the number of checkpoints follows realized history length (Section 2). Token and arm diagnostics cover 10 of the 15 released compressors, so claims based on those records are restricted to that subset. An earlier stratified report covering 12 configurations under a six-dimensional protocol is retained in the code supplement for audit traceability; its results are not pooled with the current five-dimensional release.

A.2 Realized memory length

We report mean memory length for the 10 compressors with token records in Tab. 6, averaged across both arms, together with mean compaction input per call. We relate these lengths to task-state quality within each condition in Fig. 3. Rolling memories are $1.02\times$–$1.36\times$ longer, so the quality gaps in Tab. 3 do not coincide with an order-of-magnitude difference in reader input. The input column differs by about twofold between conditions, which is why the released retention coordinate is not comparable across them (Section 3.2); it does not by itself attribute the observed retention ratios entirely to the denominator.

A.3 Five-dimensional supplementary diagnostics

The primary comparisons use the reference-grounded dimensions in Tab. 1; the three action-oriented dimensions use the shared rubric anchors. The figures below test whether the dimension ordering persists under a separate pooled rescoring.

Table 6: Mean summary length in tokens and mean compaction input per call for the 10 compressors with token records, in the Q_R order of Tab. 3. F = fresh; R = rolling. Lengths are recorded outputs, not an enforced budget.

	Memory length			Input per call	
Compressor label	F	R	R/F	F	R
`gpt-6-astra`	10,960	12,535	1.14	298,923	151,895
`claude-opus-5`	7,144	8,123	1.14	298,923	148,881
`gpt-5.6-sol`	6,942	7,327	1.06	298,923	148,480
`gpt-5.6-luna`	6,640	6,946	1.05	298,923	148,495
`glm-5.3`	6,015	7,034	1.17	299,132	148,150
`gpt-5.6-terra`	7,624	7,774	1.02	298,923	149,028
`deepseek-v4.1-flash`	5,200	5,588	1.07	298,923	147,348
`glm-5.3-flash`	5,438	5,845	1.07	298,923	147,455
`minimax-m3`	3,541	4,813	1.36	298,923	146,865
`claude-sonnet-5`	5,633	6,059	1.08	298,923	147,190

A.3.1 SUB-ITEM SUMMARIES

Population. Figs. 11, 8, and 12 summarize the five-dimensional judge at two complementary levels. The pooled views cover 170,896 rescored judgments across unevenly covered compressor and dataset cells: 84,699 fresh and 86,197 rolling. The per-compressor view uses the DeepSWE subset, with 5,564 judgments per compressor across both conditions. Because the pooled condition counts differ, those views compare condition-level aggregates rather than paired responses.

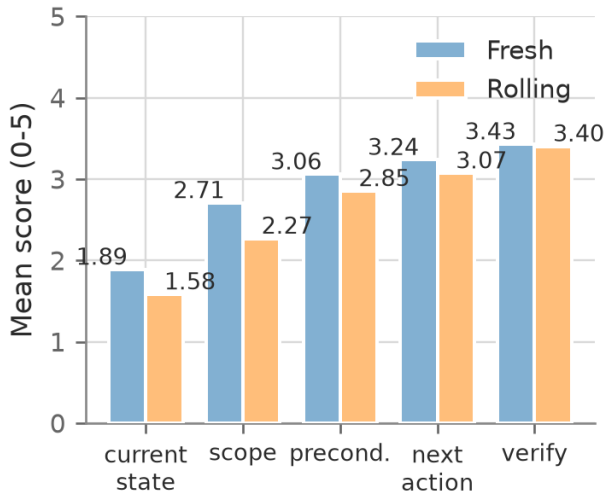

Figure 11: Mean sub-item score by condition over 170,896 rescored judgments, aggregated over unevenly covered cells under a different weighting from Tab. 1.

Condition-level pattern. Current-state correctness averages 1.888 under fresh and 1.584 under rolling, whereas verification calibration averages 3.429 and 3.404 (Fig. 11). Verification is the only dimension whose top-score share rises, by 0.27 percentage points. This separate aggregation therefore reproduces the main table's ordering and concentrates the schedule difference on factual state rather than calibrated phrasing.

Compressor profiles. We again observe the same dimension ordering across compressors (Fig. 12). In the pooled population, CSC is the lowest dimension for 81.3% of answers, and the largest fresh-minus-rolling gaps occur on scope binding (0.437) and CSC (0.304). These figures come from a separate aggregation and are not pooled with Tab. 3.

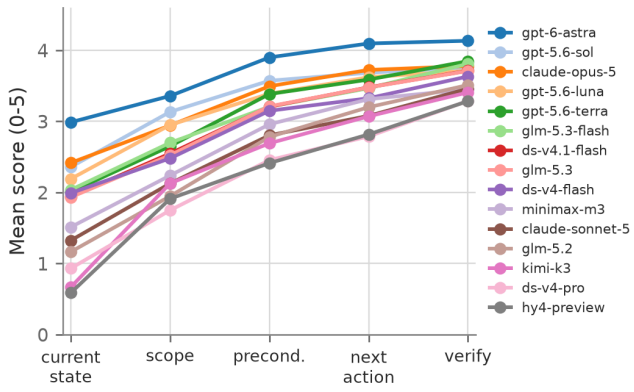

Figure 12: Per-compressor five-dimension profile on the DeepSWE subset (5,564 judgments per compressor). All 15 compressors share the dimension ordering, differing in level and spread.

B Additional Diagnostic Evidence

The retained summaries examine judging-pass sensitivity, sub-item score structure, and a separate compaction strategy. Each states its own population and is reported beside, not pooled with, Tab. 3. Together they test which main-text patterns persist across judge passes, scoring populations, and compaction strategies.

B.1 High rank agreement does not preserve every between-condition difference

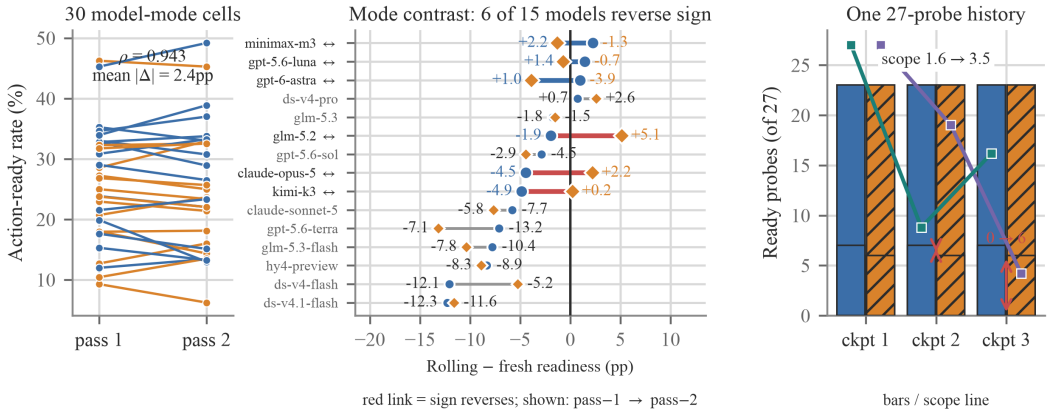

Figure 13: Two scoring passes over byte-identical inputs: readiness per compressor–condition cell (left; blue fresh, orange rolling), the rolling-minus-fresh readiness gap per compressor (middle; pass 1 circle joined to pass 2 diamond), and one rolling-condition history (right; bars ready probes, solid pass 1 and hatched pass 2; line scope binding, rescaled). Cell levels agree closely while per-compressor gaps and single histories move more, so claims rest on cell aggregates.

Two scoring passes with the same judge, the same prompt hash, and byte-identical per-record inputs give the resolution at which differences in these aggregates are meaningful.

Across the 30 compressor–condition cells, readiness changes by 2.4 percentage points on average in absolute value (maximum 6.9), with a signed mean of -0.2 points. The five-dimensional quality mean changes by 0.0916 on average and by up to 0.2206. Cross-pass rank correlations are high for readiness ($\rho = 0.943$), composite quality (0.960), and current-state correctness (0.992; Fig. 13,

left). Cell levels are thus reproducible, and the current-state ordering that carries the main claim is the most reproducible of the three.

The cell-level composite-quality changes give the practical resolution used in Section 3.1: a mean of 0.0916 and a maximum of 0.2206 points. Per-compressor readiness differences are comparable to this scale, so readiness is reported as an aggregate diagnostic and no per-compressor readiness conclusion is drawn (Fig. 13, middle). The factual dimension gaps are supported separately by their consistent direction and, for current-state correctness, by a cross-pass rank correlation of 0.992; no per-cell dimension-level first-pass aggregates are available for a direct gap-by-gap rescoring comparison.

Single histories move more than cell means do (Fig. 13, right). Conclusions in this paper are therefore drawn only from aggregates over many cells, never from a single history.

B.2 Sub-item score structure

The separate sub-item summary covers 170,896 judgments, rather than the 83,460 entries in Tab. 2. Across dimensions, 96.1%–99.8% of scores fall on 0, 3, or 5, so the low factual means are driven by outright misses rather than by uniformly mediocre answers; we plot the between-condition change in the 0-score and 5-score shares (Fig. 5). The 0-score share separates the factual dimensions from the action-discipline ones: current-state correctness scores 0 on 61.0% of judgments (57.6% fresh, 64.4% rolling) and scope binding on 38.1% (32.8%, 43.4%), against 19.4%, 18.2%, and 11.3% for preconditions, next-action validity, and verification calibration. Only 2.8% of current-state scores fall in the partial-credit band (1, 2, or 4), and its median is 0 while every other dimension has median 3. We therefore read the dimension means together with their 0-score and 5-score anchor shares.

Current-state correctness is the lowest-scoring dimension for 81.3% of answers and has the widest spread across compressor–condition cells. We further show that readiness is not a fixed threshold over the five scores (Fig. 8): even among answers clearing all five score thresholds, 40.8% are not judged ready. Supplementary means and compressor profiles appear in Figs. 11 and 12.

B.3 A separate compaction-strategy arm

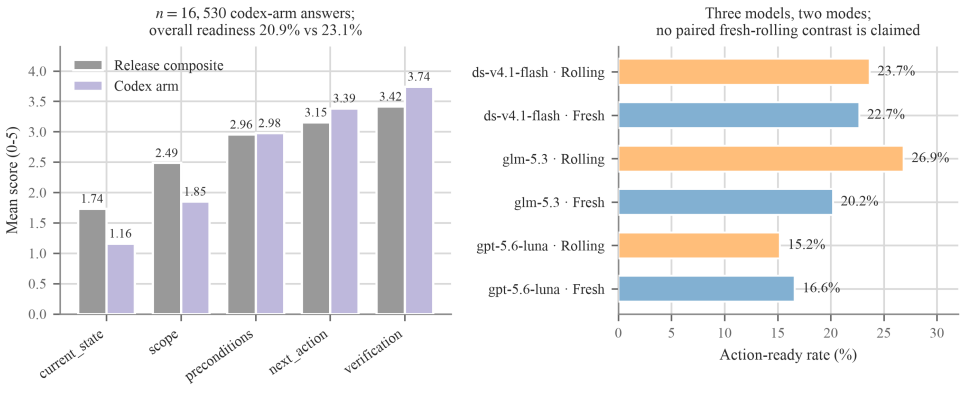

Figure 14: Separate compaction-strategy arm: five-dimensional means beside the pooled diagnostic summary (left) and judge-assigned readiness for three compressors under both schedules (right). Current-state correctness remains the weakest dimension.

The additional arm covers three compressors under a different compaction strategy. Its absolute levels are not pooled with the main release, but the ordering persists: current-state correctness is lowest (1.16) and verification calibration highest (3.74; Fig. 14).

C HISTORICAL CONSISTENCY CHECK

An earlier six-dimensional evaluation used a different judge and scoring protocol. We retain it only as a directional robustness check: the current five-dimensional results do not pool with these historical summaries.

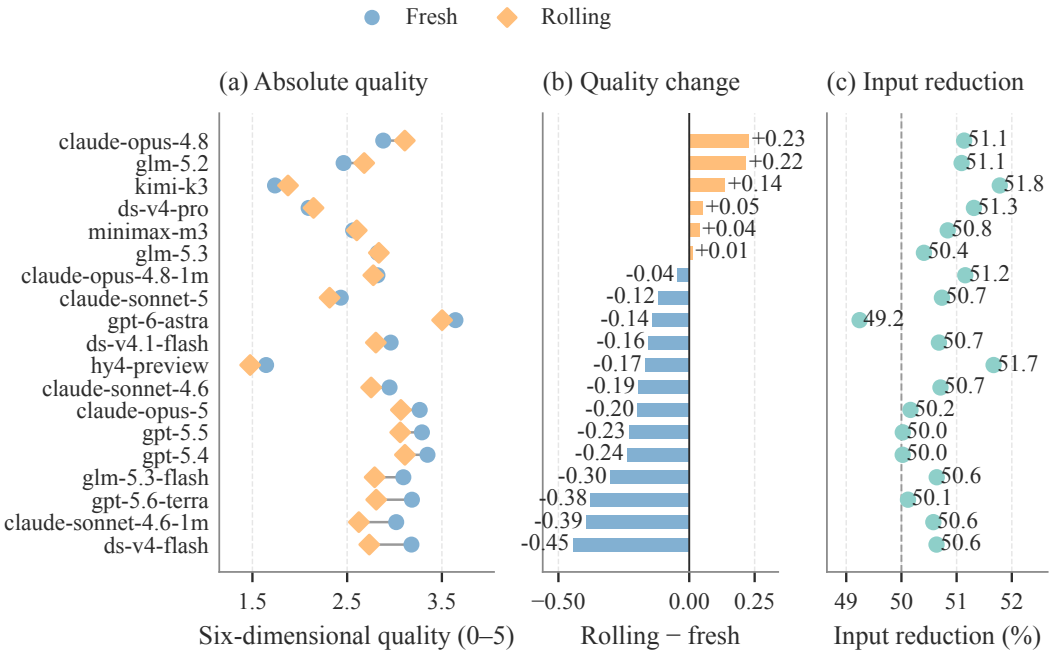

Figure 15: Historical six-dimensional audit of 19 configurations complete in both modes: quality per mode (a), rolling-minus-fresh quality (b), and recorded compressor-input reduction (c).

We align the two schedule scores for the 19 configurations complete in both modes (Fig. 15). Its input panel reports compressor-input totals rather than the released retention percentage. Despite the different rubric, recursive compaction again tends to reduce quality.

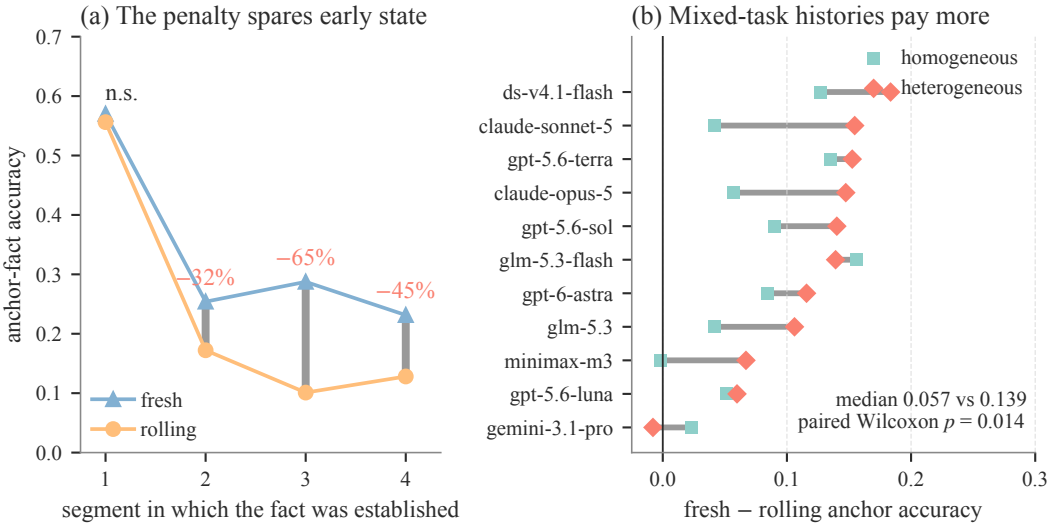

Figure 16: Historical anchor-fact accuracy by introduction segment and history composition; segments are distinct fact cohorts, and the fresh-minus-rolling gap is larger in heterogeneous histories.

We caution that Fig. 16 is not a forgetting curve, because each segment contains a different fact cohort. Its reusable result is the composition contrast: heterogeneous histories show a larger schedule gap, matching the current-protocol arm diagnostic in Fig. 6.

D Related Evaluation Contracts

Functional probes of compressed coding histories are the closest task precedent (Factory Research, 2025), while StateBench and temporal-memory benchmarks provide the closest precedents for supersession, scope, and updates (Liotta, 2025; Parslee, 2026; Wu et al., 2025a; Hu et al., 2026). EndureBench combines these concerns on fixed recorded coding histories: it asks for checkpoint-current values and their source scope while holding trajectories, probes, and checkpoint boundaries constant.

Memory systems also differ in what they may consult. Retrieval and memory-tier systems can reacquire omitted evidence (Packer et al., 2023; Gutiérrez et al., 2024; Wang et al., 2023; Wu et al., 2026); repeated text compaction and KV recompression carry different representations forward (Zerhoudi et al., 2026; O'Neill, 2026); executing-agent benchmarks let memory alter future actions and observations (Kang et al., 2026; Wu et al., 2025b; Ye et al., 2026; Zhou et al., 2026; Zeng et al., 2026). EndureBench instead fixes the reader's evidence source to compacted text memory and measures factual recovery before trajectory divergence. We summarize the key contract differences across these settings in Tab. 7.

D.1 Comparison of access and outcome contracts

The settings in Tab. 7 differ in what memory may consult and whether memory changes later execution. EndureBench fixes the recorded history and reader, then varies the compaction schedule defined in Section 2.

Table 7: Related evaluation settings under their own protocols. Raw-history access and environment execution are independent dimensions.

Setting	Evidence/access contract	Principal outcome and relation to this work
Long-context QA (Bai et al., 2024; Hsieh et al., 2024; Modarressi et al., 2025)	Supplied long input; length, content, or matching difficulty controlled	Answer quality; does not itself impose repeated text-memory replacement
Conversational memory (Maharana et al., 2024; Wu et al., 2025a)	Multi-session history with memory/retrieval alternatives	Historical QA, temporal reasoning, and updates; access rules depend on the method
Environment experience (Wu et al., 2026)	Stored trajectories can be searched to assemble evidence for a fixed reader	Environment-state and workflow QA; raw evidence can be recovered at query time
Compression probes (Factory Research, 2025)	Questions about compressed work histories	Functional answer quality and artifact/continuity rubric; closest scoring precedent
Repeated compaction (Zerhoudi et al., 2026; O'Neill, 2026)	Persistent rules under repeated text compression, or fresh/incremental KV recompression	Rule-following behavior and recompression sensitivity; distinct memory representations and targets
Executing agents (Kang et al., 2026; Zeng et al., 2026)	Memory management changes future actions and observations	Task/environment outcomes; trajectory divergence is part of the system effect
EndureBench	Same recorded histories under two compaction schedules: fresh makes one call over the raw prefix; rolling makes one call per checkpoint over the previous memory and the new block; reader sees the memory only	Checkpoint-current factual probes; recovery level per condition and the gap between them, rather than environment continuation outcomes

E TRACEABLE CHECKPOINT CASE

One homogeneous DeepSWE history, `cbv5-homo-4`, concatenates six sessions into three blocks of 151,559, 150,298, and 125,900 tokens. Its 26 anchors and one dynamic chain are established in the first block and queried at every checkpoint. Under rolling with `gpt-5.6-luna`, current-state correctness falls from 4.26 at the first checkpoint to 1.15 at the third while memory length rises from 6,206 to 6,390 tokens; the five-axis mean falls more moderately, from 4.85 to roughly 3.1–3.7.

F HARNESS ROBUSTNESS

The four harnesses were executed separately and differ in prompt form, reader, and occasional output cap, so they test robustness of the qualitative pattern rather than a causal prompt effect. Here *fresh* denotes `single_shot` in the source report.

Table 8: Comparison contract for the four harness configurations.

Harness	Prompt form	Length cue	Reader	Coverage / cap
Claude Code	Nine-section summary with analysis block	none	`gpt-5.5`	complete; 16K
WorkBuddy	Mandatory nine-section template	none	`glm-5.2`	complete; one 32K retry
DSH	Eight-section checkpoint handoff	terse bullets	`glm-5.2`	complete; some 32K retries
Codex	Loosely structured checkpoint handoff	be concise	`glm-5.2`	complete; some 32K retries

The compared compressors are `deepseek-v4.1-flash`, `gpt-5.6-luna`, `glm-5.3`, and `hy4-preview`. Across all 32 harness–compressor–condition cells, current-state correctness remains the weakest dimension.

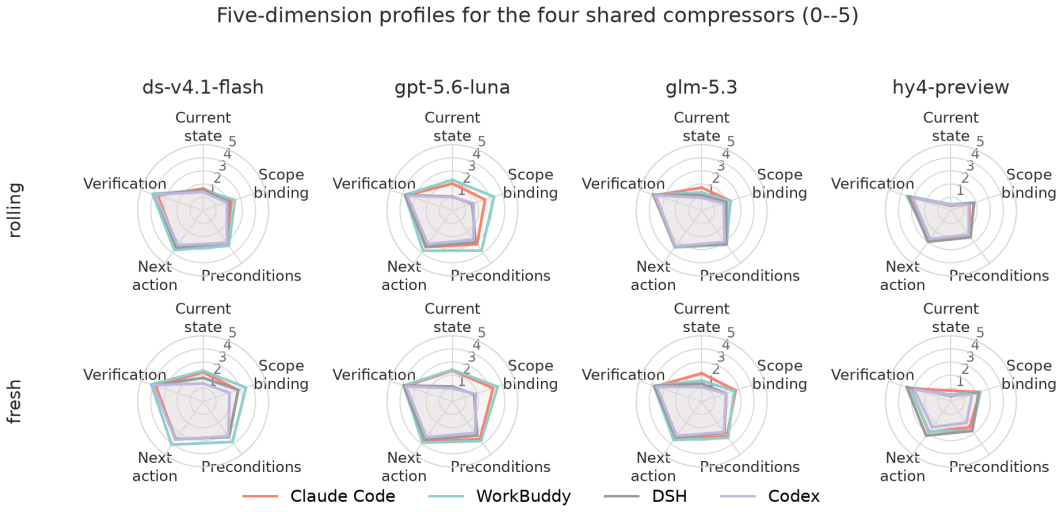

Figure 17: Five-dimensional profiles for four compressors under four harness configurations; current-state correctness is the weakest axis in every cell, though absolute levels differ.

G CHECKPOINT-RESOLVED VALIDATION

The current release aggregates over checkpoints. A future item-level release needs only the coordinates in Tab. 9 to separate elapsed context, recursive exposure, and fact revision.

With those identifiers, checkpoint accuracy can compare fresh and rolling on the same histories and valid probe cohort. Stable-anchor survival should be reported separately from pointwise accuracy: a fact recovered after an earlier failure contributes to current accuracy but not uninterrupted survival. Dynamic facts should be indexed by the effective update event so that restored values and frequently revised identities are handled correctly.

Table 9: Minimum metadata for checkpoint-resolved evaluation.

Coordinate	Purpose
History and checkpoint ID	Aligns the same recorded prefix across conditions
Source-scoped fact ID and version	Identifies the checkpoint-current value and excludes stale alternatives
Introduction and update checkpoint	Separates fact age from age of the current version
Compaction round and token counts	Separates recursive exposure from elapsed context and input size
Memory, question, reference, and answer	Supports direct attribution of omission, corruption, or extraction failure

G.1 Focused validation of factual recovery

The reported case in App. E supplies candidate facts for a small item-level audit. We define the required steps in Tab. 10.

Table 10: Focused validation plan for attributing factual-recovery errors.

Stage	Required material	Decision produced
Select	A stable anchor, a same-scope revision, and two tasks sharing an attribute; successes and failures retained	Tests current value and source scope without selecting only observed failures
Verify	Source/task identifier, evidence, effective checkpoint, question, and reference	Accepts, corrects, or excludes each reference before model scores are inspected
Attribute	Saved memories and answers, plus a checkpoint-bounded source excerpt for the same reader	Separates memory omission or corruption from reader failure despite sufficient evidence

Two checkpoints per fact and one existing compressor are sufficient. An evidence-only reader control establishes whether the question is answerable from the recorded source; a no-history control is added only when the wording may leak the answer. This audit can sharpen the interpretation of the aggregate scores without introducing a new compressor, compaction strategy, or rollout.